FULL SPECIFICATION

TCC8900

High Performance and Low-Power Processor

For Digital Media Applications

TCC8900_FULL_SPEC

Rev. 1.05

Dec 04, 2009

Telechips

DISCLAIMER

All information and data contained in this material are without any commitment, are not to be considered as an offer for conclusion of a contract, nor shall they be construed as to create any liability. Any new issue of this material invalidates previous issues. Product availability and delivery are exclusively subject to our respective order confirmation form; the same applies to orders based on development samples delivered. By this publication, Telechips, Inc. does not assume responsibility for patent infringements or other rights of third parties that may result from its use.

Further, Telechips, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.
No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Telechips, Inc.
This product is designed for general purpose, and accordingly customer be responsible for all or any of intellectual property licenses required for actual application. Telechips, Inc. does not provide any indemnification for any intellectual properties owned by third party.
Telechips, Inc. can not ensure that this application is the proper and sufficient one for any other purposes but the one explicitly expressed herein. Telechips, Inc. is not responsible for any special, indirect, incidental or consequential damage or loss whatsoever resulting from the use of this application for other purposes.

COPYRIGHT STATEMENT

Copyright in the material provided by Telechips, Inc. is owned by Telechips unless otherwise noted.
For reproduction or use of Telechips' copyright material, permission should be sought from Telechips. That permission, if given, will be subject to conditions that Telechips' name should be included and interest in the material should be acknowledged when the material is reproduced or quoted, either in whole or in part. You must not copy, adapt, publish, distribute or commercialize any contents contained in the material in any manner without the written permission of Telechips.
Trade marks used in Telechips' copyright material are the property of Telechips.

Important Notice

This product may include technology owned by Microsoft Corporation and in this case it cannot be used or distributed without a license from Microsoft Licensing, GP.

For customers who use licensed Codec ICs and/or licensed codec firmware of mp3:

"Supply of this product does not convey a license nor imply any right to distribute content created with this product in revenue-generating broadcast systems (terrestrial. Satellite, cable and/or other distribution channels), streaming applications(via internet, intranets and/or other networks), other content distribution systems(pay-audio or audio-on-demand applications and the like) or on physical media(compact discs, digital versatile discs, semiconductor chips, hard drives, memory cards and the like). An independent license for such use is required. For details, please visit http://mp3licensing.com".

For customers who use other firmware of mp3:

"Supply of this product does not convey a license under the relevant intellectual property of Thomson and/or Fraunhofer Gesellschaft nor imply any right to use this product in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit http://mp3licensing.com".

For customers who use Digital Wave DRA solution:

"Supply of this implementation of DRA technology does not convey a license nor imply any right to this implementation in any finished end-user or ready-to-use terminal product. An independent license for such use is required."

Revision History

Date	Revision	Description
2008-12-11	0.00	* Initial Release
2009-02-10	0.01	* The revision number of “PART 2. SMU & PMU” was changed to 0.01 * The revision number of “PART 3. GPIO” was changed to 0.01 * The revision number of “PART 4. CORE & MEMORY BUS” was changed to 0.01 * The revision number of “PART 5. IO BUS” was changed to 0.01 * The revision number of “PART 6. DDI BUS” was changed to 0.01 * The revision number of “PART 7. VIDEO BUS” was changed to 0.01 * The revision number of “PART 8. GRAPHIC BUS” was changed to 0.01
2009-02-25	0.02	* The revision number of “PART 2. SMU & PMU” was changed to 0.02 * The revision number of “PART 3. GPIO” was changed to 0.02 * The revision number of “PART 4. CORE & MEMORY BUS” was changed to 0.02 * The revision number of “PART 5. IO BUS” was changed to 0.02 * The revision number of “PART 6. DDI BUS” was changed to 0.02 * The revision number of “PART 7. VIDEO BUS” was changed to 0.02 * The revision number of “PART 8. GRAPHIC BUS” was changed to 0.02
2009-06-05	0.03	* The “CHIPSPEC” was appended. * The revision number of “PART 2. SMU & PMU” was changed to 0.03 * The revision number of “PART 3. GPIO” was changed to 0.03 * The revision number of “PART 5. IO BUS” was changed to 0.03 * The revision number of “PART 6. DDI BUS” was changed to 0.03 * The revision number of “PART 7. VIDEO BUS” was changed to 0.03 * The revision number of “PART 8. GRAPHIC BUS” was changed to 0.03
2009-08-07	1.00	* The revision number of “CHIPSPEC” was changed to 1.00 * The revision number of “PART 1. OVERVIEW” was changed to 1.00 * The revision number of “PART 2. SMU & PMU” was changed to 1.00 * The revision number of “PART 3. GPIO” was changed to 1.00 * The revision number of “PART 4. CORE & MEMORY BUS” was changed to 1.00 * The revision number of “PART 5. IO BUS” was changed to 1.00 * The revision number of “PART 6. DDI BUS” was changed to 1.00 * The revision number of “PART 7. VIDEO BUS” was changed to 1.00 * The revision number of “PART 8. GRAPHIC BUS” was changed to 1.00
2009-08-18	1.01	* The revision number of “CHIPSPEC” was changed to 1.01
2009-09-04	1.02	* The revision number of “CHIPSPEC” was changed to 1.02 * The revision number of “PART 2. SMU & PMU” was changed to 1.01 * The revision number of “PART 5. IO BUS” was changed to 1.01 * The revision number of “PART 6. DDI BUS” was changed to 1.01
2009-10-09	1.03	* The revision number of “CHIPSPEC” was changed to 1.03 * The revision number of “PART 2. SMU & PMU” was changed to 1.02 * The revision number of “PART 4. CORE & MEMORY BUS” was changed to 1.01 * The revision number of “PART 5. IO BUS” was changed to 1.02 * The revision number of “PART 6. DDI BUS” was changed to 1.02
2009-11-06	1.04	* The revision number of “CHIPSPEC” was changed to 1.04 * The revision number of “PART 4. CORE & MEMORY BUS” was changed to 1.02 * The revision number of “PART 5. IO BUS” was changed to 1.03
2009-12-04	1.05	* The revision number of “CHIPSPEC” was changed to 1.05

CHIP SPECIFICATION

TCC8900

High Performance and Low-Power Processor

For Digital Media Applications

Rev. 1.05

Dec 04, 2009

Telechips

Revision History

Date	Revision	Description
2008-12-05	0.00	* Initial Release
2008-12-11	0.01	* Swapping of EDIXD0 and EDIXD8 in the Pin Description * Swapping of EDIXD1 and EDIXD9 in the Pin Description * Swapping of EDIXD2 and EDIXD10 in the Pin Description * Swapping of EDIXD3 and EDIXD11 in the Pin Description
2008-12-29	0.02	* The ball maps of H1, J1, K1, L1, M1, N1, P1, T1, U1, V1, W1, Y1, L2, M2, N2, P2, R2, U2, V2, W2, Y2, N3, P3, R3, T3, U3, V3, W3, Y3, P4, R4, T4, U4, V4, W4, Y4, R5, T5, U5, V5, W5, Y5, P6, R6, V6, W6, Y6, P7, T7, U7, V7, Y7, T8, U8, V8, W8, Y8, P9, R9, V9, W9, Y9, P10, R10, W10, Y10, T13, T14, N17, N18, M20, N20, P20 were changed.
2009-01-22	0.03	The ball description on the Table 3.9 has been changed. - GPIOB[26] → G6 - GPIOB[27] → C11 - GPIOB[28] → G7 - GPIOB[29] → D10
2009-02-05	0.04	* The recommended operating frequency for each block was added.
2009-03-10	0.05	* The resolution of the ADC(TSADC) was changed from 10 bits to 12 bits.
2009-06-05	0.06	* The order of the TCO3 ~ TCO0 is changed to TCO0 ~ TCO3. * The initial states of the GPIO were changed.
2009-08-07	1.00	* Changing AC SPEC DAI(I2S) * The initial states of the GPIOs have been changed.
2009-08-18	1.01	* The ball map or pin description has been changed. * The value of Ccard for SDMMC has been changed from 50pF to 30pF. * Updating “Timing Parameters for Each Symbol”
2009-09-04	1.02	* The initial states of TMS, TCK, TDI has been changed from “I” to “IU.” * The recommended operating conditions & frequency table has been updated. * The absolute maximum rating of OTG_VBUS has been added.
2009-10-09	1.03	* “Electrical Characteristics for DDR” has been added.
2009-11-06	1.04	* The recommended operationg conditions for PWRPLL, PWRUSB12, PWRCORE has been modified.
2009-12-04	1.05	* Update minimum clock periold of HCLK in NAND flash controller

TABLE OF CONTENTS

Contents

1 Introduction 1-1
1.1 TCC8900 Features 1-2
1.2 Applications 1-5
1.3 Block Diagram 1-6
2 Hardware Features 2-7
3 PIN Description 3-15
3.1 TCC8900 Pin Description 3-15
3.2 TCC8900 I/O Type 3-25
4 Package Information 4-27
4.1 Dimension 4-27
4.2 Ball Map 4-28
5 Electrical Specification 5-29
5.1 Absolute Maximum Ratings 5-29
5.2 Recommended Operating Conditions 5-30
5.3 Recommended Operating Frequency 5-31
5.4 Electrical Characteristics for Power Supply 5-34
5.5 Electrical Characteristics for General I/O 5-35
5.6 Electrical Characteristics for PLL 5-36
5.7 Electrical Characteristics for Video DAC 5-37
5.8 Electrical Characteristics for ADC(for Touch Screen) 5-38
5.9 Electrical Characteristics for HDMI PHY 5-39
5.10 Electrical Characteristics for LVDS 5-40
5.11 Electrical Characteristics for SATA 5-41
5.12 Electrical Characteristics for LCD Interface 5-42
5.13 Electrical Characteristics for Camera Interface 5-44
5.14 Electrical Characteristics for External Host Interface (EHI) 5-46
5.15 Electrical Characteristics for SD/MMC Controller 5-47
5.16 Electrical Characteristics for I2C Controller 5-48
5.17 Electrical Characteristics for SPDI/F Transmitter 5-48
5.18 Electrical Characteristics for DAI(I2S) 5-49
5.19 Electrical Characteristics for Nand Flash Controller 5-51
5.20 Electrical Characteristics for UART Controller 5-54
5.21 Electrical Characteristics for DDR 5-57

Figures

Figure 1.1 TCC8900 Functional Block Diagram 1-6
Figure 2.1 Memory Organization 2-7
Figure 4.1 TCC8900 Package Dimension 4-27
Figure 4.2 TCC8900 Ball Map 4-28
Figure 5.1 LVDS Transmit Timing Diagram 5-40
Figure 5.2 Timing Diagram for LCD Controller 5-42
Figure 5.3 Timing Diagram Data Output Referenced to PXCLK 5-42
Figure 5.4 Timing Diagram Data Output Referenced to LCDSI 5-43
Figure 5.5 Timing Diagram for Camera Interface 5-44
Figure 5.6 Timing Diagram Data Output Referenced to CCLK 5-44
Figure 5.7 EHI Timing Diagram 5-46
Figure 5.8 Timing Diagram for SD/MMC Controller 5-47
Figure 5.9 Timing Diagram for I2C Controller 5-48
Figure 5.10 Timing Diagram for SPDI/F Transmitter 5-48
Figure 5.11 Timing Diagram for DAI (receiver) 5-49
Figure 5.12 Timing Diagram for DAI Transmitter 5-50
Figure 5.13 Timing Diagram for Command Latch Enable Cycle 5-51
Figure 5.14 Timing Diagram for Single Address Latch Cycle 5-51
Figure 5.15 Timing Diagram for Linear Address Latch Cycle 5-51
Figure 5.16 Timing Diagram for Single Data Write Cycle 5-51
Figure 5.17 Timing Diagram for Linear Data Write Cycle 5-52
Figure 5.18 Timing Diagram for Single Data Read Cycle 5-52
Figure 5.19 Timing Diagram for Linear Data Read Cycle 5-52
Figure 5.20 Timing Diagram for TXD 5-54
Figure 5.21 Timing Diagram for RXD 5-54
Figure 5.22 Timing Diagram for TX Operation with H/W Flow Control 5-55
Figure 5.23 Timing Diagram for nCTS Timing Diagram 5-55
Figure 5.24 Timing Diagram for RX Operation with H/W Flow Control 5-56
Figure 5.25 Timing Diagram for nRTS Timing Diagram 5-56

Figure 5.26 Write Cycle Timing 5-57
Figure 5.27 Read Cycle Timing 5-57

Tables

Table 1.1 TCC8900 Features 1-2
Table 2.1 ARM1176JZFS Processor 2-7
Table 2.2 Video Controller 2-8
Table 2.3 Camera Interface 2-8
Table 2.4 Video Output Interface 2-9
Table 2.5 DAI/CDIF Controller 2-10
Table 2.6 SPDIF Controller 2-10
Table 2.7 External Device Interface 2-10
Table 2.8 USB 1.1 Host 2-10
Table 2.9 USB 2.0 OTG(On-The-GO) 2-10
Table 2.10 nano PHY for USB2.0 OTG and USB1.1 Host 2-11
Table 2.11 External Host Interface 2-11
Table 2.12 SD/MMC Controller 2-11
Table 2.13 Memory Stick Controller 2-11
Table 2.14 Nand Flash Controller 2-12
Table 2.15 IDE Interface 2-12
Table 2.16 UART Interface 2-12
Table 2.17 GPSB Interface 2-13
Table 2.18 General DMA Controller 2-13
Table 2.19 Vectored Interrupt Controller 2-13
Table 2.20 Timer 2-13
Table 2.21 ADC 2-14
Table 2.22 Real Time Clock 2-14
Table 3.1 Power/Ground Information 3-15
Table 3.2 PWRGPIOC Group I/O Pin Description 3-16
Table 3.3 PWRGPIOF Group I/O Pin Description 3-17
Table 3.4 PWRGPIOE Group I/O Pin Description 3-18
Table 3.5 PWRGPIOA Group I/O Pin Description 3-19
Table 3.6 PWRADC Group I/O Pin Description 3-20
Table 3.7 PWRETC Group I/O Pin Description 3-20
Table 3.8 PWRGPIOD Group I/O Pin Description 3-21
Table 3.9 PWRGPIOB Group I/O Pin Description 3-22
Table 3.10 PWRMEMQ Group I/O Pin Description 3-23
Table 3.11 PWROSC Group I/O Pin Description 3-24
Table 3.12 PWRUSB33 Group I/O Pin Description 3-24
Table 3.13 PWRUSBH Group I/O Pin Description 3-24
Table 3.14 PWRRTC Group I/O Pin Description 3-24
Table 3.15 PWRSATAOSC Group I/O Pin Description 3-24
Table 3.16 PWRSATA Group I/O Pin Description 3-24
Table 3.17 PWRHDMIOSC Group I/O Pin Description 3-24
Table 3.18 PWRHDMI Group I/O Pin Description 3-24
Table 3.19 PWRLVDS Group I/O Pin Description 3-24
Table 3.20 PWRDAC Group I/O Pin Description 3-24
Table 3.21 TCC8900 I/O Type 3-25
Table 5.1 Recommended Operating Conditions 5-30
Table 5.2 Recommended Operating Frequency 5-31
Table 5.3 Peak Power Consumption 5-34
Table 5.4 DC Electrical Specification for General I/O 5-35
Table 5.5 DC Electrical Characteristics for PLL0 5-36
Table 5.6 AC Electrical Characteristics for PLL0 5-36
Table 5.7 DC Electrical Characteristics for PLL1/2/3 5-36
Table 5.8 AC Electrical Characteristics for PLL1/2/3 5-36
Table 5.9 DC Electrical Characteristics for DAC 5-37
Table 5.10 DC Electrical Characteristics for ADC 5-38
Table 5.11 AC Electrical Characteristics for ADC 5-38
Table 5.12 DC Electrical Characteristics for HDMI PHY 5-39
Table 5.13 AC Electrical Characteristics for HDMI Oscillator 5-39
Table 5.14 DC Electrical Characteristics for LVDS 5-40
Table 5.15 AC Electrical Characteristics for LVDS 5-40
Table 5.16 DC Electrical Characteristics for SATA 5-41
Table 5.17 AC Electrical Characteristics for SATA Oscillator 5-41
Table 5.18 AC Electrical Characteristics for SATA TX/RX 5-41
Table 5.19 Timing Parameters for Each Symbol 5-42

Table 5.20 I/O Function Name for Corresponding Signal Name 5-42
Table 5.21 Timing Parameters for Each Symbols 5-43
Table 5.22 Timing Parameters for Each Symbol 5-45
Table 5.23 I/O Function Name for Corresponding Signal Name 5-45
Table 5.24 Timing Parameters for Each Symbol 5-47
Table 5.25 Timing Parameters for Each Symbol 5-48
Table 5.26 Timing Parameters for Each Symbol 5-48
Table 5.27 Timing for DAI (receiver)..... 5-49
Table 5.28 Timing Parameters for Each Symbols 5-50
Table 5.29 Timing Parameters for Each Symbol 5-53
Table 5.30 I/O Function Name for Corresponding Signal Name 5-53
Table 5.31 Timing Parameters for Each Symbol 5-54
Table 5.32 Timing Parameters for Each Symbol 5-54
Table 5.33 Timing Parameters for Each Symbol 5-55
Table 5.34 Timing Parameters for Each Symbols 5-56
Table 5.35 DDR Interface Timing Parameters..... 5-58

1 Introduction

The TCC8900 is the system LSI for digital multimedia applications which based on ARM1176JZF-S, ARM's proprietary RISC CPU core. It can decode and encode various types of video and audio standards by software and dedicated hardware accelerators – JPEG / MPEG1 / MPEG2 / MPEG4-SP/ASP / H.264 / VC-1 / RV and other types of video standard and MP3 / AAC / MPEG4-AAC / MPEG4-BSAC and other types of standard for audio

The on-chip high speed USB2.0 device controller enables the data transmission between a personal computer and storage device such as NAND flash, HDD, CD, SD, MMC and Memory Stick etc, which can be controlled by the TCC8900.

1.1 TCC8900 Features

Table 1.1 TCC8900 Features

Category	Description
PROCESSORS (ARM1176JZF-S)	* **TrustZone™** security extensions * High-speed **AMBA AXI** Bus Interface * High performance integer processor * 8 stage pipelines * Separate load-store and arithmetic pipelines * Branch prediction with return stack * Instruction and Data MMU(Memory Management Units) * Micro TLB structures backed by a unified Main TLB. * Virtually indexed and physically addressed caches * Level one TCM(Tightly-Coupled Memory) * 16KBs ITCM and 16KBs DTCM * 16KBs Instruction/Data Caches * Including a non-blocking data cache with Hit-Under-Miss(HUM) * **Vector-Floating-Point(VFP)** coprocessor support * ARM Jazelle technology for efficient embedded Java execution * JTAG interface for code debugging
MEMORY ORGANIZATION[1]	* Internal(On-Chip) Memory * 16KB Boot-ROM (EHI, NAND, USB Boot with security and etc.) * 16KB Internal SRAM (Shared by Hardware) * External(Off-Chip) Memory * SDRAM : up to 200MHz * mDDR SDRAM : up to 200MHz(400Mbps) * DDR SDRAM : up to 200MHz(400Mbps) * DDR2 SDRAM : up to 400MHz(800Mbps) * 32bits Data Bus
VIDEO CODEC	*** Decompressor[2] (Decoder) – up to 30fps @ Full-HD (1920x1080)** * H.263 : up to 40Mbps * MPEG 1/2 * Up to Main Profile @ High Level * Max. bitrate : up to 40Mbps * MPEG4-ASP * Up to Advanced Simple Profile Including DivX3.x * Max. bitrate : up to 35Mbps * MPEG4-AVC(H.264) * Up to High Profile @ Level 5.1[3] * Max. bitrate : up to 25Mbps * VC-1 * Up to Advanced Profile @ Level 3.0 * Max. bitrate : up to 30Mbps * RV * Real Video 10 (Backward Compatible for RV8.9) * Max. bitrate : up to 30Mbps * JPEG : up to 12Mpixels *** Compressor[4] (Encoder) – up to 30fps or 24fps @HD(1280x720)** * H.263 : up to 30fps @ HD(1280x720p) * MPEG4-ASP : up to 30fps @ HD(1280x720p) * H.264 : up to 24fps @ HD(1280x720p) * JPEG : up to 12Mpixels
GRAPHIC ENGINE	*** 2D/3D Graphic** * High Geometry and Pixel Processing * Up to 7M polygon[5] * Full OpenVG v1.1 Support * Lines, Squares, Triangles, Points * Vector Graphics * ROP 3/4 * Arbitrary Rotation / Scaling * Alpha Blending

1 The maximum operation frequency can be limited by the system configuration.
2 The performance of the video decoding can be limited by the system configuration.
3 Up to 1920x1080 30fps 25Mbps : Ref.Frame : 96Mbyte for VPU Memory in case of 32 Ref frame
4 The performance of the video encoding can be varied by the system configuration.
5 The maximum performance for this can be varied by the system configuration.

	* Multitexture BItBLT * Full OpenGL ES v2.0, v1.x Support * 4X /16X FSAA * Flat/Gouraud Shading * Perspective Correct Texturing * Point Sampling/Bilinear/Trilinear Filtering * Mipmapping * Multi Texturing * Dot3 Bump Mapping * Alpha Blending * Stencil Buffering (4-bit) * JSR 184 * Point Sprites * 2 bit per pixel Texture Compressing (FLXTC) * 4 bit per pixel Texture Compressing (ETC) *** Overlay Mixer** * 8bpp (RGB332) * RGB (444, 454, 555, 565, 666, 888) * Alpha-RGB (444, 454, 555, 666, 888) * Sequential YUV (444, 422) * Separated YUV (444, 440, 422, 420, 411, 410) * Interleaved YUV (422, 420) * BitBLT (16 Raster Operations) * 3 Channel Source Mirror/Flip/90°, 180°, 270° Rotate * 1 Channel Destination Mirror/Flip/90°, 180°, 270° Rotate * 3 Channel Arithmetic Operation * 3 Channel YCbCr-to-RGB Color Space Converting * Overlay and Alphablending (2 overlay, 256-level alphablending) * Color LUT * Dithering
IMAGE ENHANCEMENT	*** Histogram Measurement** * Analyze the Luminance Components * Multi-frames Average Mode * User-defined Pixel Segments Support *** Contrast Enhancement** * User-defined Scaling Segments * Multi-frames Average Mode *** De-Interlacer** * Motion-adaptive and Pixel-based Processing * Film-mode Detection * Simple Edge-oriented Mode * Advanced Spatial-Temporal Mode *** Noise Reduction** * Directional-Smoothing Filter * Temporal-Recursive Filter * Noise Estimation *** Sharpness** * Spatial High-pass Filter
VIDEO IN/OUT	*** Video Output** * 2 Display Controllers * Controller 0 has single image channel * Controller 1 has 3 image channels * Progressive or Interlaced Digital Video Output * Supported Functions * 3 Channel Overlay / Chroma-Keying / Alpha-blending – Only for Controller 1 * Gamma Correction * Look-up table for Indexed or RGB Color * Contrast, Brightness, Hue Function Supported. * Supported Output Media * TFT-LCD Supported * HDMI Output Supported : up to 1920x1080p * LVDS Output Supported : up to 1280x720p * Composite TV-Out (NTSC/PAL) * Dual-Display Supported[6] * Two types of supported media * CPU Type Main/Sub LCD : Time Shared

6 The maximum resolution and combination of image channels can be determined by the system configuration

	*** Video Input** * CCIR-601/656 Interface * Camera Input Supported * 1 Channel Overlay / Chroma-Keying * Input Image Scaler – Output resolution is up to 4080x4080
SPECIAL HARDWARE	* **DVB-H MPE-FEC** *** ECC Controller** * **Touch Screen Interface** * 10/12 bits 8CH ADC
AUDIO	* I2S Master & Slave Interface * 7.1 Channel Supported * SPDIF Transmitter/Receiver * 5.1 Channel Supported * CD I/F * I2S Slave Interface
STORAGE INTERFACE	* USB 2 Channel Interface * 1 Channel for USB 2.0 OTG * 1 Channel for USB1.1 Host * UDMA 33/66, PIO Mode * NAND Flash Interface * 8 Bits / 16 Bits * 4 CS Supported * SD/MMC Controller : SD, MMC, SDIO, Ce-ATA * Memory Stick Pro/Pro-HG Supported * S-ATA Host * Generation 1 : 1.5Gbps * Generation 2 : 3.0Gbps
HOST INTERFACE	* EHI(External Host Interface) * 8, 16bits, 18bits * 2 Channels * Bypass to LCD Port (CPU Type)
STREAMING INTERFACE	*** TS Interface** * 2 Channel TS serial interfaces shared by GPSB * 2 Channel TS parallel interfaces
PERIPHERALS	* **UART** – up to 6 Channels * I2C (2 Master and 1 Slave) – 2 Channels * **GPSB**(General Purpose Serial-Bus – Master/Slave) – 6 Channels * Serial TS Interface Supported (Receiver Only) for 2 channels * Infra-Red Remote Receiver * **Timers** * Four 16-bit timers with PWM output/counters * Two 20-bit timers * One 32-bit timer * **DMA** - 12 Channels * **ADC** * 8-Channel General purpose 12-bit * Shared by Touch Screen Controller
PMU	* **RTC** : Power-Down Mode & Auto-wakeup * Internal power island for saving the current consumption.
PROCESS	* 65nm CMOS

1.2 Applications

Category	Description
Mobile Application Processor	* Smart-Phone Application * Support of all the Mobile / Digital broadcasting services * T-DMB/S-DMB/DVB-H/ CMMB / ATSC-MH * DVB-T / DTTB * Support of Touch Screen Controller * Low-power consumption for power-down mode
Mobile Co-processor	* Mobile-TV Solution * Mobile Multimedia Co-Processor
Portable Devices	* High quality Multimedia Player * Low cost PDA application * Multimedia Host Player
Portable Navigation	* 2D/3D Navigation * Navigation with A/V System and D-TV
CAR	* Car Navigator - Multimedia Play * Multimedia Host Player - USB1.1 Host

1.3 Block Diagram

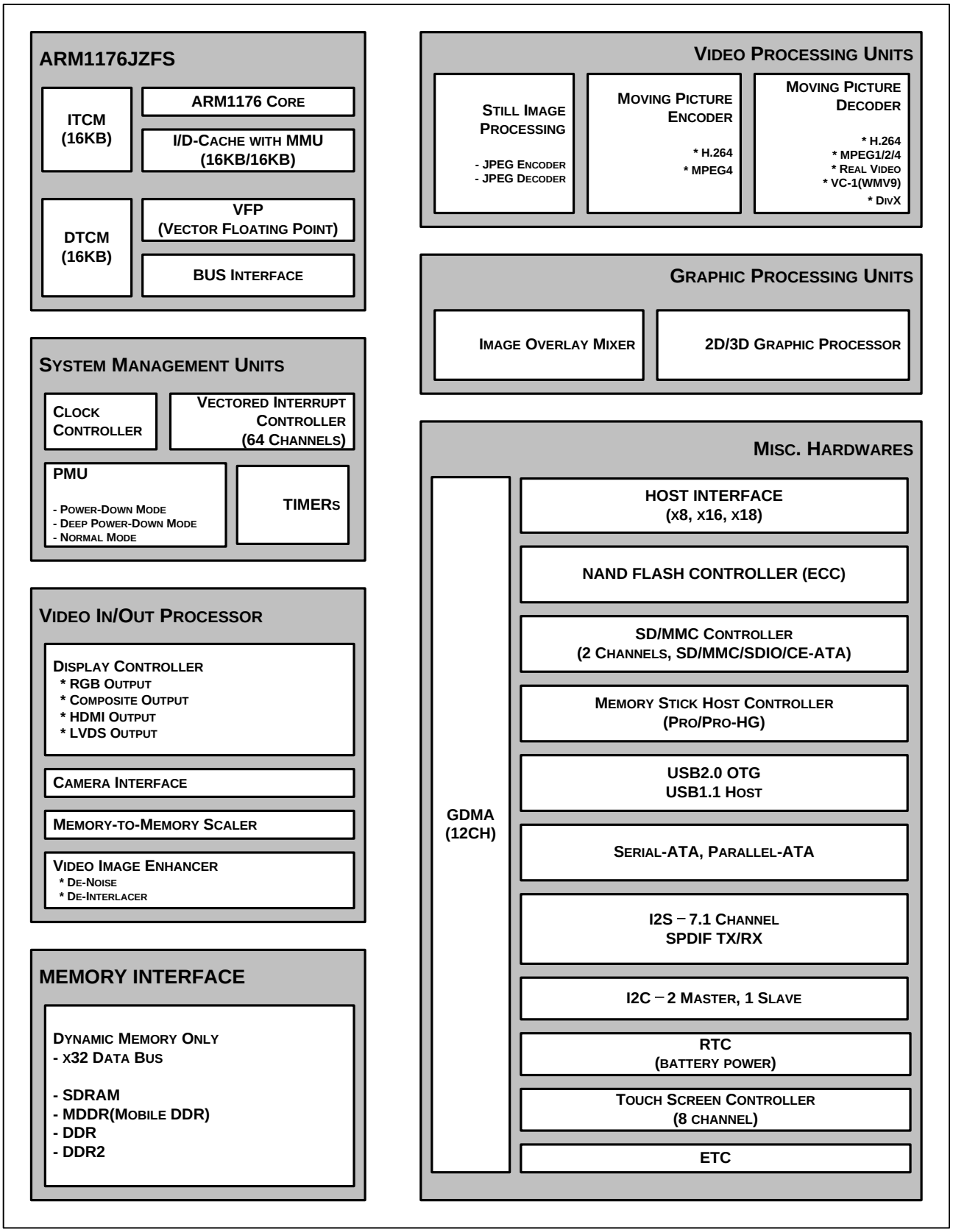


Figure 1.1 TCC8900 Functional Block Diagram

2 Hardware Features

Table 2.1 ARM1176JZFS Processor

ARM1176JZFS[7]	Key Features
Cache Organizations	* 16KBs/16KBs I/D Caches * I/D MMU Supported * Java Accelerator
Tightly Coupled Memory	* 16KB Instruction TCM * 16KB Data TCM
Debug Interface	* JTAG Synchronized Ports with RTCK

Memory Map	Description
0x00000000	Remapped Area * REMAP = 00b : Boot-ROM * REMAP = 01b : Internal SRAM * REMAP = 10b : External SDRAM * REMAP = 11b : No-Remap
0x10000000	Internal SRAM * 16KBs Size
0x20000000	Reserved for TLB for virtual MMU Table * Do not use this area
0x40000000 ~ 0x80000000	External SDRAM * Up to 1GBs[8]
0xE0000000	Boot-ROM * 16KBs size
0xF0000000	Special Function Registers * Hardware Control Registers

Figure 2.1 Memory Organization

7 If more detailed information is required, refer to the ARM1176JZFS Technical Reference Manual on ARM site.
8 The maximum size for external SDRAM is determined by the system configuration and the size of external SDRAM.

Table 2.2 Video Controller

Video Codec	Key Features
Encoder	* H.264 Encoding - 24fps @ HD Resolution (1280x720p) * MPEG-4-ASP Encoding - 30fps @ HD Resolution (1280x720p) * H.263 Encoding - 30fps @ HD Resolution (1280x720p)
Decoder	* H.264 Decoding - 30fps @ Full-HD Resolution * MPEG4-ASP Decoding - 30fps @ Full-HD Resolution * H.263 Decoding - 30fps @ Full-HD Resolution * VC-1 Decoding - 30fps @ Full-HD Resolution * RV Decoding - 30fps @ Full-HD Resolution

Table 2.3 Camera Interface

CAMERA I/F	Key Features
Various Input Formats	* CCIR601/656 4:2:2 * Down Scaling for Preview Display : 1/2, 1/4, 1/8 * Change the Image size and windowing. * Support the master clock for camera module.
Hardware Format Converter	* Reconfigurable Packing the Pixel Data * Dispatching the Pixel Data into Y/Cb/Cr * Horizontal and Vertical Window Clipping * Overlaying the Background Frame for Still or Moving Pictures → Chroma-Keying → Alpha-blending (0%, 25%, 50%, 75%, 100%, XOR) * Support the Master Clock for Camera Module → w/o External Oscillator
Maximum Resolutions	* up to 120MHz[9] for Still Image[10]

9 The maximum frequency can be limited by the timing specification of the camera sensor or external device.
10 The maximum resolution can be limited by the system configuration.

Table 2.4 Video Output Interface

Video Output Interface	Key Features
Color TFT LCD	* Various type image sources → RGB565, RGB555, RGB666, RGB24, YCbCr4:2:0, YCbCr4:2:2 * Various type YCbCr4:2:0 and YCbCr4:2:2 to RGB converter * Parallel 24bits and 18bits pixel data output * 6(R):6(G):6(B)bits and 8(R):8(G):8(B)bits pixel data output
Mono/Color STN LCD	* Mono: 1, 2, 4bpp image source * Color: 8(332), 12(444), 555, 565 bpp image source * 4 and 8-bit pixel data interface
NTSC/PAL Encoder Interface	* CCIR601/656 interlace/non-interlace * RGB to YCbCr4:2:2 converter
NTSC/PAL composite output	* Supports all NTSC and PAL formats (NTSC-M/4.43, PAL-B/D/G/H/I/M/N/Combination N)
HDMI Output[11]	* The supported formats are - 1920x1080p @ 60Hz - 1920x1080i @ 30Hz - 1280x720p @ 30Hz - 720x480i @ 60Hz - 720x480p @ 60Hz - etc.
LVDS Output	* 5 Differential Data Lanes * Each data in each lane can be mapped to any pixel data and sync signals.
Image Processing	* OSD/Overlay: can mix up to 3 image sources. - Channel 0 can't mix up for only 1 image channel - Channel 1 has the 3 overlay channels. - Chroma-keying - 256 level Alpha-blending - Contrast/Brightness/Hue Control - Simple Gamma Correction Supported - LUT for each image channels * Virtual Window: Panning / Sliding the Window * Subsampling: 1/2, 1/3, 1/4, 1/8 * Duplication: x2, x3, x4, x8

11 The maximum resolution can be limited by the system configuration.

Table 2.5 DAI/CDIF Controller

I2S (DAI/CDIF)	Key Features
DAI (Digital Audio Interface)	* System clock: 256fs, 384fs, 512fs. * Maximum 7.1 channel supported * Support of Master/Slave Mode with Reconfigurable Clock Polarity * Wide Range of Sampling Frequency in Audio application : 8kHz, 16kHz, 11.05kHz, 24kHz, 32kHz 44.1kHz, 48kHz * Supports the I2S (MSB Justified Mode) * Controls the Digital Audio Volume over the range 0dB to -90dB * Using 2 Double Buffers for Audio I/O Data
CDIF (CD Interface)	* CD Interface for Feasible Implementation of CD Application * Slave Mode * I2S (LSB Justified Mode)

Table 2.6 SPDIF Controller

SPDIF	Key Features
General Features	* Transmitter/Receiver Included * Bit Rate is 64 times the sampling frequency * Configurable 16/24 Bits Mode
Maximum Operating Frequency	* 24MHz Output Data-Rate → SPDIF Clock = 12.288MHz, Ratio = 1 ➔ 3.072Mbps / 48kHz (Data Rate)

Table 2.7 External Device Interface

External Device Interface	Key Features
Static Memory Controller	* Support of 4 Types Static Memory (NAND/IDE/ROM/SRAM) * Controllable Setup / Pulse Width / Hold Time * 8/16 Bits Width

Table 2.8 USB 1.1 Host

USB 1.1 Host	Key Features
General Features	* USB1.1 Host Compatible * OHCI 1.0 Compliant * 2 Down Stream Port - 1 for Dedicated USB1.1 Host Port - 1 for nanoPHY for USB 2.0 OTG Port
PHY Interface	* On-Chip UTMI PHY Serial Interface
Maximum Operating Frequency	* 48MHz

Table 2.9 USB 2.0 OTG(On-The-GO)

USB 2.0 OTG	Key Features
General Feature	* Compliant USB2.0 Specification * Support Interrupt, Bulk Transfer * Support FS/HS dual mode operation * 16bit interface * FIFO size configuration
USB DMA	* 3 Channel DMA (EP1,EP2,EP3) * Support 16/32bit MCU interface * Single / Fly mode * 4x32 FIFO for Each Endpoint
PHY Interface	* On-Chip UTMI PHY Parallel Interface
Maximum Operating Frequency	* 12 External Oscillator (Main Oscillator) * 30MHz with 16bits parallel interface

Table 2.10 nano PHY for USB2.0 OTG and USB1.1 Host

UTMI PHY	Key Features
Supported Specification	* Compliant with USB 2.0 Transceiver Macrocell Interface Spec. Ver-1.04
General Features	* 480Mbps High Speed / 12Mbps Full Speed, FS Only, 1.5Mbps Low Speed * Separate 8/16 bit Unidirectional Parallel Interface * Dual-Mode Device Support (HS/FS) * Data and Clock Recovery from Serial Data on the USB Connector * SYNC/End-Of-Packet Generation and Checking * Bit Stuffing and unstuffing, Bit-stuffing Error Detection * NRZI Encoding/Decoding * Support of Suspend, Resume, Remote Wakeup Operations * Integrated HS and FS Termination and Signaling Switching * On-Chip PLL for 480Mbps * Low Power Dissipation while Active, Idle, or on Standby
System Features	* 45-Ohm Termination / 1.5k Pull-up 15k Pull-down Integrated * Minimal External Components – Single Resistor
Maximum Operating Frequency	* up to 480MHz

Table 2.11 External Host Interface

EHI	Key Features
SRAM Type Interface	* 68/80 Series Interface with 8/16 Bit Width * Burst Transfer and Address Auto-Increment * Internal Interrupt Generation by an External Host Device * Semaphore Register for Improving Data Transfer Efficiency * READY can be Checked via Status Register and Pin. * LOCK MODE: External Host Device can Occupy System bus without any Handover.
Host Booting Procedure	* Configures 8/16 Bits Host Booting Mode * Host Downloads the Program into On-Chip SRAM or Off-Chip Memory * Restarts with Downloaded Program Code
Peak Access Bandwidth	* 8 Bits Configuration : 20MB/s * 16 Bits Configuration : 40MB/s

Table 2.12 SD/MMC Controller

SD/MMC	Key Features
Supported Specification	* SD ver.2.0 * SDIO ver.2.0 * MMC ver.4.3 * Ce-ATA Digital Protocol rev1.1
General Features	* Automatic CRC Generation & Checking the Data/Command * Data transmit/receive FIFO (32bits x 8) * Supported SD/MMC Mode → 1 Bit Serial or 4 Bit Parallel SD → 1 Bit Serial for MMC → 4/8 Bits Parallel Transfer * External DMA Handshaking for Burst & Fast Transfer
Maximum Frequency	* up to 50MHz[12] * Clock generation block Inside

Table 2.13 Memory Stick Controller

Memory Stick	Key Features
Supported Specifications	* Memory Stick Ver.1.x * Memory Stick Pro * Memory Stick Pro-HG
General Features	* Data transmit/receive FIFO (64bits x 4) * External DMA Handshaking for Burst & Fast Transfer
Maximum Operating Frequency	* Memory Stick serial clock (Serial : 20MHz, Parallel : 40MHz)

12 The maximum operating frequency for storage devices can be limited by the system configuration and corresponding interface ports.

Table 2.14 Nand Flash Controller

NAND I/F	Key Features
NAND I/F	* Automatic Detection of External READY Signal * Configurable Cycle Times based-on Bus Frequency * 8bit, 16bit, 32bit Interface to Buffer Memory * 8x32Bits FIFO Included * External DMA Handshaking for Burst and Fast Transfer
External Configuration	* 1 NAND - Single 8 bit NAND / Single 16bit NAND * 2 NAND - Double Series 8 bit NAND / Double Series 16 Bit NAND * 4 NAND - Double Parallel & Series 8 Bit NAND
SLC	* 2 Bit Error Detection & 1 Bit Error Correction per 256 bytes. * Configurable Memory Regions for ECC Calculations
MLC	* 4/8/12/14/16 Bit Error Detection/Correction Based on BCH Algorithm * 8x32 FIFO

Table 2.15 IDE Interface

<table>
<tr><th>HDD (UDMA)</th><th>Key Features</th></tr>
<tr><td>Supported Specification</td><td>* It Supports for the following mode.
- maximum theoretical bandwidths for various operating mode
<table>
<tr><th>PIO MODE</th><th>SPEED</th><th>UDMA MODE</th><th>SPEED</th></tr>
<tr><td>PIO MODE 0</td><td>3.3 MBps</td><td>MODE 0</td><td>16.67 MBps</td></tr>
<tr><td>PIO MODE 1</td><td>5.22 MBps</td><td>MODE 1</td><td>25 MBps</td></tr>
<tr><td>PIO MODE 2</td><td>8.33 MBps</td><td>MODE 2</td><td>33.33 MBps</td></tr>
<tr><td>PIO MODE 3</td><td>11.11 MBps</td><td>MODE 3</td><td>44.44 MBps</td></tr>
<tr><td>PIO MODE 4</td><td>16.67 MBps</td><td>MODE 4</td><td>66.67 MBps</td></tr>
</table></td></tr>
<tr><td>General Features</td><td>* It has a 16x32 bit FIFO that supports internal DMA operation.</td></tr>
<tr><td>Maximum Operating Frequency</td><td>TBD</td></tr>
</table>

Table 2.16 UART Interface

UART	Key Features
General Features	* 16 bytes TX/RX FIFO * Support of Hardware Flow Control (CTS/RTS) * 16 bits clock divider * 16C550 Compatible Core * Smart Card Interface
Maximum Operating Frequency	* Baud Rate Clock (3MHz ← 48MHz / 16)

Table 2.17 GPSB Interface

GPSB	Key Features
General Feature	* MSB / LSB Selection mode * Support Variable transfer (1~16bit) * Clock frequency/ Polarity selection mode * Support configurable Frame signal mode
Maximum Operating Frequency	* 60MHz for slave only * 30MHz for master mode

Table 2.18 General DMA Controller

General DMA	Key Features
General Features	* 12-Channel DMA * Dedicated Bus Interface for Various Storage Interface Controllers * Support of Byte/Half-word/Word Transfer * Support of Circular Buffer Interface → Masking of the Source/Destination Address Bits * 1/2/4/8 Burst Transfers * Byte Swapping Function * Support of Single/Continuous/Burst Mode * 8x32bits FIFO Included
DMA Request/Acknowledge	* External DMA Request/Acknowledge * Interfacing the On-chip Storage Controllers
Inter-channel Arbitration	* Configurable Priority for Each Channel * Round-robin Arbitration / Fixed Priority Arbitration

Table 2.19 Vectored Interrupt Controller

VPIC	Key Features
General Features	* 64 Individual Interrupt Sources * FIQ/IRQ Configurable * Priority Reconfigurable for Each Interrupt Sources * Polarity Controllable * Edge/Level Sensitivity Controllable * Dual/Single Edge Controllable when Edge Sensitivity Selected
Vectored Interrupt Handler	* Vector ID Returned for Fast Handling * Vector ID is one of 0 ~ 63 * 64x32 Vector Table Needed for Vector Handler on On-Chip Memory

Table 2.20 Timer

TIMER & WDT	Key Features
Timer Counters	* Four 16-bit timers with PWM output/counters, two 20-bit timers, and one 32-bit timer * External Event Counter * Stop Mode / Free Running Mode * Various Clock Sources (PLL outputs ~ Divided Sub-Clock) * PWM Functions → TREFn, TMREFn
Watchdog Timers	* Watchdog Timer Interrupt / Reset

Table 2.21 ADC

TSADC	Key Features
General Features	* 12bit Resolution * 0 ~ 3.3V Input Range (In case of 3.3V AVDD)
Operating Frequency	* 1MSPS / 5MHz
Touch Screen	* X/Y Position * Up/Down Wake-up

Table 2.22 Real Time Clock

RTC	Key Features
General Features	* Sub Oscillator Included * Clock and Calendar Function (BCD Display) → Sec/Min/Hour/Date/Day-of-Week/Month/Year * Leap Year Generation * Wakeup Signal Generation from the Deep Power-down Mode
Interrupt & Round Reset	* Alarm Interrupt in Normal Operation Mode * Cyclic interrupts 1/256, 1/64, 1/16, 1/4, 1/2, 1 second interrupts * Round-reset function 30-, 40-, 50- second
Wakeup Function	* Dedicated Wake-up Port

3 PIN Description

3.1 TCC8900 Pin Description

Table 3.1 Power/Ground Information

Group	# of Balls	Ball#	MIN(V)	TYP(V)	MAX(V)	Description
GNDCORE	17	F4, J5, F7, L7, T9, E10, E12, U12, K13, M13, N13, R13, K14, J15, P15, R15, K16	-	-	-	Internal Core Ground
PWRCORE[13]	15	H4, J4, F6, R8, G9, G11, G12, P13, W13, H14, J14, L15, N15, T15, M4	1.14	1.20	1.26	Digital Internal Core Power @ F_{CPU} ≤ 500MHz
GNDIO	9	P4, H5, F8, E11, T11, V11, F12, E14, F15	-	-	-	I/O Ground
PWRGPIOA	1	V13	1.71 2.38 3.14	1.8 2.5 3.3	1.89 2.62 3.46	GPIOA Group I/O Power
PWRGPIOB	2	F11, F9				GPIOB Group I/O Power
PWRGPIOC	2	E6, E9				GPIOC Group I/O Power
PWRGPIOD	2	F13, F14				GPIOD Group I/O Power
PWRGPIOE	2	V10, V12				GPIOE Group I/O Power
PWRGPIOF	2	G4, P5				GPIOF Group I/O Power
PWRETC	1	G14	1.8		3.6	ETC Group I/O Power
PWRMEMQ	6	J13, L14, M14, M15, N14, P14	1.71 2.38 3.14	1.8 2.5 3.3	1.89 2.62 3.46	SDR/DDR/mDDR I/O Power
PWRMEMZQ	1	J12				DDR2 ZQ Calibration Power
GNDMEMZQ	1	H12	-	-	-	SDR/DDR/mDDR/DDR2 I/F Ground
PWROSC	1	L6	2.7	3.0	3.3	Oscillator Power
GNDOSC	1	M6	-	-	-	Oscillator Ground
PWRUSB1213	1	L4	1.14	1.20	1.26	UTMI Digital Core Power
PWRUSB33	1	L5	3.0	3.3	3.6	UTMI Analog Core Power
GNDUSB	1	K1	-	-	-	UTMI Ground
PWRUSBH	1	B11				USB 1.1 Host Transceiver Power
GNDUSBH	1	B10				USB 1.1 Host Transceiver Ground
PWRLVDS33A	2	P7, R7				LVDS Transmitter Power
GNDLVDS33A	3	T6, W5, Y5				LVDS Transmitter Ground
PWRSATA1	1	R4	1.14	1.20	1.26	SATA Core Power 1
PWRSATA2	1	P3	1.14	1.20	1.26	SATA Core Power 2
GNDSATA	2	M1, R1				SATA Core Ground
PWRSATAOSC	1	U2	3.0	3.3	3.6	SATA Oscillator Power
GNDSATAOSC	1	V1				SATA Oscillator Ground
PWRSATAPLL	1	P2	1.14	1.20	1.26	SATA PLL Power
GNDSATAPLL	1	R2				SATA PLL Ground
PWRHDMI	1	T5	1.14	1.20	1.26	HDMI Core Power
GNDHDMI	3	V2, V3, U6				HDMI Core Ground
PWRHDMIPLL1	1	R5	1.14	1.20	1.26	HDMI PLL Power 1
PWRHDMIPLL2	1	T4	1.14	1.20	1.26	HDMI PLL Power 2
GNDHDMIPLL	1	R6				HDMI PLL Ground
PWRHDMIOSC	1	U5	3.0	3.3	3.6	HDMI Oscillator Power
GNDHDMIOSC	1	V5				HDMI Oscillator Ground
PWRADC	1	T13	2.7[14]	3.3	3.6	ADC Power
GNDADC	1	T14	-	-	-	ADC Ground
PWRDAC	1	U9	2.7	3.0	3.3	DAC Power
GNDDAC	2	T8, U7	-	-	-	DAC Ground
PWRPLL13	1	N5	1.14	1.20	1.26	PLL Power
GNDPLL	1	M3	-	-	-	PLL Ground
PWRRTC	1	N4	1.5	3.0	3.3	RTC Core & I/O Power

13 Refer to Table 5.1
14 PWRADC : When PWRADC is 2.7~3.0V, note that ADC error rate is increased.

Table 3.2 PWRGPIOC Group I/O Pin Description

NAME	BALL	I/O [15]	INIT [16]	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6
GPIOC[0]	A9	B	IU	GPIOC[0]	LXD[0]	L0_LPD[0]	TSD5(3)		L1_LPD[0]	GPIOF[0]
GPIOC[1]	A8	B	IU	GPIOC[1]	LXD[1]	L0_LPD[1]	TSD6(3)		L1_LPD[1]	GPIOF[1]
GPIOC[2]	B9	B	IU	GPIOC[2]	LXD[2]	L0_LPD[2]	TSD7(3)		L1_LPD[2]	GPIOF[2]
GPIOC[3]	C9	B	IU	GPIOC[3]	LXD[3]	L0_LPD[3]			L1_LPD[3]	GPIOF[3]
GPIOC[4]	B8	B	IU	GPIOC[4]	LXD[4]	L0_LPD[4]			L1_LPD[4]	GPIOF[4]
GPIOC[5]	D9	B	IU	GPIOC[5]	LXD[5]	L0_LPD[5]			L1_LPD[5]	GPIOF[5]
GPIOC[6]	E8	B	IU	GPIOC[6]	LXD[6]	L0_LPD[6]	SDO(3)		L1_LPD[6]	GPIOF[6]
GPIOC[7]	A7	B	IU	GPIOC[7]	LXD[7]	L0_LPD[7]	SDI(3)		L1_LPD[7]	GPIOF[7]
GPIOC[8]	C8	B	I	GPIOC[8]	LXD[8]	L0_LPD[8]	SCLK(3)		L1_LPD[8]	GPIOF[8]
GPIOC[9]	D8	B	I	GPIOC[9]	LXD[9]	L0_LPD[9]	SFRM(3)		L1_LPD[9]	GPIOF[9]
GPIOC[10]	B7	B	I	GPIOC[10]	LXD[10]	L0_LPD[10]	SDO(2)		L1_LPD[10]	GPIOF[10]
GPIOC[11]	A6	B	I	GPIOC[11]	LXD[11]	L0_LPD[11]	SDI(2)		L1_LPD[11]	GPIOF[11]
GPIOC[12]	C7	B	I	GPIOC[12]	LXD[12]	L0_LPD[12]	SCLK(2)		L1_LPD[12]	GPIOF[12]
GPIOC[13]	D7	B	I	GPIOC[13]	LXD[13]	L0_LPD[13]	SFRM(2)		L1_LPD[13]	GPIOF[13]
GPIOC[14]	E7	B	I	GPIOC[14]	LXD[14]	L0_LPD[14]	SD_D7(0)	MS_D7(0)	L1_LPD[14]	GPIOF[14]
GPIOC[15]	C6	B	I	GPIOC[15]	LXD[15]	L0_LPD[15]	SD_D6(0)	MS_D6(0)	L1_LPD[15]	GPIOF[15]
GPIOC[16]	B6	B	I	GPIOC[16]	LXD[16]	L0_LPD[16]	SD_D5(0)	MS_D5(0)	L1_LPD[16]	GPIOF[16]
GPIOC[17]	A5	B	I	GPIOC[17]	LXD[17]	L0_LPD[17]	SD_D4(0)	MS_D4(0)	L1_LPD[17]	GPIOF[17]
GPIOC[18]	A4	B	IU	GPIOC[18]	LXD[18]	L0_LPD[18]	SD_D3(0)	MS_D3(0)	L1_LPD[18]	
GPIOC[19]	B5	B	IU	GPIOC[19]	LXD[19]	L0_LPD[19]	SD_D2(0)	MS_D2(0)	L1_LPD[19]	
GPIOC[20]	D6	B	IU	GPIOC[20]	LXD[20]	L0_LPD[20]	SD_D1(0)	MS_D1(0)	L1_LPD[20]	
GPIOC[21]	C5	B	IU	GPIOC[21]	LXD[21]	L0_LPD[21]	SD_D0(0)	MS_D0(0)	L1_LPD[21]	
GPIOC[22]	B4	B	I	GPIOC[22]	LXD[22]	L0_LPD[22]	SD_CLK(0)	MS_CLK(0)	L1_LPD[22]	
GPIOC[23]	A3	B	I	GPIOC[23]	LXD[23]	L0_LPD[23]	SD_CMD(0)	MS_BUS(0)	L1_LPD[23]	
GPIOC[24]	D5	B	IU	GPIOC[24]	LWEN	L0_LDE	TSD4(3)		L1_LDE	GPIOF[19]
GPIOC[25]	C4	B	IU	GPIOC[25]	LOEN	L0_LCK	TSD3(3)		L1_LCK	GPIOF[18]
GPIOC[26]	B3	B	I	GPIOC[26]	LXA[0]	L0_LHS	TSD2(3)		L1_LHS	GPIOF[22]
GPIOC[27]	E5	B	IU	GPIOC[27]	LCSN0	L0_LVS	TSD1(3)		L1_LVS	GPIOF[20]
GPIOC[28]	A2	B	IU	GPIOC[28]	LCSN1	SDO(10)	TSVALID(3)			GPIOF[21]
GPIOC[29]	C3	B	I	GPIOC[29]		SDI(10)	TSCLK(3)			
GPIOC[30]	D4	B	I	GPIOC[30]	EXTLVS0(0)	SCLK(10)	TSD0(3)			
GPIOC[31]	B2	B	I	GPIOC[31]	EXTLVS1(0)	SFRM(10)	TSSYNC(3)			

15 I/O type. B = bi-direction, I = Input, O = Output
16 INIT column represents the corresponding I/O state while the reset signal is asserted.
OL = output low, OH =output high, O=output unknown, IU = input/pull-up, ID = input /pull-down, I = input floating

Table 3.3 PWRGPIOF Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6
BPEN	F5	B	I	BPEN	BPEN	BPEN	BPEN	BPEN	BPEN	BPEN
GPIOF[0]	A1	B	I	GPIOF[0]	HPXD[0]	SD_D0(3)	HDDXD0	TSD0(0)	MS_D0(3)	EDIXA[3]
GPIOF[1]	D3	B	I	GPIOF[1]	HPXD[1]	SD_D1(3)	HDDXD1	TSD1(0)	MS_D1(3)	EDIXA[4]
GPIOF[2]	B1	B	I	GPIOF[2]	HPXD[2]	SD_D2(3)	HDDXD2	TSD2(0)	MS_D2(3)	EDIXA[5]
GPIOF[3]	E4	B	I	GPIOF[3]	HPXD[3]	SD_D3(3)	HDDXD3	TSD3(0)	MS_D3(3)	EDIXA[6]
GPIOF[4]	C2	B	I	GPIOF[4]	HPXD[4]	SD_D4(3)	HDDXD4	TSD4(0)	MS_D4(3)	EDIXA[7]
GPIOF[5]	C1	B	I	GPIOF[5]	HPXD[5]	SD_D5(3)	HDDXD5	TSD5(0)	MS_D5(3)	EDIXA[8]
GPIOF[6]	D2	B	I	GPIOF[6]	HPXD[6]	SD_D6(3)	HDDXD6	TSD6(0)	MS_D6(3)	EDIXA[9]
GPIOF[7]	E3	B	I	GPIOF[7]	HPXD[7]	SD_D7(3)	HDDXD7	TSD7(0)	MS_D7(3)	EDIXA[10]
GPIOF[8]	E2	B	I	GPIOF[8]	HPXD[8]	SD_CMD(3)	HDDXD8	TSVALID(0)	MS_BUS(3)	EDIXA[11]
GPIOF[9]	F3	B	I	GPIOF[9]	HPXD[9]	SD_CLK(3)	HDDXD9	TSCLK(0)	MS_CLK(3)	EDIXA[12]
GPIOF[10]	F2	B	I	GPIOF[10]	HPXD[10]	SDO(7)	HDDXD10	TSSYNC(0)		EDIXA[13]
GPIOF[11]	D1	B	I	GPIOF[11]	HPXD[11]	SDI(7)	HDDXD11			EDIXA[14]
GPIOF[12]	E1	B	I	GPIOF[12]	HPXD[12]	SCLK(7)	HDDXD12			EDIXA[15]
GPIOF[13]	G3	B	I	GPIOF[13]	HPXD[13]	SFRM(7)	HDDXD13			EDIXA[16]
GPIOF[14]	F1	B	I	GPIOF[14]	HPXD[14]	SDO(8)	HDDXD14			EDIXA[17]
GPIOF[15]	G2	B	I	GPIOF[15]	HPXD[15]	SDI(8)	HDDXD15			EDIXA[18]
GPIOF[16]	G1	B	I	GPIOF[16]	HPXD[16]	SCLK(8)	HDDXA0			
GPIOF[17]	H2	B	I	GPIOF[17]	HPXD[17]	SFRM(8)	HDDXA1			
GPIOF[18]	H3	B	I	GPIOF[18]	HPRDN	SD_D3(1)	HDDXA2	MS_D3(1)		
GPIOF[19]	J2	B	I	GPIOF[19]	HPWRN	SD_D2(1)	HDDCSN1	MS_D2(1)		
GPIOF[20]	J3	B	I	GPIOF[20]	HPCSN0	SD_D1(1)	HDDRDY	MS_D1(1)		
GPIOF[21]	J6	B	I	GPIOF[21]	HPCSN1	SD_D0(1)	HDDCSN0	MS_D0(1)		
GPIOF[22]	K7	B	I	GPIOF[22]	HPXA	SD_CMD(1)	HDDAK	MS_BUS(1)		
GPIOF[23]	K2	B	I	GPIOF[23]	HPINT0	SD_CLK(1)	HDDRQ	MS_CLK(1)		
GPIOF[24]	K4	B	I	GPIOF[24]	HPINT1	SDO(9)	HDDIOW			
GPIOF[25]	K3	B	I	GPIOF[25]		SDI(9)	HDDIOR			
GPIOF[26]	K6	B	I	GPIOF[26]	EXTLVS1(1)	SCLK(9)				
GPIOF[27]	K5	B	I	GPIOF[27]	EXTLVS0(1)	SFRM(9)				

Table 3.4 PWRGPIOE Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6
GPIOE[0]	N7	B	IU	GPIOE[0]	UTXD(0)					
GPIOE[1]	P8	B	IU	GPIOE[1]	URXD(0)					
GPIOE[2]	U10	B	IU	GPIOE[2]	UCTS(0)	SFRM(5)				
GPIOE[3]	M7	B	IU	GPIOE[3]	URTS(0)	SCLK(5)				
GPIOE[4]	L8	B	IU	GPIOE[4]	UTXD(1)					
GPIOE[5]	N8	B	IU	GPIOE[5]	URXD(1)					
GPIOE[6]	R9	B	IU	GPIOE[6]	UCTS(1)	SDI(5)	SD_CLK(4)	MS_CLK(4)		
GPIOE[7]	U11	B	IU	GPIOE[7]	URTS(1)	SDO(5)	SD_CMD(4)	MS_BUS(4)		
GPIOE[8]	W11	B	I	GPIOE[8]	UTXD(2)	SFRM(4)	SD_D0(4)	MS_D0(4)		
GPIOE[9]	Y11	B	I	GPIOE[9]	URXD(2)	SCLK(4)	SD_D1(4)	MS_D1(4)		
GPIOE[10]	Y12	B	I	GPIOE[10]	UTXD(3)	SDI(4)	SD_D2(4)	MS_D2(4)		
GPIOE[11]	T10	B	I	GPIOE[11]	URXD(3)	SDO(4)	SD_D3(4)	MS_D3(4)		
GPIOE[12]	Y13	B	I	GPIOE[12]	CPD[0]	SD_D0(2)	TSD0(1)	MS_D0(2)		
GPIOE[13]	P9	B	I	GPIOE[13]	CPD[1]	SD_D1(2)	TSD1(1)	MS_D1(2)		
GPIOE[14]	Y14	B	I	GPIOE[14]	CPD[2]	SD_D2(2)	TSD2(1)	MS_D2(2)		
GPIOE[15]	M8	B	I	GPIOE[15]	CPD[3]	SD_D3(2)	TSD3(1)	MS_D3(2)		
GPIOE[16]	Y15	B	IU	GPIOE[16]	CPD[4]	SD_D4(2)	TSD4(1)	MS_D4(2)		
GPIOE[17]	L9	B	IU	GPIOE[17]	CPD[5]	SD_D5(2)	TSD5(1)	MS_D5(2)		
GPIOE[18]	W12	B	IU	GPIOE[18]	CPD[6]	SD_D6(2)	TSD6(1)	MS_D6(2)		
GPIOE[19]	R10	B	IU	GPIOE[19]	CPD[7]	SD_D7(2)	TSD7(1)	MS_D7(2)		
GPIOE[20]	Y16	B	IU	GPIOE[20]	CCKI	SD_CLK(2)	TSCLK(1)	MS_CLK(2)		
GPIOE[21]	P10	B	IU	GPIOE[21]	CVS	SD_CMD(2)	TSSYNC(1)	MS_BUS(2)		
GPIOE[22]	Y17	B	IU	GPIOE[22]	CHS		TSVALID(1)			
GPIOE[23]	R11	B	IU	GPIOE[23]	CCKO	CFIELD				

Table 3.5 PWRGPIOA Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6
GPIOA[0]	Y18	B	IU	GPIOA[0]	SCL0					
GPIOA[1]	T12	B	IU	GPIOA[1]	SDA0					
GPIOA[2]	W14	B	I	GPIOA[2]	CLK_OUT0					
GPIOA[3]	N9	B	I	GPIOA[3]	CLK_OUT1					
GPIOA[4]	W15	B	IU	GPIOA[4]	WDTRSTO	TCO0				
GPIOA[5]	M9	B	IU	GPIOA[5]	IRDI	TCO1				
GPIOA[6]	W16	B	IU	GPIOA[6]	HDMI_CECO	TCO2				EDIXA[19]
GPIOA[7]	N10	B	IU	GPIOA[7]	HDMI_CECI	TCO3				EDIXA[20]
GPIOA[8]	V15	B	IU	GPIOA[8]	SCL1					
GPIOA[9]	M10	B	IU	GPIOA[9]	SDA1					
GPIOA[10]	U14	B	I	GPIOA[10]	CBCLK(0)	CBCLK(1)				
GPIOA[11]	U13	B	I	GPIOA[11]	CLRCK(0)	CLRCK(1)				
GPIOA[12]	V14	B	I	GPIOA[12]	CDATA(0)	CDATA(1)				
GPIOA[13]	P11	B	I	GPIOA[13]	EXTCLKI					
GPIOA[14]	N11	B	I	GPIOA[14]	HDMI_HPD	TCO4				
GPIOA[15]	R12	B	I	GPIOA[15]	UTM_DRVVBUS	TCO5				

Table 3.6 PWRADC Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6
AIN[0]	V16	B	I	GPIOE[24]	AIN[0]					
AIN[1]	U15	B	I	GPIOE[25]	AIN[1]					
AIN[2]	W17	B	I	GPIOE[26]	AIN[2]	SD_CMD(7)	MS_BUS(7)			
AIN[3]	W18	B	I	GPIOE[27]	AIN[3]	SD_CLK(7)	MS_CLK(7)			
AIN[4]	Y19	B	I	GPIOE[28]	TSC_YM	SD_D0(7)	MS_D0(7)			
AIN[5]	V17	B	I	GPIOE[29]	TSC_YP	SD_D1(7)	MS_D1(7)			
AIN[6]	U16	B	I	GPIOE[30]	TSC_XM	SD_D2(7)	MS_D2(7)			
AIN[7]	R14	B	I	GPIOE[31]	TSC_XP	SD_D3(7)	MS_D3(7)			

Table 3.7 PWRETC Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6
BM[0]	H16	I	I	BM[0]						
BM[1]	G15	I	I	BM[1]						
BM[2]	C20	I	I	BM[2]						
TDO	B20	O	O	TDO						
TEST	B19	I	I	TEST						
RSTN	C19	I	I	RSTN						
NTRST	E17	I	I	NTRST						
TMS	D17	I	IU	TMS						
TCK	F16	I	IU	TCK						
TDI	E16	I	IU	TDI						
RTCK	E15	O	O	RTCK						

Table 3.8 PWRGPIOD Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6
GPIOD[0]	A20	B	IU	GPIOD[0]	BCLK(1)	BCLK(0)				
GPIOD[1]	D16	B	IU	GPIOD[1]	LRCK(1)	LRCK(0)				
GPIOD[2]	A19	B	IU	GPIOD[2]	MCLK(1)	MCLK(0)				
GPIOD[3]	C17	B	IU	GPIOD[3]	DAO0(1)	DAO0(0)				
GPIOD[4]	C18	B	IU	GPIOD[4]	DAI0(1)	DAI0(0)				
GPIOD[5]	A18	B	IU	GPIOD[5]	DAO1(1)	SFRM(11)				
GPIOD[6]	B18	B	IU	GPIOD[6]	DAI1(1)	SCLK(11)				
GPIOD[7]	A17	B	IU	GPIOD[7]	DAO2(1)	SDI(11)				
GPIOD[8]	G13	B	I	GPIOD[8]	DAI2(1)	SDO(11)				
GPIOD[9]	D15	B	I	GPIOD[9]	DAO3(1)	SFRM(6)	TSD7(2)			
GPIOD[10]	H13	B	I	GPIOD[10]	DAI3(1)	SCLK(6)	TSD6(2)			
GPIOD[11]	C16	B	I	GPIOD[11]	SPD_TX(1)	SDI(6)	SPD_TX(0)			
GPIOD[12]	L11	B	I	GPIOD[12]	SPD_RX(1)	SDO(6)	TSSYNC(2)			
GPIOD[13]	B17	B	I	GPIOD[13]	UTXD(4)		TSD5(2)			
GPIOD[14]	L10	B	I	GPIOD[14]	URXD(4)		TSD4(2)			
GPIOD[15]	C15	B	I	GPIOD[15]	UCTS(4)	SFRM(12)	TSVALID(2)			
GPIOD[16]	K11	B	IU	GPIOD[16]	URTS(4)	SCLK(12)	TSCLK(2)			
GPIOD[17]	D14	B	IU	GPIOD[17]	UTXD(5)		TSD3(2)			
GPIOD[18]	J11	B	IU	GPIOD[18]	URXD(5)		TSD2(2)			
GPIOD[19]	B16	B	IU	GPIOD[19]	UCTS(5)	SDI(12)	TSD1(2)			
GPIOD[20]	K10	B	IU	GPIOD[20]	URTS(5)	SDO(12)	TSD0(2)			
GPIOD[21]	A16	B	IU	GPIOD[21]						
GPIOD[22]	K8	B	IU	GPIOD[22]						
GPIOD[23]	C14	B	IU	GPIOD[23]						
GPIOD[24]	K9	B	I	GPIOD[24]						
GPIOD[25]	D13	B	I	GPIOD[25]						

Table 3.9 PWRGPIOB Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6
GPIOB[0]	J7	B	IU	GPIOB[0]	EDIXD8	SD_D0(5)	MS_D0(5)			
GPIOB[1]	B15	B	IU	GPIOB[1]	EDIXD9	SD_D1(5)	MS_D1(5)			
GPIOB[2]	J10	B	IU	GPIOB[2]	EDIXD10	SD_D2(5)	MS_D2(5)			
GPIOB[3]	E13	B	IU	GPIOB[3]	EDIXD11	SD_D3(5)	MS_D3(5)			
GPIOB[4]	J8	B	IU	GPIOB[4]	EDIXD4	SD_D4(5)	MS_D4(5)			
GPIOB[5]	B14	B	IU	GPIOB[5]	EDIXD5	SD_D5(5)	MS_D5(5)			
GPIOB[6]	J9	B	IU	GPIOB[6]	EDIXD6	SD_D6(5)	MS_D6(5)			
GPIOB[7]	C13	B	IU	GPIOB[7]	EDIXD7	SD_D7(5)	MS_D7(5)			
GPIOB[8]	H11	B	I	GPIOB[8]	EDIXD0			SFRM(1)		
GPIOB[9]	A15	B	I	GPIOB[9]	EDIXD1			SCLK(1)		
GPIOB[10]	H10	B	I	GPIOB[10]	EDIXD2			SDI(1)		
GPIOB[11]	D12	B	I	GPIOB[11]	EDIXD3			SDO(1)		
GPIOB[12]	H6	B	I	GPIOB[12]	EDIXD12	SD_CMD(5)	MS_BUS(5)			
GPIOB[13]	B13	B	I	GPIOB[13]	EDIXD13	SD_CLK(5)	MS_CLK(5)			
GPIOB[14]	F10	B	I	GPIOB[14]	EDIXD14					
GPIOB[15]	A14	B	I	GPIOB[15]	EDIXD15					
GPIOB[16]	G10	B	I	GPIOB[16]	EDIWEN0					
GPIOB[17]	A13	B	I	GPIOB[17]	EDIWEN1[17]			SFRM(0)		
GPIOB[18]	H7	B	I	GPIOB[18]	EDIOEN0					
GPIOB[19]	A12	B	I	GPIOB[19]	EDIOEN1[18]			SCLK(0)		
GPIOB[20]	H9	B	I	GPIOB[20]	EDIXA[0][19]					
GPIOB[21]	D11	B	I	GPIOB[21]	EDIXA[1][20]	SD_D4(6)	MS_D4(6)			
GPIOB[22]	H8	B	IU	GPIOB[22]	EDIXA[2]	SD_D5(6)	MS_D5(6)			
GPIOB[23]	C12	B	IU	GPIOB[23]	EDICSN0[21]	SD_D6(6)	MS_D6(6)			
GPIOB[24]	G5	B	IU	GPIOB[24]	EDICSN1	SD_D7(6)	MS_D7(6)			
GPIOB[25]	B12	B	IU	GPIOB[25]	EDICSN2	SD_D0(6)	MS_D0(6)			
GPIOB[26]	G6	B	IU	GPIOB[26]	EDICSN3	SD_D1(6)	MS_D1(6)			
GPIOB[27]	C11	B	IU	GPIOB[27]	EDICSN4	SD_D2(6)	MS_D2(6)			EDIXA[23]
GPIOB[28]	G7	B	I	GPIOB[28]	EDIRDY0[22]	SD_D3(6)	MS_D3(6)			
GPIOB[29]	D10	B	I	GPIOB[29]	EDIRDY1	SD_CMD(6)	MS_BUS(6)			
GPIOB[30]	G8	B	IU	GPIOB[30]	EDICSN5	SD_CLK(6)	MS_CLK(6)	SDI(0)		EDIXA[22]
GPIOB[31]	C10	B	IU	GPIOB[31][22]	EDICSN6			SDO(0)		EDIXA[21]

17 Dedicated to NAND WEN signal for NAND booting
18 Dedicated to NAND OEN signal for NAND booting
19 Dedicated to NAND CLE signal for NAND booting
20 Dedicated to NAND ALE signal for NAND booting
21 Dedicated to NAND CSN0 signal for NAND booting
22 Dedicated to NAND Ready signal for NAND booting, Either EDIRDY0 or GPIOB[31] can be used for NAND ready signal as described in the BOOT PROCEDURE in datasheet.

Table 3.10 PWRMEMQ Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0 (DDR2-2CS)	FUNC1 (DDR2-1CS)	FUNC2 (DDR/mDDR)	FUNC3 (SDR)	FUNC4	FUNC5	FUNC6
DRAM_XD28	W19			XD28	XD28	XD28	XD28			
DRAM_XD27	U17			XD27	XD27	XD27	XD27			
DRAM_XD25	T17			XD25	XD25	XD25	XD25			
DRAM_XD30	Y20			XD30	XD30	XD30	XD30			
DRAM_DQM3	V18			DQM3	DQM3	DQM3	DQM3			
DRAM_DQS3	U18			DQS3	DQS3	DQS3				
DRAM_DQSB3	T18			DQSB3	DQSB3	-				
DRAM_XD31	U19			XD31	XD31	XD31	XD31			
DRAM_XD24	V19			XD24	XD24	XD24	XD24			
DRAM_XD26	W20			XD26	XD26	XD26	XD26			
DRAM_XD29	V20			XD29	XD29	XD29	XD29			
DRAM_XD20	R16			XD20	XD20	XD20	XD20			
DRAM_XD19	T19			XD19	XD19	XD19	XD19			
DRAM_XD17	U20			XD17	XD17	XD17	XD17			
DRAM_XD22	T20			XD22	XD22	XD22	XD22			
DRAM_DQM2	R17			DQM2	DQM2	DQM2	DQM2			
DRAM_DQS2	R18			DQS2	DQS2	DQS2				
DRAM_DQSB2	P18			DQSB2	DQSB2	-				
DRAM_XD23	R19			XD23	XD23	XD23	XD23			
DRAM_XD16	R20			XD16	XD16	XD16	XD16			
DRAM_XD18	P17			XD18	XD18	XD18	XD18			
DRAM_XD21	P19			XD21	XD21	XD21	XD21			
DRAM_CLKB	P20			CLKB	CLKB	CLKB	-			
DRAM_CLK	N20			CLK	CLK	CLK	CLK			
DRAM_XA6	M20			XA6	XA6	XA6	XA6			
DRAM_XA12	P12			XA12	XA12	XA12	XA12			
DRAM_XA11	N19			XA11	XA11	XA11	XA11			
DRAM_RASN	N18			RASN	RASN	RASN	RASN			
DRAM_XA0	N12			XA0	XA0	XA0	XA0			
DRAM_XA5	N17			XA5	XA5	XA5	XA5			
DRAM_XA10	N16			XA10	XA10	XA10	XA10			
DRAM_CASN	M11			CASN	CASN	CASN	CASn			
DRAM_CSN1	M17			CSN1	**XA13**	CSN1	CSN1			
DRAM_XA2	M18			XA2	XA2	XA2	XA2			
DRAM_ODT0	M12			ODT0	ODT0		XA14			
DRAM_CSN0	M19			CSN0	CSN0	CSN0	CSN0			
DRAM_BA0	L19			BA0	BA0	BA0	BA0			
DRAM_BA2	L18			BA2	BA2	XA13	XA13			
DRAM_XA3	M16			XA3	XA3	XA3	XA3			
DRAM_BA1	L17			BA1	BA1	BA1	BA1			
DRAM_XA1	L13			XA1	XA1	XA1	XA1			
DRAM_XA4	L16			XA4	XA4	XA4	XA4			
DRAM_XA7	K19			XA7	XA7	XA7	XA7			
DRAM_ODT1	L12			ODT1	**XA14**	CKE1	CKE1			
DRAM_XA9	J20			XA9	XA9	XA9	XA9			
DRAM_CKE	K18			CKE	CKE	CKE0	CKE0			
DRAM_XA8	K12			XA8	XA8	XA8	XA8			
DRAM_WEN	K17			WEN	WEN	WEN	WEN			
DRAM_XD12	H20			XD12	XD12	XD12	XD12			
DRAM_XD11	G20			XD11	XD11	XD11	XD11			
DRAM_XD9	F20			XD9	XD9	XD9	XD9			
DRAM_XD14	H19			XD14	XD14	XD14	XD14			
DRAM_DQM1	J19			DQM1	DQM1	DQM1	DQM1			
DRAM_DQS1	J18			DQS1	DQS1	DQS1				
DRAM_DQSB1	J17			DQSB1	DQSB1	-				
DRAM_XD15	G19			XD15	XD15	XD15	XD15			
DRAM_XD8	J16			XD8	XD8	XD8	XD8			
DRAM_XD10	H17			XD10	XD10	XD10	XD10			
DRAM_XD13	H18			XD13	XD13	XD13	XD13			
DRAM_XD4	E20			XD4	XD4	XD4	XD4			
DRAM_XD3	F19			XD3	XD3	XD3	XD3			
DRAM_XD1	D20			XD1	XD1	XD1	XD1			
DRAM_XD6	E19			XD6	XD6	XD6	XD6			
DRAM_DQM0	F18			DQM0	DQM0	DQM0	DQM0			
DRAM_DQS0	G18			DQS0	DQS0	DQS0				
DRAM_DQSB0	G17			DQSB0	DQSB0	-				
DRAM_XD7	D19			XD7	XD7	XD7	XD7			
DRAM_XD0	D18			XD0	XD0	XD0	XD0			
DRAM_XD2	E18			XD2	XD2	XD2	XD2			
DRAM_XD5	F17			XD5	XD5	XD5	XD5			
DRAM_VREF0	H15			VREF0	VREF0					
DRAM_VREF1	K15			VREF1	VREF1					
DRAM_VREF2	P16			VREF2	VREF2					
DRAM_VREF3	T16			VREF3	VREF3					
DRAM_GATEI	L20			GATEI	GATEI	GATEI				
DRAM_GATEO	K20			GATEO	GATEO	GATEO				
DRAM_ZQ	G16			ZQ	ZQ	ZQ				

Table 3.11 PWROSC Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6
XI	L2	I		XI						
XO	L1	O		XO						

Table 3.12 PWRUSB33 Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6
OTG_VBUS	L3			OTG_VBUS						
OTG_DP	H1			OTG_DP						
OTG_DM	J1			OTG_DM						
OTG_REXT	M2			OTG_REXT						

Table 3.13 PWRUSBH Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6
UBH_DP	A11			UBH_DP						
UBH_DM	A10			UBH_DM						

Table 3.14 PWRRTC Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0						
RTC_XTI	N3	I		RTC_XTI						
RTC_XTO	N2	O		RTC_XTO						
RTC_PMWKUP	M5	O	OH	RTC_PMWKUP						
RTC_RSTN	N6	I	IL	RTC_RSTN						

Table 3.15 PWRSATAOSC Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0						
SATA_XI	R3	I		SATA_XI						
SATA_XO	T3	O		SATA_XO						

Table 3.16 PWRSATA Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0						
SATA_REXT	T2			SATA_REXT						
SATA_TXP	U1			SATA_TXP						
SATA_TXN	T1			SATA_TXN						
SATA_RXP	N1			SATA_RXP						
SATA_RXN	P1			SATA_RXN						

Table 3.17 PWRHDMIOSC Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0						
HDMI_XI	U4			HDMI_XI						
HDMI_XO	U3			HDMI_XO						

Table 3.18 PWRHDMI Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0						
HDMI_TXCP	Y1			HDMI_TXCP						
HDMI_TXCN	W1			HDMI_TXCN						
HDMI_TX0P	Y2			HDMI_TX0P						
HDMI_TX0N	W2			HDMI_TX0N						
HDMI_TX1P	Y3			HDMI_TX1P						
HDMI_TX1N	W3			HDMI_TX1N						
HDMI_TX2P	Y4			HDMI_TX2P						
HDMI_TX2N	W4			HDMI_TX2N						
HDMI_REXT	V4			HDMI_REXT						

Table 3.19 PWRLVDS Group I/O Pin Description

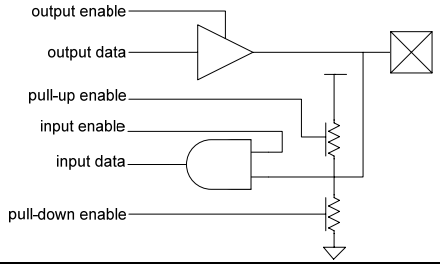
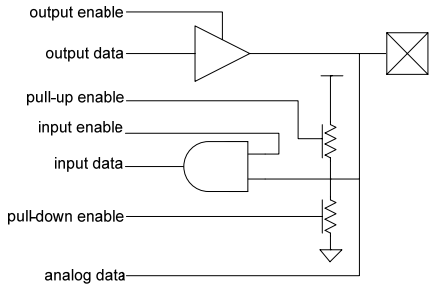
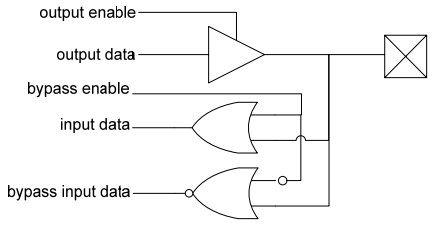
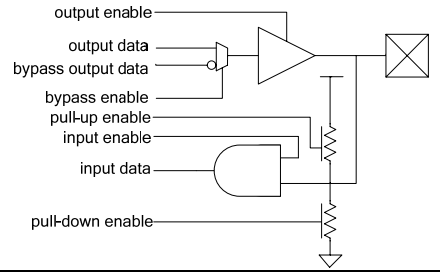
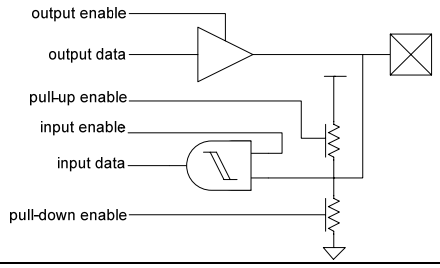
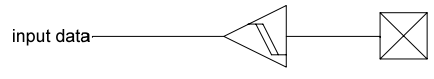
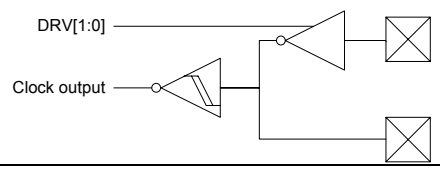
NAME	BALL	I/O	INIT	FUNC0						
LVDS_TCLKP	V6			LVDS_TCLKP						
LVDS_TCLKN	V7			LVDS_TCLKN						
LVDS_TEP	Y6			LVDS_TEP						
LVDS_TEN	W6			LVDS_TEN						
LVDS_ROUT	P6			LVDS_ROUT						
LVDS_TDP	Y7			LVDS_TDP						
LVDS_TDN	W7			LVDS_TDN						
LVDS_TCP	Y8			LVDS_TCP						
LVDS_TCN	W8			LVDS_TCN						
LVDS_TBP	Y9			LVDS_TBP						
LVDS_TBN	W9			LVDS_TBN						
LVDS_TAP	Y10			LVDS_TAP						
LVDS_TAN	W10			LVDS_TAN						

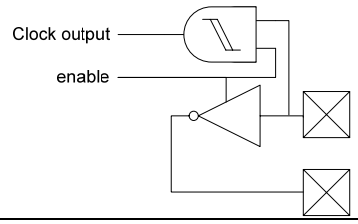
Table 3.20 PWRDAC Group I/O Pin Description

NAME	BALL	I/O	INIT	FUNC0						
DAC_OUT	U8			DAC_OUT						
DAC_COMP	T7			DAC_COMP						
DAC_IREF	V8			DAC_IREF						
DAC_VREF	V9			DAC_VREF						

3.2 TCC8900 I/O Type

Table 3.21 TCC8900 I/O Type

TYPE	Diagram	DESCRIPTION	PAD Name
A		cmos input with controllable pull-up/down and output PAD	TDO RTCK PMWKUP
B		analog and digital mixed PAD	AIN[0]~AIN[7]
C		cmos input, bypass input and output PAD	GPIOF Group
D		cmos input with controllable pull-up/down, output, and bypass output PAD	GPIOC Group
E		schmitt trigger input with controllable pull-up/down and output PAD	GPIOA Group GPIOB Group GPIOD Group GPIOE Group
F		schmitt trigger input	BPEN RSTN BM[2]~BM[0] TEST
G		Real Time Clock Oscillator	XTIN,XTOUT

H		oscillator for XIN/XOUT	XIN,XOUT

4 Package Information

4.1 Dimension

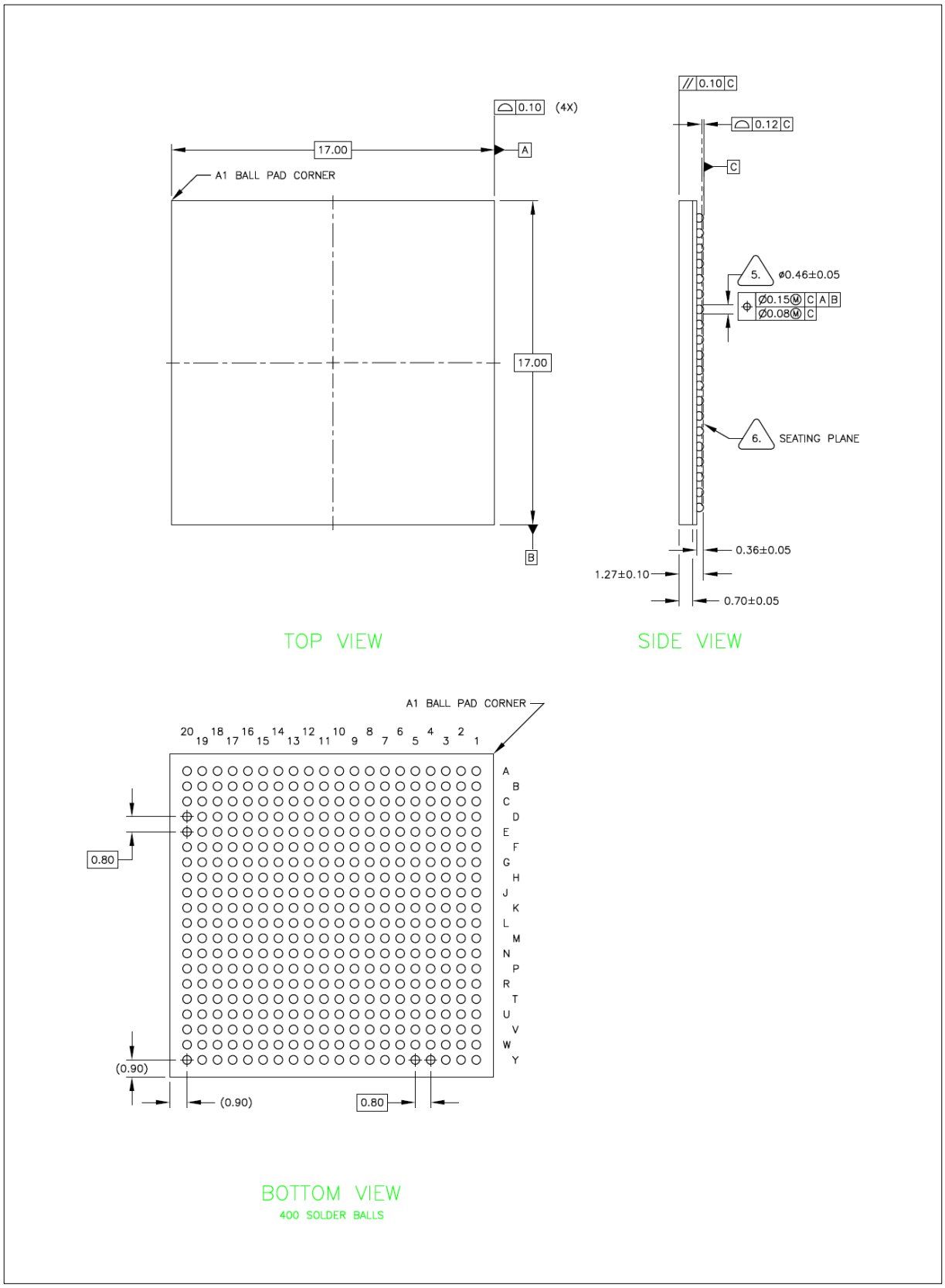


Figure 4.1 TCC8900 Package Dimension

4.2 Ball Map

S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	BGA1
A	GPIOF[0]	GPIOQ[28]	GPIOQ[23]	GPIOQ[18]	GPIOQ[17]	GPIOQ[11]	GPIOQ[7]	GPIOQ[1]	GPIOC[0]	UBH_DM	UBH_DP	GPIOB[19]	GPIOB[17]	GPIOB[15]	GPIOB[9]	GPIOD[21]	GPIOD[7]	GPIOD[5]	GPIOD[2]	GPIOD[0]	A
B	GPIOF[2]	GPIOQ[31]	GPIOQ[26]	GPIOQ[22]	GPIOQ[19]	GPIOQ[16]	GPIOQ[10]	GPIOQ[4]	GPIOC[2]	GNDUSBH	PWRUSBH	GPIOB[25]	GPIOB[13]	GPIOB[5]	GPIOB[1]	GPIOD[19]	GPIOD[13]	GPIOD[6]	TEST	TDO	B
C	GPIOF[5]	GPIOF[4]	GPIOQ[29]	GPIOQ[25]	GPIOQ[21]	GPIOQ[15]	GPIOQ[12]	GPIOQ[8]	GPIOC[3]	GPIOB[31]	GPIOB[27]	GPIOB[23]	GPIOB[7]	GPIOD[23]	GPIOD[15]	GPIOD[11]	GPIOD[3]	GPIOD[4]	RSTN	BM[2]	C
D	GPIOF[11]	GPIOF[6]	GPIOF[1]	GPIOQ[30]	GPIOQ[24]	GPIOQ[20]	GPIOQ[13]	GPIOQ[9]	GPIOC[5]	GPIOB[29]	GPIOB[21]	GPIOB[11]	GPIOD[25]	GPIOD[17]	GPIOD[9]	GPIOD[1]	TMS	DRAM_XD0	DRAM_XD7	DRAM_XD1	D
E	GPIOF[12]	GPIOF[8]	GPIOF[7]	GPIOF[3]	GPIOQ[27]	PWRGPIOC	GPIOQ[14]	GPIOQ[6]	PWRGPIOC	GNDCORE	GNDIO	GNDCORE	GPIOB[3]	GNDIO	RTCK	TDI	NTRST	DRAM_XD2	DRAM_XD6	DRAM_XD4	E
F	GPIOF[14]	GPIOF[10]	GPIOF[9]	GNDCORE	BPEN	PWRCORE	GNDCORE	GNDIO	PWRGPIOB	GPIOB[14]	PWRGPIOB	GNDIO	PWRGPIOD	PWRGPIOD	GNDIO	TCK	DRAM_XD5	DRAM_DQM0	DRAM_XD3	DRAM_XD9	F
G	GPIOF[16]	GPIOF[15]	GPIOF[13]	PWRGPIOF	GPIOB[24]	GPIOB[26]	GPIOB[28]	GPIOB[30]	PWRCORE	GPIOB[16]	PWRCORE	PWRCORE	GPIOD[8]	PWRETC	BM[1]	DRAM_ZQ	DRAM_DQS80	DRAM_DQS0	DRAM_XD15	DRAM_XD11	G
H	OTG_DP	GPIOF[17]	GPIOF[18]	PWRCORE	GNDIO	GPIOB[12]	GPIOB[18]	GPIOB[22]	GPIOB[20]	GPIOB[10]	GPIOB[8]	GNDMEMZQ	GPIOD[10]	PWRCORE	DRAM_VREF0	BM[0]	DRAM_XD10	DRAM_XD13	DRAM_XD14	DRAM_XD12	H
J	OTG_DM	GPIOF[19]	GPIOF[20]	PWRCORE	GNDCORE	GPIOF[21]	GPIOB[0]	GPIOB[4]	GPIOB[6]	GPIOB[2]	GPIOD[18]	PWRMEMZQ	PWRMEMQ	PWRCORE	GNDCORE	DRAM_XD8	DRAM_DQSB1	DRAM_DQS1	DRAM_DQM1	DRAM_XA9	J
K	GNDUSB	GPIOF[23]	GPIOF[25]	GPIOF[24]	GPDF[27]	GPIOF[26]	GPIOF[22]	GPIOD[22]	GPIOD[24]	GPIOD[20]	GPIOD[16]	DRAM_XA8	GNDCORE	GNDCORE	DRAM_VREF1	GNDCORE	DRAM_WEN	DRAM_CKE	DRAM_XA7	DRAM_GATE0	K
L	XO	XI	OTG_VBUS	PWRUSB12	PWRUSB33	PWROSC	GNDCORE	GPIOE[4]	GPIOE[17]	GPIOD[14]	GPIOD[12]	DRAM_ODT1	DRAM_XA1	PWRMEMQ	PWRCORE	DRAM_XA4	DRAM_BA1	DRAM_BA2	DRAM_BA0	DRAM_GATE1	L
M	GNDSATA	OTG_REXT	GNDPLL	PWRCORE	RTC_PWKUP	GNDOSC	GPIOE[3]	GPIOE[15]	GPIOA[5]	GPIOA[9]	DRAM_CASN	DRAM_ODT0	GNDCORE	PWRMEMQ	PWRMEMQ	DRAM_XA3	DRAM_CSN1	DRAM_XA2	DRAM_CSN0	DRAM_XA6	M
N	SATA_RXP	RTC_XTO	RTC_XTI	PWRRTC	PWRPLL	RTC_RSTN	GPIOE[0]	GPIOE[5]	GPIOA[3]	GPIOA[7]	GPIOA[14]	DRAM_XA0	GNDCORE	PWRMEMQ	PWRCORE	DRAM_XA10	DRAM_XA5	DRAM_RASN	DRAM_XA11	DRAM_CLK	N
P	SATA_RXN	PWRSATAPLL	PWRSATA2	GNDIO	PWRGPIOF	LVDS_ROUT	PWRLVDS33A	GPIOE[1]	GPIOE[13]	GPIOE[21]	GPIOA[13]	DRAM_XA12	PWRCORE	PWRMEMQ	GNDCORE	DRAM_VREF2	DRAM_XD18	DRAM_DQSB2	DRAM_XD21	DRAM_CLKB	P
R	GNDSATA	GNDSATAPLL	SATA_XI	PWRSATA1	PWRHDMIPLL1	GNDHDMIPLL	PWRLVDS33A	PWRCORE	GPIOE[6]	GPIOE[19]	GPIOE[23]	GPIOA[15]	GNDCORE	AIN[7]	GNDCORE	DRAM_XD20	DRAM_DQM2	DRAM_DQS2	DRAM_XD23	DRAM_XD16	R
T	SATA_TXN	SATA_REXT	SATA_XO	PWRHDMIPLL2	PWRHDMI	GNDLVDS33A	DAC_COMP	GNDDAC	GNDCORE	GPIOE[11]	GNDIO	GPIOA[1]	PWRADC	GNDADC	PWRCORE	DRAM_VREF3	DRAM_XD25	DRAM_DQSB3	DRAM_XD19	DRAM_XD22	T
U	SATA_TXP	PWRSATAOSC	HDMI_XO	HDMI_XI	PWRHDMIOSC	GNDHDMI	GNDDAC	DAC_OUT	PWRDAC	GPIOE[2]	GPIOE[7]	GNDCORE	GPIOA[11]	GPIOA[10]	AIN[1]	AIN[6]	DRAM_XD27	DRAM_DQS3	DRAM_XD31	DRAM_XD17	U
V	GNDSATAOSC	GNDHDMI	GNDHDMI	HDMI_REXT	GNDHDMIOSC	LVDS_TCLKP	LVDS_TCLKN	DAC_REF	DAC_VREF	PWRGPIOE	GNDIO	PWRGPIOE	PWRGPIOA	GPIOA[12]	GPIOA[8]	AIN[0]	AIN[5]	DRAM_DQM3	DRAM_XD24	DRAM_XD29	V
W	HDMI_TXCN	HDMI_TX0N	HDMI_TX1N	HDMI_TX2N	GNDLVDS33A	LVDS_TEN	LVDS_TDN	LVDS_TCN	LVDS_TBN	LVDS_TAN	GPIOE[8]	GPIOE[18]	PWRCORE	GPIOA[2]	GPIOA[4]	GPIOA[6]	AIN[2]	AIN[3]	DRAM_XD28	DRAM_XD26	W
Y	HDMI_TXCP	HDMI_TX0P	HDMI_TX1P	HDMI_TX2P	GNDLVDS33A	LVDS_TEP	LVDS_TDP	LVDS_TCP	LVDS_TBP	LVDS_TAP	GPIOE[9]	GPIOE[10]	GPIOE[12]	GPIOE[14]	GPIOE[16]	GPIOE[20]	GPIOE[22]	GPIOA[0]	AIN[4]	DRAM_XD30	Y
BGA1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	BGA1

Figure 4.2 TCC8900 Ball Map

5 Electrical Specification

5.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
DC Supply Voltage for I/O (Various I/O power excepts for analog – ADC, DAC, PLL, USB)	V_{DDIO}	4.6	V
DC Supply Voltage for Internal Digital Logic (PWR12)	V_{DDI}	1.8	V
DC Supply Voltage for Analog Part of ADC (PWRADC)	V_{DDADC}	4.6	V
DC Supply Voltage for PLL (PWRPLL)	V_{DDPLL}	1.8	V
DC Supply Voltage for USB2.0 (PWRUSB)	V_{DDUSB}	4.6	V
DC Supply Voltage for RTC (PWRRTC)	V_{DDRTC}	4.6	V
Digital Input Voltage for Input Buffer	V_{IN}	4.6	V
Digital Input Voltage for OTG_VBUS	V_{OTG_VBUS}	6.0	V
Digital Output Voltage for Output Buffer	V_{OUT}	4.6	V
In/Out Current for Digital I/O	$I_{I/O}$	±20	mA
Analog Input Voltage for ADC	V_{IN_ADC}	0 ~ V_{DDADC}	V
Storage Temperature	T_{STG}	-55 to 150	°C

Note:
Absolute maximum ratings specify the values beyond which the device may be damaged permanently. Exposure to absolute maximum rating conditions for extended periods may affect reliability. Each condition value is applied with the other values kept within the following operating conditions and functional operation under any of these conditions is not implied.

(1) All voltages are measured with respect to VSS unless otherwise specified.
(2) V_{DDI} must always be less than V_{DDIO}
(3) The voltage difference between analog and digital grounds must always be within 0.3V.

5.2 Recommended Operating Conditions

Table 5.1 Recommended Operating Conditions[23]

Parameter	Symbol	MIN	TYP	MAX	Unit
Output Load Resistance for DAC [±1% tolerance]	R_{LOAD}	-	37.5	-	Ω
Core (PWRCORE) and PLL (PWRPLL) and USB(PWRUSB12) supply Voltage	V_{DDI} V_{DDPLL} $V_{DDUSB12}$	0.95	1.0	1.05	V
		1.05	1.1	1.15	
		1.14	1.2	1.26	
		1.23	1.3	1.37	
		1.33	1.4	1.47	
		1.43	1.5	1.57	
Operating Temperature [Extended]	T_{OPER}	-30		85	°C
Operating Temperature [Industrial]	T_{OPER}	-40	-	85	°C

23 The recommended operating conditions for power/ground are described on the Power/Ground Information in the Pin Descriptions.

5.3 Recommended Operating Frequency

Table 5.2 Recommended Operating Frequency

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
XIN Oscillator	@ 1.0V(TYP)	F_{XIN}	12	12	12	MHz
	@ 1.1V(TYP)					
	@ **1.2V(TYP)**					
	@ 1.3V(TYP)					
	@ 1.4V(TYP)					
	@ 1.5V(TYP)					
XTIN Oscillator	@ 1.2V(TYP)	F_{XTIN}	32.768		32.768*128	KHz
PLL0 VCO Range	@ 1.0V(TYP)	F_{PLL0_VCO}	800		960	MHz
	@ 1.1V(TYP)		800		1280	
	@ **1.2V(TYP)**		**800**		**1600**	
	@ 1.3V(TYP)		800		1920	
	@ 1.4V(TYP)		800		2240	
	@ 1.5V(TYP)		800		2560	
PLL1/2/3 VCO Range	@ 1.0V(TYP)	F_{PLL123_VCO}	250		360	MHz
	@ 1.1V(TYP)		250		480	
	@ **1.2V(TYP)**		**250**		**600**	
	@ 1.3V(TYP)		250		720	
	@ 1.4V(TYP)		250		840	
	@ 1.5V(TYP)		250		960	
Input Clock of Bus Clock Generator (1 ~ 7)[24]	@ 1.0V(TYP)	$F_{BCLKGEN}$			546	MHz
	@ 1.1V(TYP)				728	
	@ **1.2V(TYP)**				**910**	
	@ 1.3V(TYP)				1092	
	@ 1.4V(TYP)				1274	
	@ 1.5V(TYP)				1456	
Input Clock of CPU Clock Generator[25]	@ 1.0V(TYP)	F_{CPUGEN}			300	MHz
	@ 1.1V(TYP)				400	
	@ **1.2V(TYP)**				**500**	
	@ 1.3V(TYP)				600	
	@ 1.4V(TYP)				700	
	@ 1.5V(TYP)				800	
Input Clock of PLL Divider	@ 1.0V(TYP)	$F_{PLLDIVIN}$			498	MHz
	@ 1.1V(TYP)				664	
	@ **1.2V(TYP)**				**830**	
	@ 1.3V(TYP)				996	
	@ 1.4V(TYP)				1162	
	@ 1.5V(TYP)				1328	
Input Clock of I/O Clock Generator[26]	@ 1.0V(TYP)	$F_{IOCLKGEN}$			300	MHz
	@ 1.1V(TYP)				400	
	@ **1.2V(TYP)**				**500**	
	@ 1.3V(TYP)				600	
	@ 1.4V(TYP)				700	
	@ 1.5V(TYP)				800	
ARM1176JZF-S Core Clock	@ 1.0V(TYP)	F_{CPU}			300	MHz
	@ 1.1V(TYP)				400	
	@ **1.2V(TYP)**				**500**	
	@ 1.3V(TYP)				600	
	@ 1.4V(TYP)				700	
	@ 1.5V(TYP)				800	
Bus Clock of DDI Bus	@ 1.0V(TYP)	F_{BUS_DDI}			144	MHz
	@ 1.1V(TYP)				192	
	@ **1.2V(TYP)**				**240**	
	@ 1.3V(TYP)				288	
	@ 1.4V(TYP)				336	
	@ 1.5V(TYP)				384	
Bus Clock of Graphic Bus	@ 1.0V(TYP)	F_{BUS_GRP}			114	MHz
	@ 1.1V(TYP)				152	
	@ **1.2V(TYP)**				**190**	

24 The related clocks are F_{BUS_DDI}, F_{BUS_GRP}, F_{BUS_MEM}, F_{BUS_VBUS}, F_{BUS_VCODEC},F_{BUS_SMU}, F_{BUS_IOB}.
25 The related clock is F_{CPU}.
26 The prefix of related clocks is $F_{IO\_}$.

	@ 1.3V(TYP)				228	
	@ 1.4V(TYP)				266	
	@ 1.5V(TYP)				304	
Bus Clock of I/O Bus	@ 1.0V(TYP)	F_{BUS_IOB}			100	MHz
	@ 1.1V(TYP)				133	
	@ **1.2V(TYP)**				**166**	
	@ 1.3V(TYP)				200	
	@ 1.4V(TYP)				232	
	@ 1.5V(TYP)				266	
Bus Clock of SMU Controller	@ 1.0V(TYP)	F_{BUS_SMU}			75	MHz
	@ 1.1V(TYP)				100	
	@ **1.2V(TYP)**				**125**	
	@ 1.3V(TYP)				150	
	@ 1.4V(TYP)				175	
	@ 1.5V(TYP)				200	
Bus Clock of Video Bus	@ 1.0V(TYP)	F_{BUS_VBUS}			129	MHz
	@ 1.1V(TYP)				175	
	@ **1.2V(TYP)**				**215**	
	@ 1.3V(TYP)				258	
	@ 1.4V(TYP)				301	
	@ 1.5V(TYP)				344	
Core Clock of Video Codec	@ 1.0V(TYP)	F_{BUS_VCODEC}			96	MHz
	@ 1.1V(TYP)				128	
	@ **1.2V(TYP)**				**160**	
	@ 1.3V(TYP)				192	
	@ 1.4V(TYP)				224	
	@ 1.5V(TYP)				256	
Bus Clock of Memory Interface (DDR2)	@ 1.0V(TYP)	F_{BUS_MEM}			156	MHz
	@ 1.1V(TYP)				208	
	@ **1.2V(TYP)**				**260**	
	@ 1.3V(TYP)				312	
	@ 1.4V(TYP)				364	
	@ 1.5V(TYP)				400	
Bus Clock of Memory Interface (DDR,MDDR,SDRAM)	@ 1.0V(TYP)	F_{BUS_MEM}			108	MHz
	@ 1.1V(TYP)				144	
	@ **1.2V(TYP)**				**180**	
	@ 1.3V(TYP)				200	
	@ 1.4V(TYP)				200	
	@ 1.5V(TYP)				200	
Operating Clock of Camera Interface[27]	@ 1.0V(TYP)	F_{IO_CIF}			120	MHz
	@ 1.1V(TYP)				160	
	@ **1.2V(TYP)**				**200**	
	@ 1.3V(TYP)				240	
	@ 1.4V(TYP)				280	
	@ 1.5V(TYP)				320	
Operating Clock of EHI[27]	@ 1.0V(TYP)	F_{IO_EHI}			100	MHz
	@ 1.1V(TYP)				133	
	@ **1.2V(TYP)**				**166**	
	@ 1.3V(TYP)				200	
	@ 1.4V(TYP)				232	
	@ 1.5V(TYP)				266	
Operating Clock of GPSB Controller[27]	@ 1.0V(TYP)	F_{IO_GPSB}			100	MHz
	@ 1.1V(TYP)				133	
	@ **1.2V(TYP)**				**166**	
	@ 1.3V(TYP)				200	
	@ 1.4V(TYP)				232	
	@ 1.5V(TYP)				266	
Operating Clock of Memory Stick Controller[27]	@ 1.0V(TYP)	F_{IO_MSTICK}			60	MHz
	@ 1.1V(TYP)				80	
	@ **1.2V(TYP)**				**100**	
	@ 1.3V(TYP)				120	
	@ 1.4V(TYP)				140	
	@ 1.5V(TYP)				160	
Operating Clock of SD/MMC Controller[27]	@ 1.0V(TYP)	F_{IO_SDMMC}			60	MHz
	@ 1.1V(TYP)				80	
	@ **1.2V(TYP)**				**100**	
	@ 1.3V(TYP)				120	
	@ 1.4V(TYP)				140	
	@ 1.5V(TYP)				160	

27 The operating frequencies of external interface are not same as this. More detailed information is described on the timing characteristics of I/O interface. Refer to the corresponding timing information.

Operating Clock of UART Controller[27]	@ 1.0V(TYP)	F_{IO_UART}			100	MHz
	@ 1.1V(TYP)				133	
	@ 1.2V(TYP)				**166**	
	@ 1.3V(TYP)				200	
	@ 1.4V(TYP)				232	
	@ 1.5V(TYP)				266	
Operating Clock of LCD Controller[27]	@ 1.0V(TYP)	F_{IO_LCD}			100	MHz
	@ 1.1V(TYP)				133	
	@ 1.2V(TYP)				**166**	
	@ 1.3V(TYP)				200	
	@ 1.4V(TYP)				232	
	@ 1.5V(TYP)				266	
Operating Clock of PMU	@ 1.0V(TYP)	F_{IO_PMU}	12	12	12	MHz
	@ 1.1V(TYP)					
	@ 1.2V(TYP)					
	@ 1.3V(TYP)					
	@ 1.4V(TYP)					
	@ 1.5V(TYP)					
Operating Clock of Timer	@ 1.0V(TYP)	F_{IO_TIMER}			75	MHz
	@ 1.1V(TYP)				100	
	@ 1.2V(TYP)				**125**	
	@ 1.3V(TYP)				150	
	@ 1.4V(TYP)				175	
	@ 1.5V(TYP)				200	
Operating Clock of TSIF(Not GPSB)	@ 1.0V(TYP)	F_{IO_TSIF}			48	MHz
	@ 1.1V(TYP)				64	
	@ 1.2V(TYP)				**80**	
	@ 1.3V(TYP)				96	
	@ 1.4V(TYP)				112	
	@ 1.5V(TYP)				128	
Operating Clock of SPDIF Transmitter/Receiver[27]	@ 1.0V(TYP)	F_{IO_SPDIF}			60	MHz
	@ 1.1V(TYP)				80	
	@ 1.2V(TYP)				**100**	
	@ 1.3V(TYP)				120	
	@ 1.4V(TYP)				140	
	@ 1.5V(TYP)				160	
Operating Clock of Audio (ADMA/DAI/CDIF) [27]	@ 1.0V(TYP)	F_{IO_AUDIO}			100	MHz
	@ 1.1V(TYP)				133	
	@ 1.2V(TYP)				**166**	
	@ 1.3V(TYP)				200	
	@ 1.4V(TYP)				232	
	@ 1.5V(TYP)				266	
Operating Clock of I2C[27]	@ 1.0V(TYP)	F_{IO_I2C}			100	MHz
	@ 1.1V(TYP)				133	
	@ 1.2V(TYP)				**166**	
	@ 1.3V(TYP)				200	
	@ 1.4V(TYP)				232	
	@ 1.5V(TYP)				266	
Operating Clock of USB 1.1 Host	@ 1.2V(TYP)	F_{IO_USBH}	48	48	48	MHz

➔The maximum operating frequency can be changed without any notice until approved for mass production.

5.4 Electrical Characteristics for Power Supply

Table 5.3 Peak Power Consumption

Parameter	Power	Condition	MIN	TYP	MAX	Unit
Internal Core Power	PWRCORE	@ 1.2V			TBD	mA
GPIO Power	PWRGPIOn, (n=A,B,C,D,E,F) PWRETC	@ 1.8V @ 2.7V @ 3.3V			TBD	mA
Memory I/O Power	PWRMEMQ, PWRMEMZQ	@ 1.8V @ 2.5V @ 3.3V			TBD	mA
Oscillator Power (XI/XO)	PWROSC	@ 3.3V			TBD	mA
USB 1.2 Power of nanoPHY	PWRUSB12	@ 1.2V			TBD	mA
USB 3.3 Power of nanoPHY	PWRUSB33	@ 3.3V			TBD	mA
USB 1.1 Host Tranceiver Power	PWRUSBH	@ 3.3V			TBD	mA
RTC Power	PWRRTC	@ 2.7V			TBD	mA

➔The rests of the power which are not described in the above table are shown in the corresponding sub-section.

➔***The value in the above table does not mean the average power***. Refer to this at the designing of the power circuit.

5.5 Electrical Characteristics for General I/O

Table 5.4 DC Electrical Specification for General I/O

Parameter	Symbol	Test Condition	MIN	TYP	MAX	Unit
High Level Input Voltage	V_{IH}		$0.7V_{DDIO}$		V_{DDIO}+0.3	V
Low Level Input Voltage	V_{IL}		-0.3		$0.3V_{DDIO}$	V
Hysteresis Voltage	△V		$0.1V_{DDIO}$			V
High Level Input Current	I_{IH}	VIN = VDDIO, pull-down disabled	-10		10	μA
		VIN = VDDIO, pull-down enabled	TBD		TBD	μA
Low Level Input Current	I_{IL}	VIN = VSSIO, pull-up disabled	-10		10	μA
		VIN = VSSIO, pull-up enabled	TBD		TBD	μA
High Level Output Voltage	V_{OH}	IOH = -100uA	V_{DDIO}–0.2			V
Low Level Output Voltage	V_{OL}	IOL = 100uA			0.2	V
Tri-state Output Leakage Current	I_{OZ}	VOUT = VSSIO or VDDIO	-10		10	μA
Input capacitance	C_{IN}	Any input and Bidirectional buffers			5	pF
Output capacitance	C_{OUT}	Any output buffer			5	pF
XI/XO Frequency	F_{OSC1}		-	12	-	MHz
XTIN/XTOUT Frequency	F_{OSC2}	Normal High Drive = Normal * 128	-	32.768 4194.304	-	kHz

Ta = 25oC, VSS = 0.0V unless otherwise specified.

Note:

(1) 12MHz is recommended for XI/XO frequency.
(2) PLL Output Frequencies are determined by XI/XO frequency.

5.6 Electrical Characteristics for PLL

Table 5.5 DC Electrical Characteristics for PLL0

Parameter	Symbol	Test Condition	MIN	TYP	MAX	Unit
Power Down Current	I_{PD}	V_{DDPLL} = 1.2V			80	μA
Power Consumption	P_{DD}	V_{DDPLL} = 1.2V			3.0	mW

Ta = 25oC unless otherwise specified.

Table 5.6 AC Electrical Characteristics for PLL0

Parameter	Symbol	Test Condition	MIN	TYP	MAX	Unit
Input Frequency	Fin	V_{DDPLL} = 1.2V		12		MHz
VCO output frequency	Fvco	V_{DDPLL} = 1.2V	800	-	1600	MHz
Output Frequency	Fout	V_{DDPLL} = 1.2V	40		1600	MHz
Locking Time	T_{LT}	V_{DDPLL} = 1.2V			300	us

Ta = 25oC unless otherwise specified.

Table 5.7 DC Electrical Characteristics for PLL1/2/3

Parameter	Symbol	Test Condition	MIN	TYP	MAX	Unit
Power Down Current	I_{PD}	V_{DDPLL} = 1.2V			80	μA
Power Consumption	P_{DD}	V_{DDPLL} = 1.2V			1.5	mW

Ta = 25oC unless otherwise specified.

Table 5.8 AC Electrical Characteristics for PLL1/2/3

Parameter	Symbol	Test Condition	MIN	TYP	MAX	Unit
Input Frequency	Fin	V_{DDPLL} = 1.2V		12		MHz
VCO output frequency	Fvco	V_{DDPLL} = 1.2V	250	-	600	MHz
Output Frequency	Fout	V_{DDPLL} = 1.2V	8		600	MHz
Locking Time	T_{LT}	V_{DDPLL} = 1.2V			300	us

Ta = 25oC unless otherwise specified.

5.7 Electrical Characteristics for Video DAC

Table 5.9 DC Electrical Characteristics for DAC

Parameter	Symbol	Test Condition	MIN	TYP	MAX	Unit
Resolution	Bit		-	-	10	bits
Conversion Rate	F_{CLK}		-	-	27	MHz
Differential Non-Linearity	DNL		-	-	±1	LSB
Integral Non-Linearity	INL		-	-	±2.5	LSB
Full Scale Output Voltage	V_O		1.17	1.3	1.43	V
Output Load	R_{LOAD}	±1% tolerance		37.5		Ω
External Reference Voltage	V_{REF}		-	1.26	-	V

(VDDDAC =3.0V, VSSDAC =0V, Power Down = OFF, Top=30° C,
R(IREF) =1.2kΩ, Load Resistance=37.5Ω unless otherwise specified.)

5.8 Electrical Characteristics for ADC(for Touch Screen)

Table 5.10 DC Electrical Characteristics for ADC

Parameter	Symbol	Test Condition	MIN	TYP	MAX	Unit
Resolution	Bit		-	-	10(TBD)	bits
Differential Non-Linearity	DNL	VREF=3.3V, GND=0V	-	-	±1(TBD)	LSB
Integral Non-Linearity	INL	VREF=3.3V, GND=0V	-	-	±3(TBD)	LSB
Offset Voltage	TOPOFF BOTOFF	VREF=3.3V, GND=0V	-	-	10(TBD)	LSB

(Converter Specifications: VDDADC= 3.3V, VSSADC= 0V, Top=25°C, VREF=3.3V, GND=0.0V unless otherwise specified)

Table 5.11 AC Electrical Characteristics for ADC

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Maximum conversion rate	fc	f_{CKIN} = 5MHz	–	–	1	MSPS
Standby supply current	–	STBY = VDD	–	-	60(TBD)	uA
Dynamic supply current	IVDD	f_{CKIN} = 5MHz (without system load)	–	3.0(TBD)	5(TBD)	mA
Reference current	IREF	VREF = 3.3V	–	0.4(TBD)	0.6(TBD)	mA
Total harmonic distortion	THD	f_{CKIN} = 5MHz AINT = 50kHz	–	– 60(TBD)	– 56(TBD)	dB
Signal-to-noise & distortion ratio	SNDR	f_{CKIN} = 5MHz AINT = 50kHz	48(TBD)	54(TBD)	–	dB

(Converter Specifications: VDDADC =3.3V, VSSADC=0V, Top=25°C, VREF=3.3V, GND=0.0V unless otherwise specified) VDDIO = 3.3V±0.3V

5.9 Electrical Characteristics for HDMI PHY

Table 5.12 DC Electrical Characteristics for HDMI PHY

Parameter	Symbol	Test Condition	MIN	TYP	MAX	Unit
Normal Mode Operating Power	P_{CC}	Internal Video PLL ON Internal Video PLL OFF	-	72 42	-	mW
Power-Down Mode Power	P_{PD}	-	-	TBD	-	mW

Table 5.13 AC Electrical Characteristics for HDMI Oscillator

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
HDMI Oscillator Frequency	F_R	-	–	27		MHz
Frequency Tolerance	F_{TOL}		-100		100	ppm
Duty Cycle	D_C		40		60	%
Jitter	J_{CLKI}	Peak-to-Peak Jitter RMS Jitter			100 7	ps ps

5.10 Electrical Characteristics for LVDS

Table 5.14 DC Electrical Characteristics for LVDS

Parameter	Symbol	Test Condition	MIN	TYP	MAX	Unit
Output differential voltage	V_{OD}		3.0	3.3	3.6	V
Change in V_{OD} between complementary output states	DV_{OD}	-	-		50	mV
Output offset voltage	V_{OS}		1.125	1.25	1.375	V
Change in V_{OS} between complementary output states	DV_{OS}				50	mV
Dynamic Current	I_{DD}				70	mA
Power Down Current	I_{PD}				100	uA

Typical values measured at PWRLVDS=3.3V and TA=25 oC

Table 5.15 AC Electrical Characteristics for LVDS

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
TXCLKIN Period	T_{TCP}		10		40	ns
DLL Lock Time	T_{DLL}	-			100	us
Skew Between Channels	T_{SK}		-200		200	ps
Duty Cycle	D_C		40		60	%
Jitter	J_{CLKI}	Peak-to-Peak Jitter RMS Jitter			100 7	ps ps

These values are measured at the typical condition.

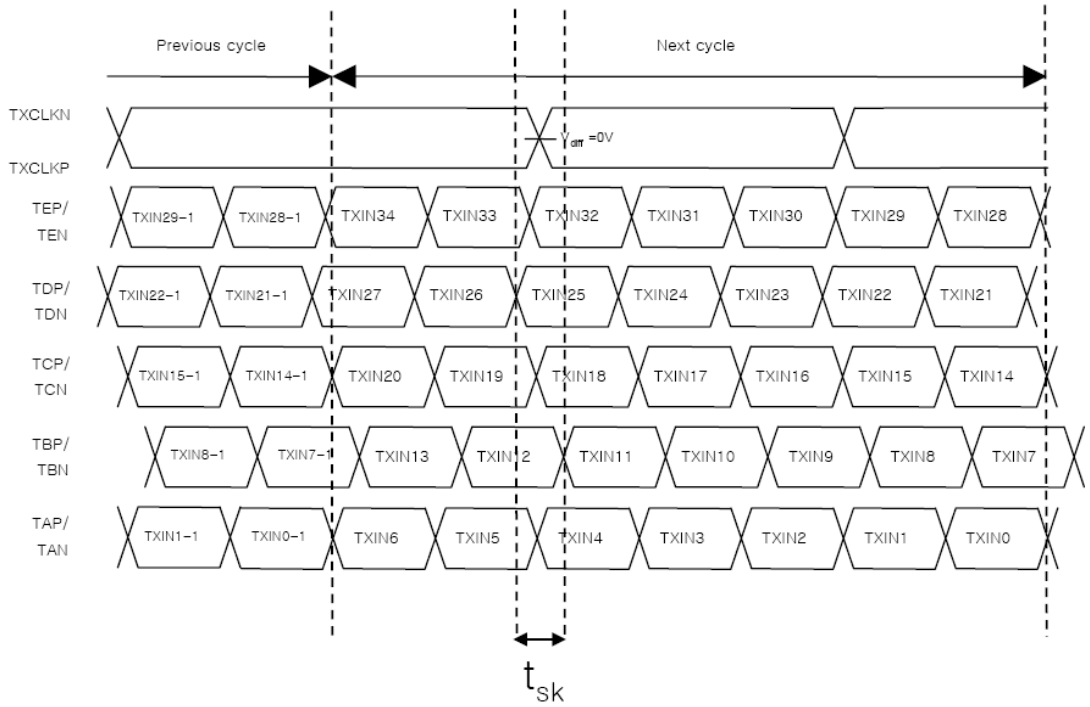


Figure 5.1 LVDS Transmit Timing Diagram

5.11 Electrical Characteristics for SATA

Table 5.16 DC Electrical Characteristics for SATA

Parameter	Symbol	Test Condition	MIN	TYP	MAX	Unit
TX serial data output voltage	DV_{TX}	1.5Gbps 3.0Gbps	400 400		700 700	mVp-p
RX serial data input voltage	DV_{RX}	1.5Gbps 3.0Gbps	325 275		600 750	mVp-p
Dynamic Current	I_{DD}	Normal Mode Partial Mode		80 40		mW
Power Down Current	I_{PD}	Slumber Mode		12		mW

All values in the above table are measured at typical condition.

Table 5.17 AC Electrical Characteristics for SATA Oscillator

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Oscillator Frequency	F_R		–	25 100		MHz
Frequency Tolerance	F_{TOL}		-100		100	ppm
Duty Cycle	D_C		40		60	%
Clock Transition Time	T_{CLKT}	Rising Falling			1.5 1.5	ns
Jitter	J_{CLKI}	Peak-to-Peak Jitter RMS Jitter			40 2.5	ps ps

Table 5.18 AC Electrical Characteristics for SATA TX/RX

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Unit Interval	UI	1.5Gbps 3.0Gbps	–	666.67 333.33		ps
TX Serial Output Rise Time (20% → 80%)	$T_{TX.RISE}$	1.5Gbps 3.0Gbps	100 67		273 136	ps
TX Serial Output Fall Time (80% → 20%)	D_C	1.5Gbps 3.0Gbps	100 67		273 136	ps
TX Serial Data Output Voltage (Differential Peak-to-Peak)	D_{VTX}	1.5Gbps 3.0Gbps	400 400		700 700	mVp-p
RX Serial Data Input Voltage (Differential Peak-to-Peak)	T_{RJ}	1.5Gbps 3.0Gbps	325 275		600 750	mVp-p

5.12 Electrical Characteristics for LCD Interface

The following figure shows the timing diagram for TFT-LCD with RGB interface. All the timing parameters can be configured by LHTIME1, LHTIME2, LVTIME1 ~ 4 registers.

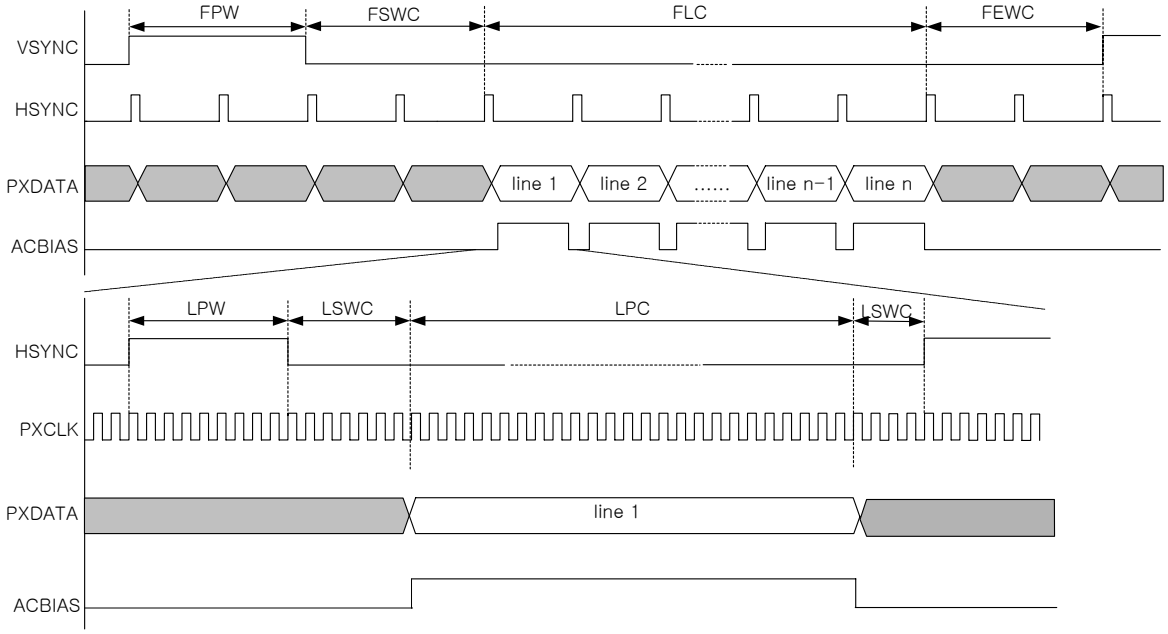


Figure 5.2 Timing Diagram for LCD Controller

The LHS (HSYNC), LVS (VSYNC), LBIAS (ACBIAS, Data Enable) and LPD[17:0] (PXDATA[17:0]) signals are referenced by LCK (PXCLK). Each min and max timing for the output delay are shown in the following figure.

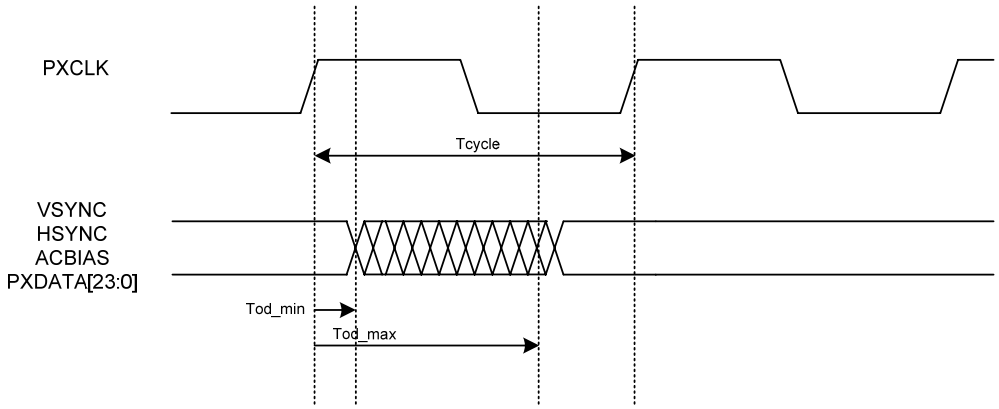


Figure 5.3 Timing Diagram Data Output Referenced to PXCLK

Table 5.19 Timing Parameters for Each Symbol

Parameter	Symbol	Min	Max	Unit	Remark
Clock Cycle	T_{CYCLE}	10	-	ns	
Output Delay	T_{OD}	0	7	ns	

Table 5.20 I/O Function Name for Corresponding Signal Name

Signal Name	I/O Function Name
VSYNC	L0_LVS, L1_LVS
HSYNC	L0_LHS, L1_LHS
ACBIAS	L0_LDE, L1_LDE
PXDATA[23:0]	L0_LPD[23:0], L1_LPD[23:0]
PXCLK	L0_LCK, L1_LCK

The following figure shows the timing diagram of bus interface to CPU I/F LCD device. The reference clock is used internally and the cycle time is defined as the register value written by software.

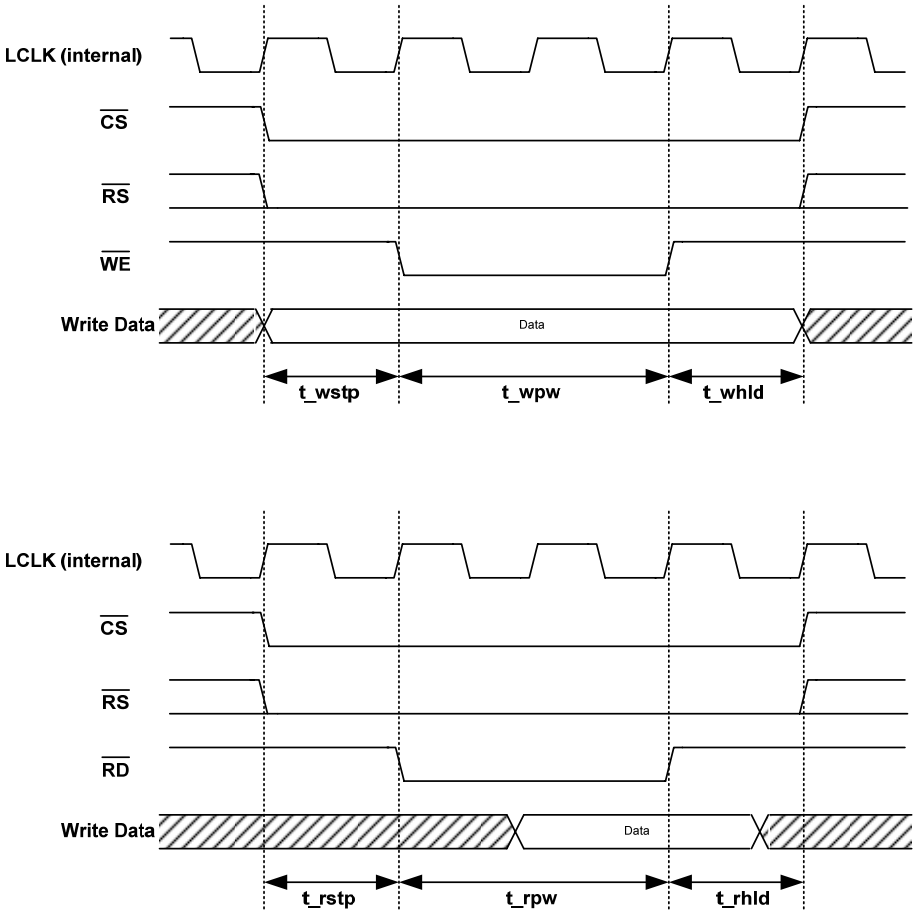


Figure 5.4 Timing Diagram Data Output Referenced to LCDSI

Table 5.21 Timing Parameters for Each Symbols

Parameter	Symbol	Min	Max	Unit	Remark
LCLK Clock Period (LCDSI Clock)	tCLK	18		ns	
RD/WE Setup Time Referenced to LCLK	T_rstp	0 * tCLK	7 * tCLK	ns	
RD/WE Pulse Width Referenced to LCLK	T_rpw	1 * tCLK	256 * tCLK	ns	
RD/WE Hold Time Referenced to LCLK	T_rhld	0 * tCLK	7 * tCLK	ns	

Signal Name	I/O Function Name
#CS	LCSN0, LCSN1
#RS	LXA[0]
#RD	LOEN
#WE	LWEN
Write Data[17:0]	LXD[17:0]

5.13 Electrical Characteristics for Camera Interface

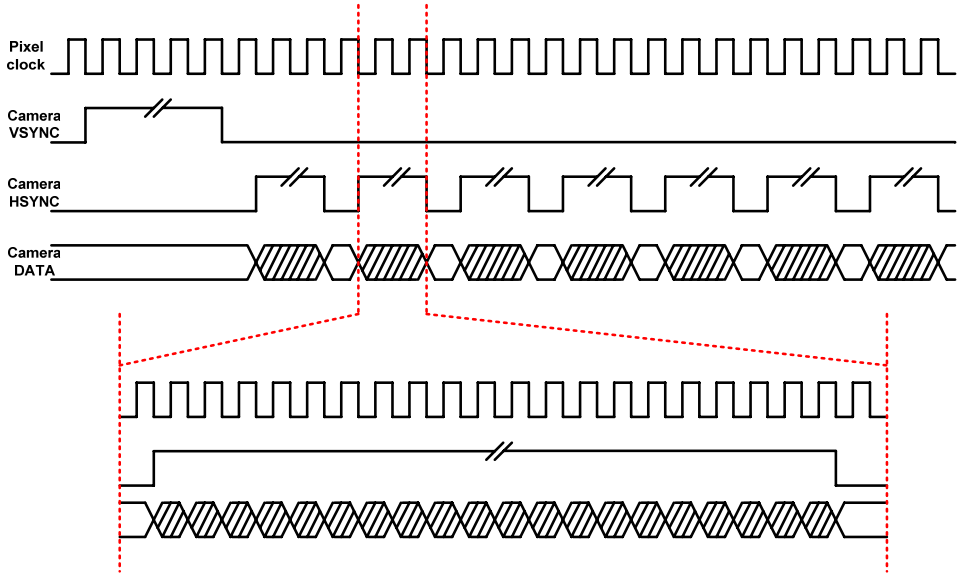


Figure 5.5 Timing Diagram for Camera Interface

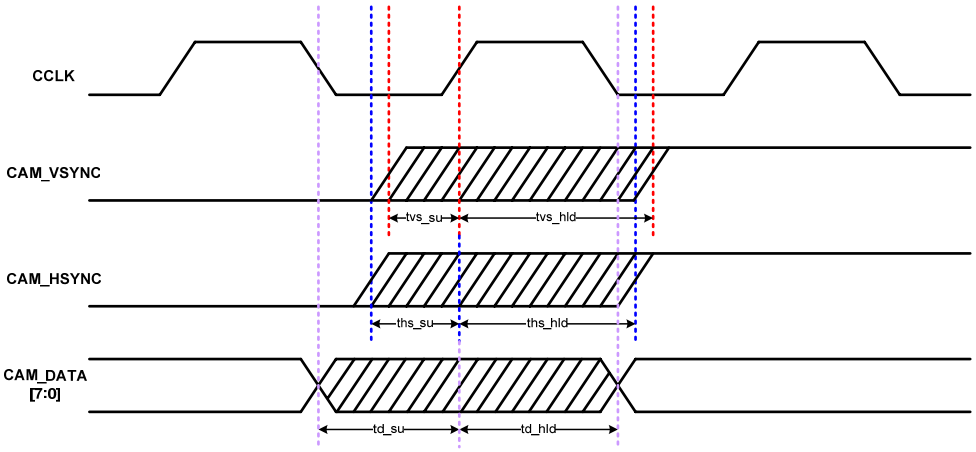


Figure 5.6 Timing Diagram Data Output Referenced to CCLK

Table 5.22 Timing Parameters for Each Symbol

Parameter	Symbol	Min	Max	Unit	Remark
Clock Frequency	TCCK		120	MHz	
Setup time for CVS(CAM_VSYNC)	Tvs_su	2		ns	
Hold time for CVS(CAM_VSYNC)	Tvs_hld	2		ns	
Setup time for CHS(CAM_HSYNC)	Ths_su	2		ns	
Hold time for CHS(CAM_HSYNC)	Ths_hld	2		ns	
Setup time for CPD[7:0](CAM_DATA)	Td_su	2		ns	
Hold time for CPD[7:0](CAM_DATA)	Td_hld	2		ns	

Table 5.23 I/O Function Name for Corresponding Signal Name

Signal Name	I/O Function Name
CCLK	CCKI
CAM_VSYNC	CVS
CAM_HSYNC	CHS
CAM_DATA[7:0]	CPD[7:0]

5.14 Electrical Characteristics for External Host Interface (EHI)

The EHI has two clock inputs; one is HCLK, which is for the on-chip system bus, the other is ECLK, which is for interface with the external host device. Therefore, interface timing with the external host is only related with ECLK.

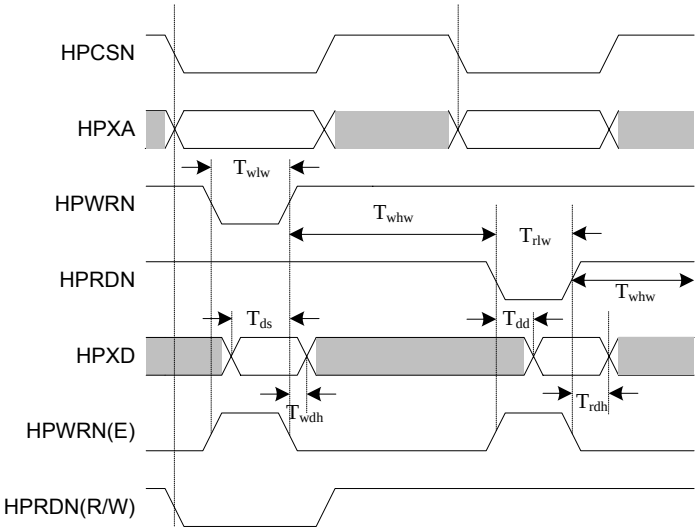


Figure 5.7 EHI Timing Diagram

Symbol	Description	Min.	Max.	Unit
Twlw	Write low width	2TP[28]	-	ns
Twhw	Write high width	3TP	-	ns
Trlw	Read low width	4TP	-	ns
Trhw	Read high width	3TP	-	ns
Tds	Data setup time	10	-	ns
Twdh	Write data hold	5	25ns	ns
Tdd	Data delay time	-	3TP +10ns	-
Trdh	Read data hold	0		ns

28 TP = ECLK period (ns)

5.15 Electrical Characteristics for SD/MMC Controller

The SD/MMC host controller is designed to supports high-speed mode (SD rev.1.10, up to 50 MHz Clock) as well as default speed mode (SD rev.1.01, up to 25 MHz Clock). A user doesn't need to set differently our SD/MMC host controller for mode change between default mode and high speed mode. If you want to change mode to high-speed mode from default mode and vice versa, by using switch-function command (CMD6), the SD/MMC cards are set to such mode. The timing diagram shows the input/output timing criterion referenced to SD/MMC clock.

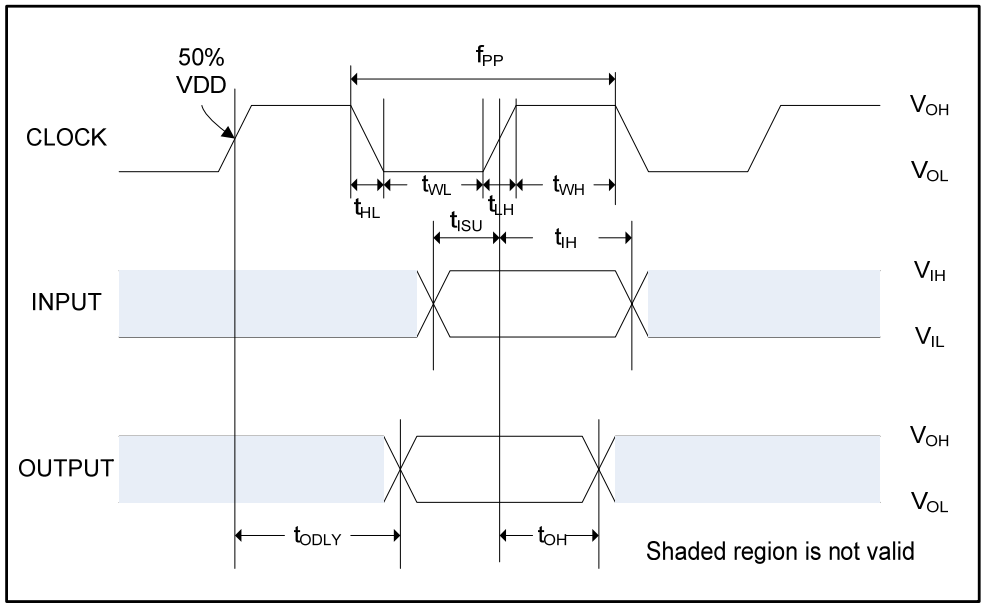


Figure 5.8 Timing Diagram for SD/MMC Controller

Table 5.24 Timing Parameters for Each Symbol

Parameter	Symbol	Min	Max	Unit	Remark
Clock frequency Data Transfer Mode	fPP	0	50	MHz	Ccard ≤ 30pF
Clock low time	tWL	7		ns	Ccard ≤ 30pF
Clock high time	tWH	7		ns	Ccard ≤ 30pF
Clock rise time	tLH		3	ns	Ccard ≤ 30pF
Clock fall time	tHL		3	ns	Ccard ≤ 30pF
Input set-up time	tISU	6		ns	Ccard ≤ 30pF
Input hold time	tIH	2.5		ns	Ccard ≤ 30pF
Output delay time	tODLY	10		ns	Ccard ≤ 30pF
Output hold time	tOH	2		ns	Ccard ≤ 30pF

5.16 Electrical Characteristics for I2C Controller

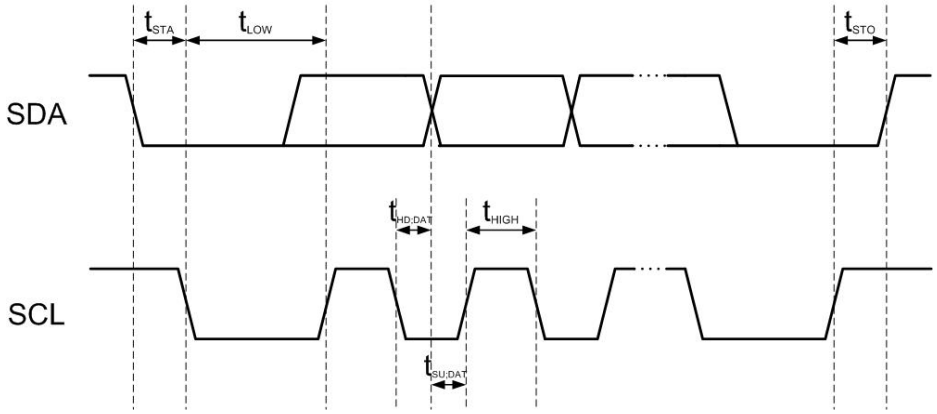


Figure 5.9 Timing Diagram for I2C Controller

Table 5.25 Timing Parameters for Each Symbol

Parameter	Symbol	Min	Max	Unit	Remark
SCL clock frequency		0	400	KHz	
Hold time(repeated) START condition	tSTA	0.95	-	us	
Data hold time	tHD:DAT	0.9		us	
Data setup time	tSU:DAT	0.4	-	us	
HIGH period of the SCL clock	tHIGH	0.96	-	us	
LOW period of the SCL clock	tLOW	1.4	-	us	
Setup time for STOP condition	tSTO	1.0	-	us	

5.17 Electrical Characteristics for SPDI/F Transmitter

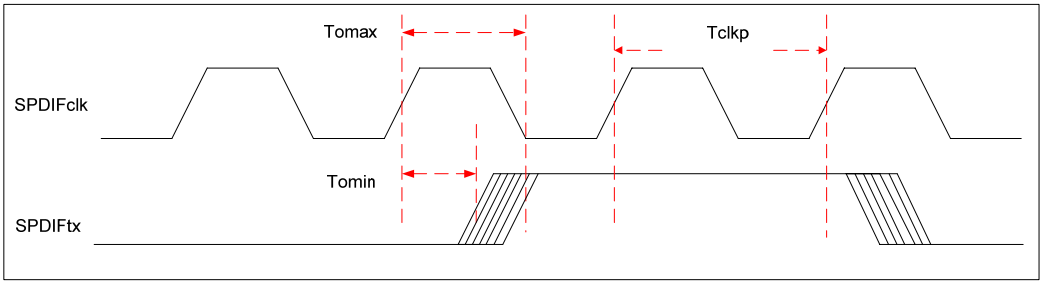


Figure 5.10 Timing Diagram for SPDI/F Transmitter

Table 5.26 Timing Parameters for Each Symbol

Parameter	Symbol	Min	Max	Unit	Remark
SPDIFclk Clock Cycle Time	Tclkp	110		ns	
SPDIFclk Data Output Time Referenced to SPDIFclk	Tomin/Tomax	1	10	ns	CL = 50pF

5.18 Electrical Characteristics for DAI(I2S)

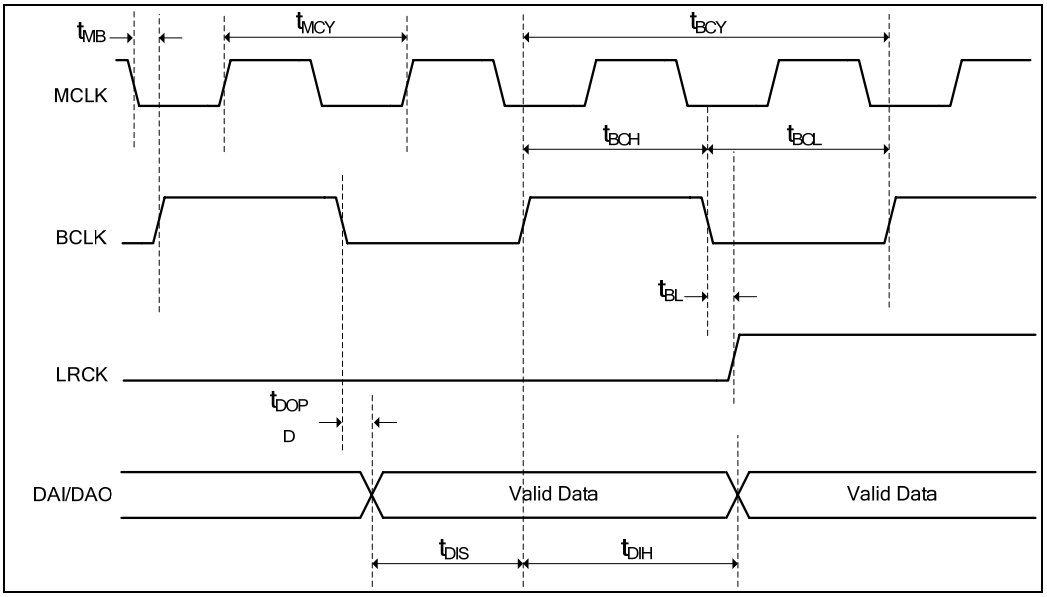


Figure 5.11 Timing Diagram for DAI (receiver)

Test Conditions
MODE = I2S, fs = 48KHz, MCLK = 256fs, BCLK = 64fs

Table 5.27 Timing for DAI (receiver)

Parameter	Symbol	Min	Typ	Max	Unit	Remark
MCLK cycle time	tMCY	19.40	81.40		ns	
BCLK cycle time	tBCY	4 * tMCY	4 * tMCY		ns	
BCLK pulse width high	tBCH	39	163		ns	
BCLK pulse width low	tBCL	38	162		ns	
MCLK to BCLK	tMB	0.18	0.18		ns	
BCLK to LRCK	tBL	14.77	14.77		ns	
DAI setup time to BCLK rising edge	tDIS	1	1		ns	
DAI hold time from BCLK rising edge	tDIH	1	1		ns	
DAO Output Timing Referenced to BCLK	tDOPD			1	ns	

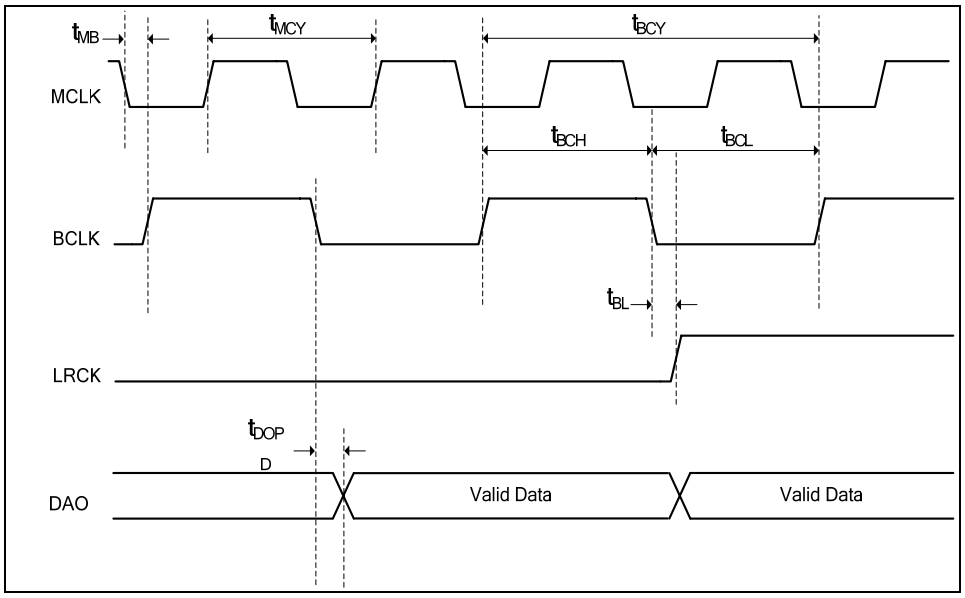


Figure 5.12 Timing Diagram for DAI Transmitter

Table 5.28 Timing Parameters for Each Symbols

Parameter	Symbol	Min	Max	Unit	Remark
MCLK cycle time	tMCY	36		ns	
BCLK cycle time	tBCY	16 * tMCY		ns	
BCLK pulse width high	tBCH	8 * tMCY		ns	
BCLK pulse width low	tBCL	8 * tMCY		ns	
MCLK to BCLK	tMB	19		ns	
BCLK to LRCK	tBL	13		ns	
DAO Output Timing Referenced to BCLK	tDOPD	1		ns	

5.19 Electrical Characteristics for Nand Flash Controller

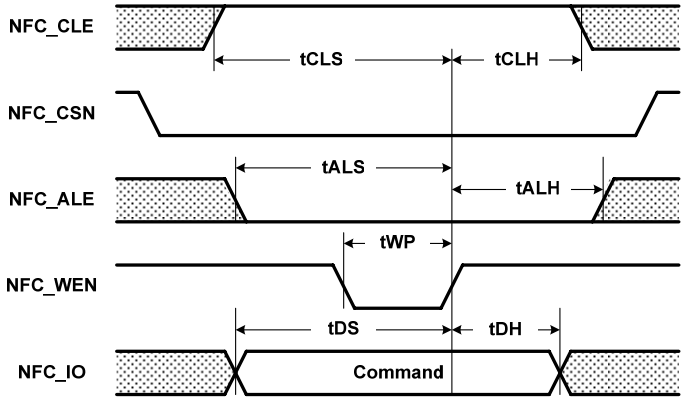


Figure 5.13 Timing Diagram for Command Latch Enable Cycle

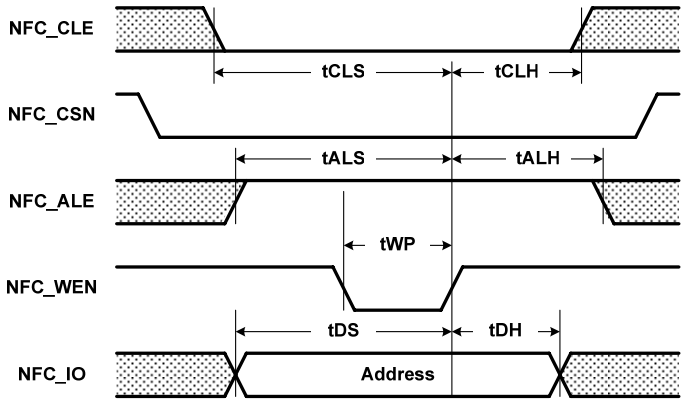


Figure 5.14 Timing Diagram for Single Address Latch Cycle

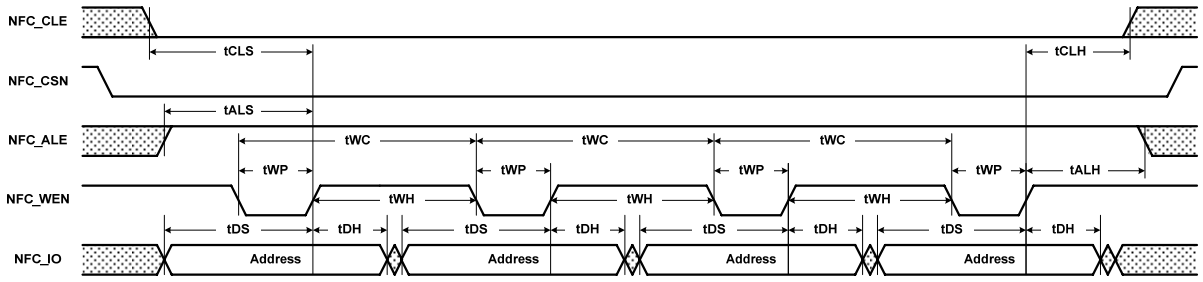


Figure 5.15 Timing Diagram for Linear Address Latch Cycle

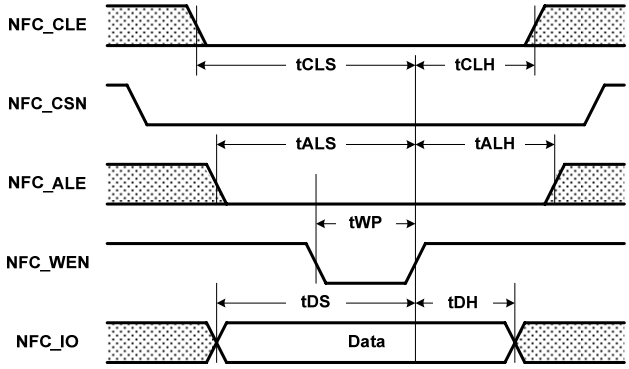


Figure 5.16 Timing Diagram for Single Data Write Cycle

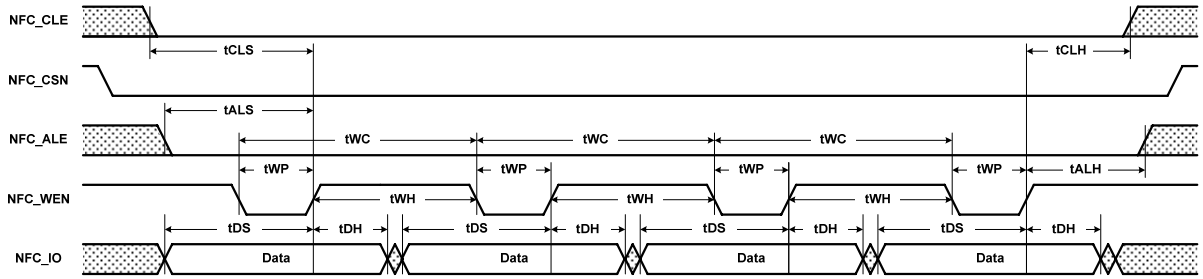


Figure 5.17 Timing Diagram for Linear Data Write Cycle

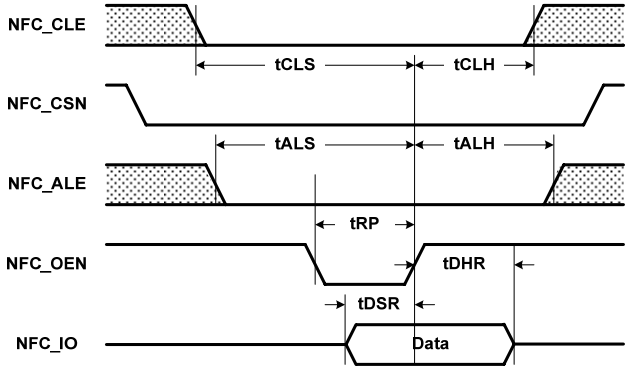


Figure 5.18 Timing Diagram for Single Data Read Cycle

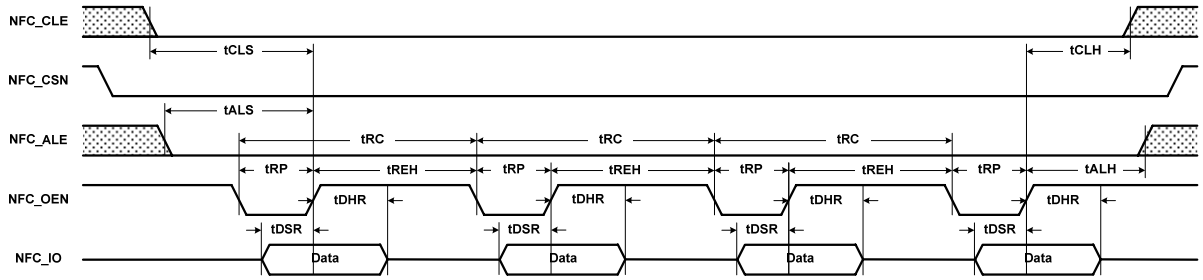


Figure 5.19 Timing Diagram for Linear Data Read Cycle

Table 5.29 Timing Parameters for Each Symbol

Parameter	Symbol	Min	Max	Unit
Clock Period	tHCLK	6		ns
CLE Set-Up Time	tCLS	(STP + PW) x tHCLK + 1.0	(STP + PW) x tHCLK + 2	ns
CLE Hold Time	tCLH	HLD x tHCLK – 2.0	HLD x tHCLK – 1.0	ns
WEN Pulse Width	tWP	PW x tHCLK	PW x tHCLK	ns
WEN High Hold Time	tWH	(STP + HLD) x tHCLK	(STP + HLD) x tHCLK	ns
Write Cycle Time	tWC	(STP + PW + HLD) x tHCLK	(STP + PW + HLD) x tHCLK	ns
OEN Pulse Width	tRP	PW x tHCLK	PW x tHCLK	ns
OEN High Hold Time	tREH	(STP + HLD) x tHCLK	(STP + HLD) x tHCLK	ns
Read Cycle Time	tRC	(STP + PW + HLD) x tHCLK	(STP + PW + HLD) x tHCLK	ns
ALE Set-Up Time	tALS	(STP + PW) x tHCLK – 1.00	(STP + PW) x tHCLK + 2.00	ns
ALE Hold Time	tALH	HLD x tHCLK – 2.00	HLD x tHCLK + 1.00	ns
Data Set-Up Time	tDS	(STP + PW) x tHCLK – 7.00	(STP + PW) x tHCLK – 1.00	ns
Data Hold Time	tDH	HLD x tHCLK – 1.00	HLD x tHCLK + 1.00	ns
Data Set-Up Time in READ	tDSR	5.00	15.0	ns
Data Hold Time in READ	tDHR	0	0	ns

Table 5.30 I/O Function Name for Corresponding Signal Name

Signal Name	I/O Function Name
NFC_CSN	NAND_CSN0, NAND_CSN1
NFC_ALE	NAND_ALE
NFC_CLE	NAND_CLE
NFC_OEN	NAND_OEN
NFC_WEN	NAND_WEN
NFC_IO[15:0]	NANDXD[15:0]

5.20 Electrical Characteristics for UART Controller

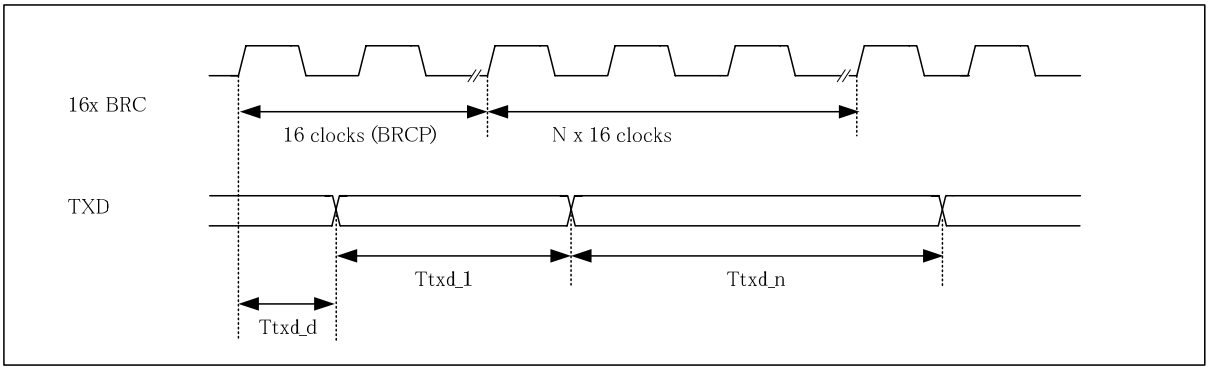


Figure 5.20 Timing Diagram for TXD

Table 5.31 Timing Parameters for Each Symbol

Parameter	Symbol	Min	Max	Unit	Remark
Pulse duration of 1bit TXD	Ttxd_1	BRCP -15	BRCP +15	ns	3.3V
Pulse duration of nbit TXD	Ttxd_n	N x BRCP -15	N x BRCP + 15	ns	3.3V
TXD output delay time	Ttxd_d	0.5	15	ns	3.3V

- BRC : Baud-Rate Clock
- BRCP : Baud-Rate Clock Period
- CL : 71pF

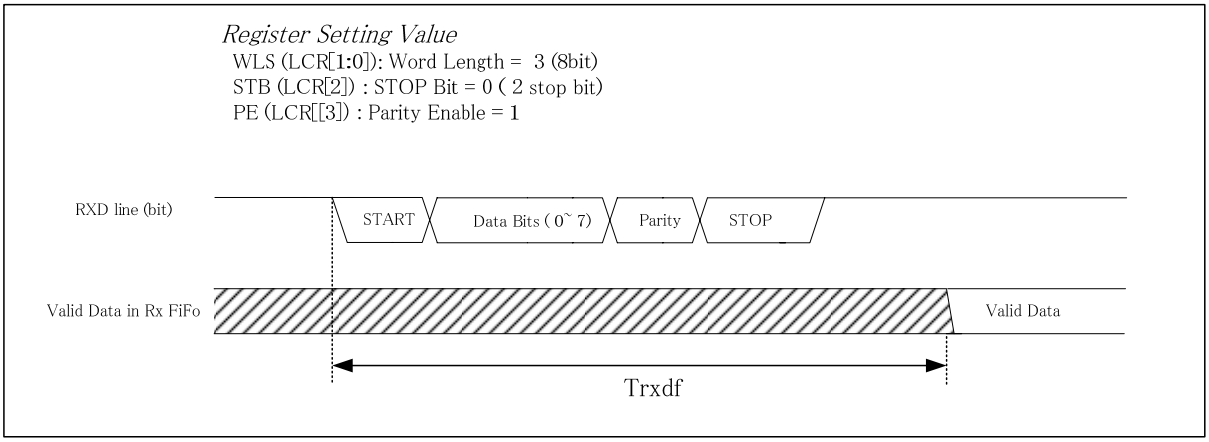


Figure 5.21 Timing Diagram for RXD

Table 5.32 Timing Parameters for Each Symbol

Parameter	Symbol	Min	Max	Unit	Remark
RXD Start to Rx FiFo.	Trxdf	10.5 x BRCP	11 x BRCP	ns	3.3V

- BRC : Baud-Rate Clock
- BRCP : Baud-Rate Clock Period
- CL : 71pF

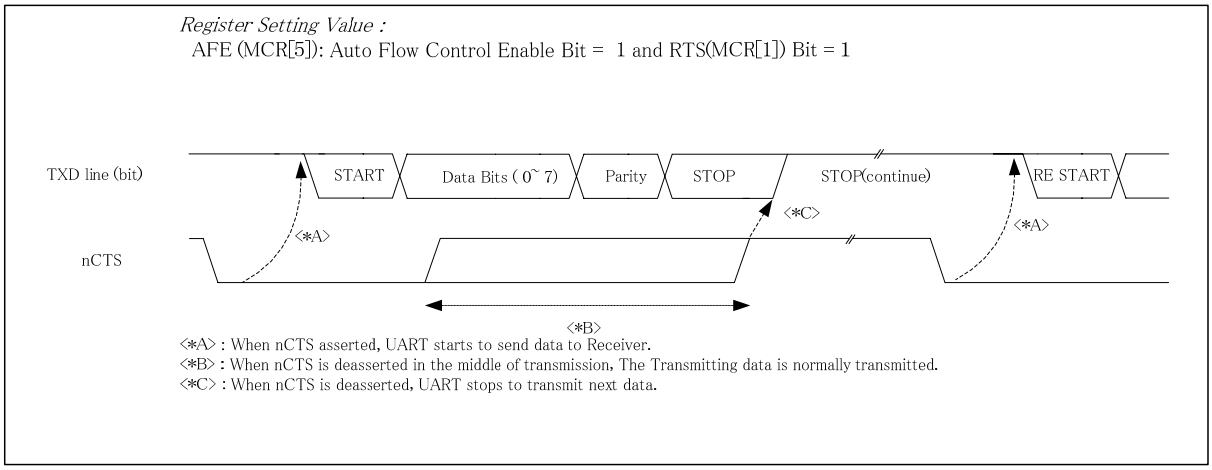


Figure 5.22 Timing Diagram for TX Operation with H/W Flow Control

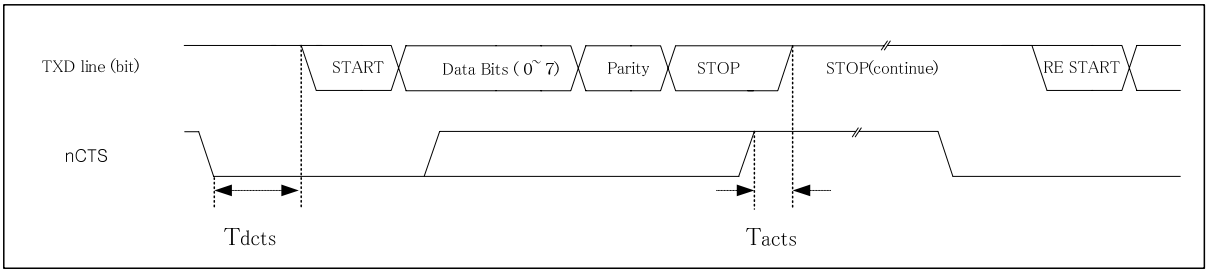


Figure 5.23 Timing Diagram for nCTS Timing Diagram

Table 5.33 Timing Parameters for Each Symbol

Parameter	Symbol	Min	Max	Unit	Remark
Deasserted nCTS to Tx Start	Tdcts	-	BRCP	ns	3.3V
Deasserted nCTS to Tx Stop :to stop next transmission(setup time)	Tacts	4 x BRCP/16	-	ns	3.3V

- BRC : Baud-Rate Clock , BRCP : Baud-Rate Clock Period , CL : 71pF

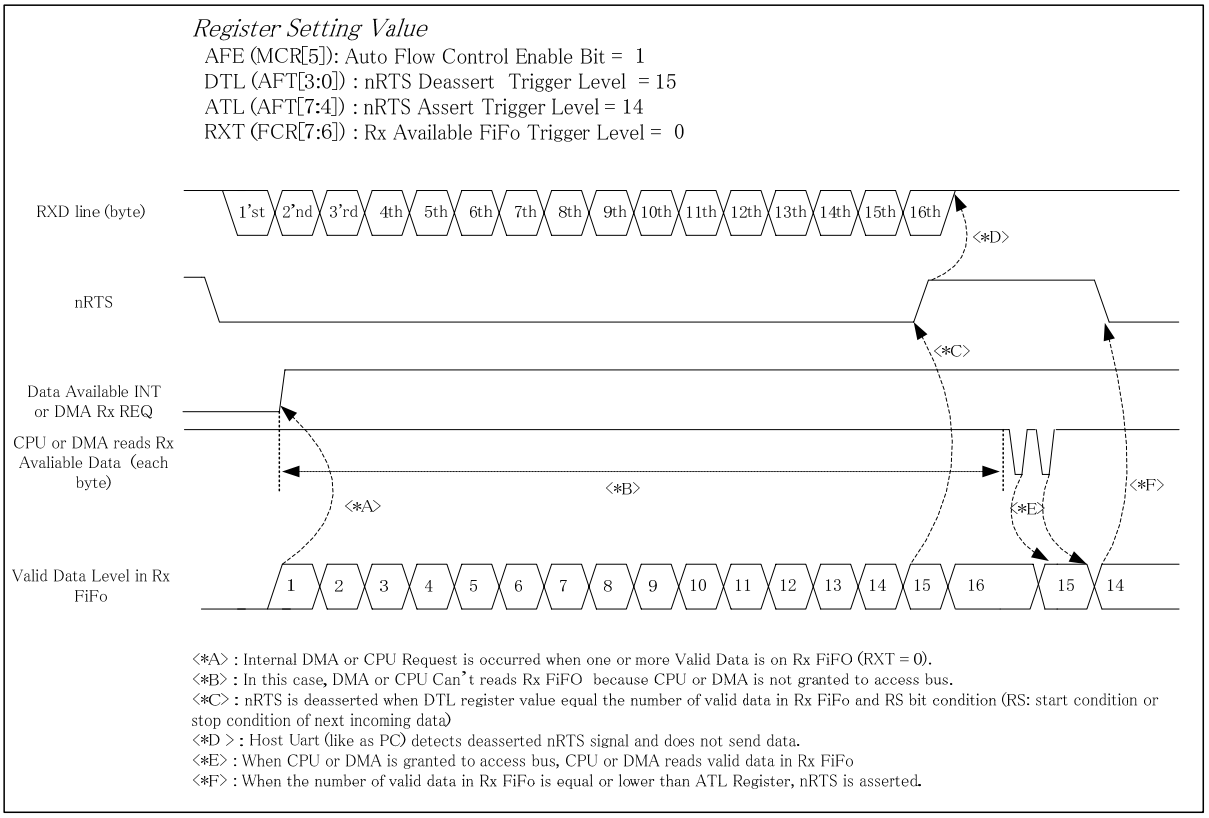


Figure 5.24 Timing Diagram for RX Operation with H/W Flow Control

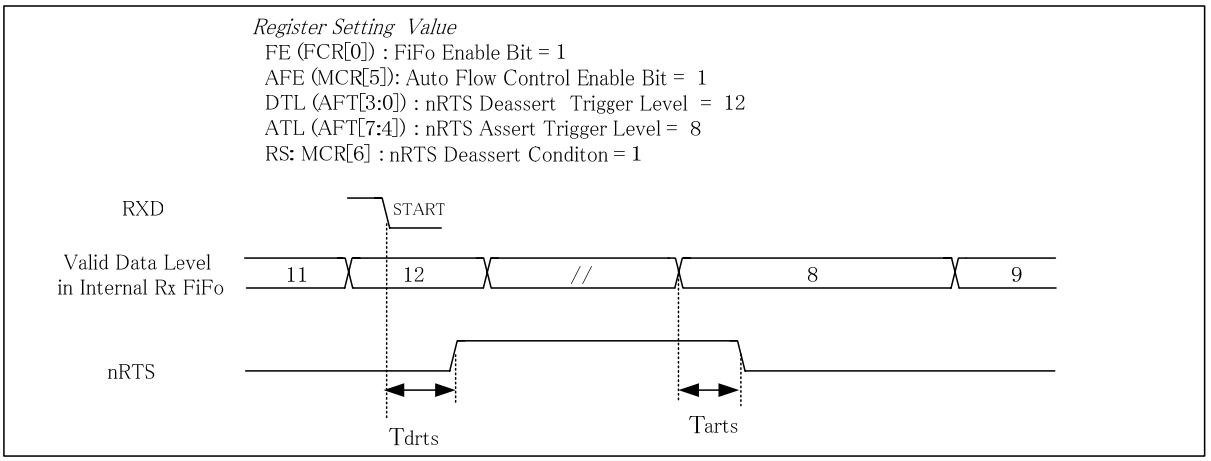


Figure 5.25 Timing Diagram for nRTS Timing Diagram

Table 5.34 Timing Parameters for Each Symbols

Parameter	Symbol	Min	Max	Unit	Remark
DTL(and RXD start condition) to deasserted nRTS	Tdrts	-	BRCP	ns	3.3V
ATL to asserted nRTS	Tarts	-	BRCP/16 + 8	ns	3.3V

- BRC : Baud-Rate Clock , BRCP : Baud-Rate Clock Period , CL : 71pF

5.21 Electrical Characteristics for DDR

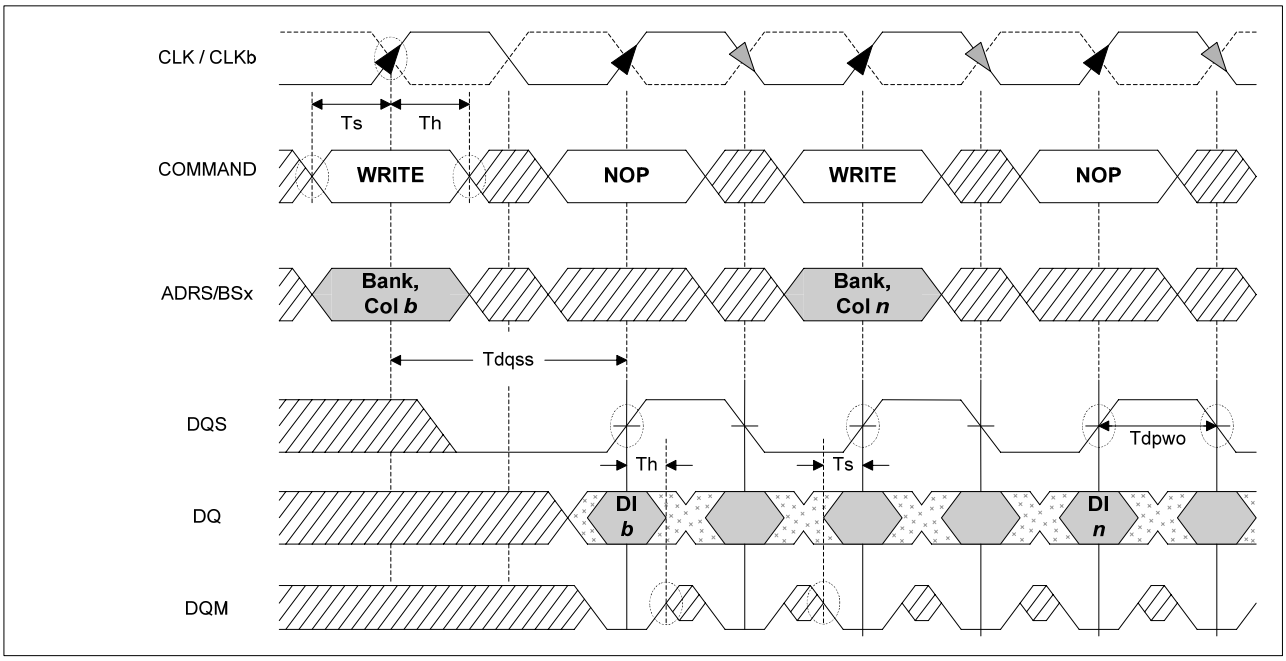


Figure 5.26 Write Cycle Timing

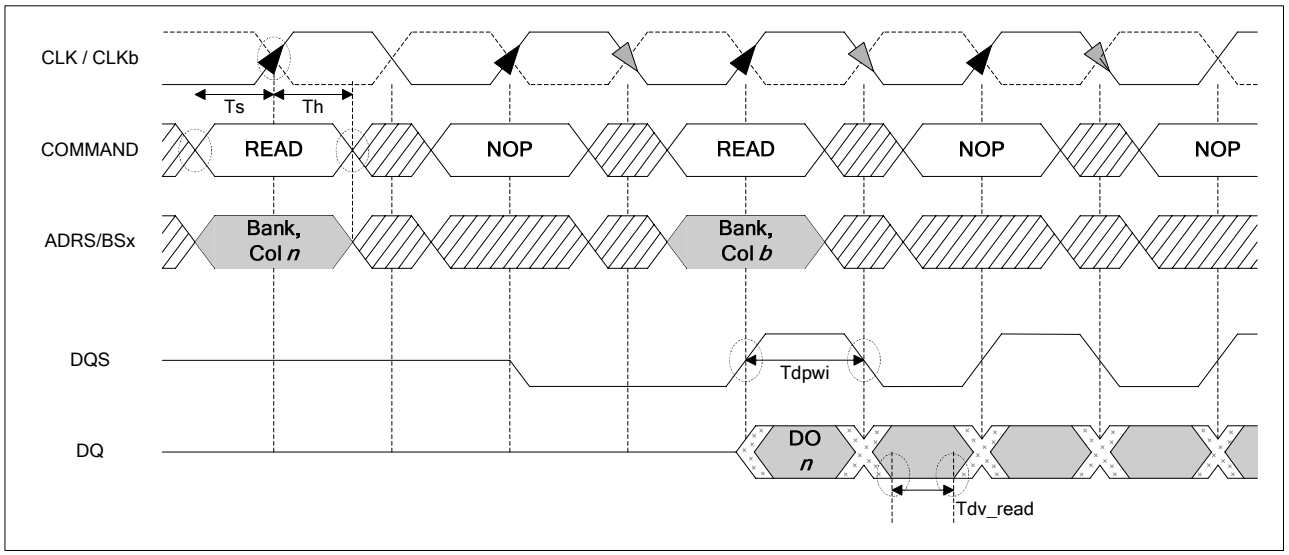


Figure 5.27 Read Cycle Timing

Table 5.35 DDR Interface Timing Parameters

S.No	Parameter	Notation	Min (ns)	Max (ns)
1	Clk Period	tCK	2.5	-
2	Clk High level width	tCH	TBD	-
3	Clk Low level width	tCL	TBD	-
4	RASb output setup time with regard to (w.r.t.) clock	Ts	0.6	-
5	CASb output setup time with regard to clock	Ts	0.6	-
6	WEb output setup time with regard to clock	Ts	0.6	-
7	CKE output setup time with regard to clock	Ts	0.6	-
8	Addr output setup time with regard to clock	Ts	0.6	-
9	BA output setup time with regard to clock	Ts	0.6	-
10	RASb output hold time with regard to clock	Th	0.6	-
11	CASb output hold time with regard to clock	Th	0.6	-
12	WEb output hold time with regard to clock	Th	0.6	-
13	CKE output hold time with regard to clock	Th	0.6	-
14	Addr output hold time with regard to clock	Th	0.6	-
15	BA output hold time with regard to clock	Th	0.6	-
16	DQS output pulse width	Tdpwo	TBD	-
17	DQ output setup time with regard to DQS	Ts	0.105	-
18	DQ output hold time with regard to DQS	Th	0.105	-
19	DQM output setup time with regard to DQS	Ts	0.105	-
20	DQM output hold time with regard to DQS	Th	0.105	-
21	Required input data window for reads (DQ)	Tdv_read	0.144	-
22	Required input DQS pulse width (DQS)	Tdpwi	TBD	-

PART1 - OVERVIEW
TCC8900

High Performance and Low-Power Processor
For Digital Media Applications

Rev. 1.00

Aug 07, 2009

Telechips

Revision History

Date	Revision	Description
2008-12-11	0.00	* Initial Release
2009-08-07	1.00	* Deleting Change Log

TABLE OF CONTENTS

Contents

1 Introduction 1-1
1.1 Feature 1-1
1.2 Block Diagram 1-3
1.3 Pin Descriptions 1-3
2 Address and Register Map 2-5
2.1 Address Map 2-5
3 Bus Architecture 3-7
3.1 Overall Bus Architecture 3-7

Figures

Figure 1.1 TCC8900 Functional Block Diagram 1-3
Figure 3.1 The TCC8900 Bus Architecture 3-7

Tables

Table 2.1 Remap & Address Map of TCC8900 2-5
Table 2.2 Base Address of Bus Components 2-5
Table 3.1 Description of the Clocks from SMU 3-7

1 Introduction

The TCC8900 is the system LSI for digital multimedia applications based on ARM1176JZ(F)-S, ARM's proprietary 32-bit RISC CPU core. It can decode and encode various types of audio/video standards with software and dedicated hardware accelerators – MP3 / AAC / MPEG4-AAC / MPEG4-BSAC, JPEG / MPEG1 / MPEG2 / MPEG4-SP/ASP / H.264 / VC-1 / Real Video and other types of audio/video standard.

The on-chip USB2.0 OTG controller enables the data transmission between a personal computer and storage device such as NAND flash, HDD, CD, SD, MMC and Memory Stick etc, which can be controlled by the TCC8900.

1.1 Feature

- 32 Bits ARM1176JZ(F)-S RISC CPU Core for System and Audio Processing
  - 16KB/16KB Instruction and Data Caches
  - MMU Supported
  - Java Acceleration Supported
  - 16KB Instruction TCM(Tightly Coupled Memory for Exception Handler)
  - 16KB Data TCM (Tightly Coupled Memory for Fast Data Transfer H/W)
  - JTAG interface for code debugging
- MEMORY ORGANIZATION
  - Boot ROM of 16Kbytes for various boot procedure (HPI BOOT, NAND, USB) and security
  - Internal SRAM of 16KB for program shared memory with hardware
  - Memory controller for various memories including PROM, NOR type Flash, SRAM, and SDRAM etc.
- GRAPHIC HARDWARE ACCELERATOR
  - Overlay Mixer
  - 2D / 3D Graphic Engine
- VIDEO HARDWARE ACCELERATOR
  - Supported Video Decoder
    - MJPEG
    - MPEG1, MPEG2
    - MPEG4-SP
    - MPEG4-ASP
    - MPEG4-AVC (H.264)
    - DIVX
    - H.263
    - WMV9
    - VC-1
    - RV
  - Supported Video Encoder
    - MPEG4-SP
    - H.263
    - Bitstream Accelerator
  - Well-Organized Pipelined Architecture Controller
- VIDEO I/O
  - LCD Interface
    - Progressive or Interlaced Digital Video Output
    - Support of STN LCD
    - Support of CCIR-601/656
    - Support of TFT LCD
      - RGB Interface : Up to 24-bit data width
      - CPU Interface : Up to 18-bit data width
    - 16-Level Image Overlay & Chroma-Keying, OSD
    - Color Space Converter
    - Three 256-Level Color Look-Up Table
    - Dual-LCD Controller (CPU Interface only)
  - Camera Interface
    - Support of CCIR-601/656 Format
    - Image Effecter

- Hardware Scaler
- 4-Level Image Overlay, Chroma-Keying & Scaler for Preview
- Various Input Format : YCbCr 4:2:2/4:2:0, RGB 4:2:2, 4:2:0, Bayer RGB
- Color Space Dispatcher
- Up-to 120MHz Pixel Clock

- NTSC/PAL composite output
  - NTSC-M/4.43, PAL-B/D/G/H/I/M/N/Combination N
  - 10-bit DAC
- HDMI Host Interface

- AUDIO I/O
  - I2S Master & Slave Interface
  - S/PDIF TX Interface
- HIGH SPEED INTERFACE
  - High speed USB2.0 OTG / USB1.1 host
  - 2-Channel Host port interface via 80/68 compatible – 8 / 16 bits
- STORAGE I/F
  - UDMA33/66, PIO, IDE Interface for HDD
  - SATA Host Interface
  - Various Memory Card Interfaces for NAND / SD / MMC / Memory Stick etc.
- MISC. INTERFACE
  - 6-Channel UART for serial host interface
  - Configurable 2-Channel I2C
    - 2-Channel Masters
    - 1 Channel Master and 1 Channel I2C Slave
  - 6-Channel GPSB with TS interface for supporting various serial interfaces
  - 2-Channel dedicated TS parallel interface
- ON-CHIP PERIPHERALS
  - Four 16bit timer with PWM output/counters and one 32bit timer
  - 12 Channel DMA for transferring bulk data
    - Including Hardware DMA Request for Each Channel
  - Touch Screen (or 8 channel General purpose) 12-bit ADC
  - RTC
  - 2 Channel Clock Outputs Controlled by Clock Generator
  - CAN Controller
  - MPEFEC
- PROCESS & ELECTRICAL CHARACTERISTICS
  - 65nm CMOS process

1.2 Block Diagram

The following figure shows the block diagram of the TCC8900.

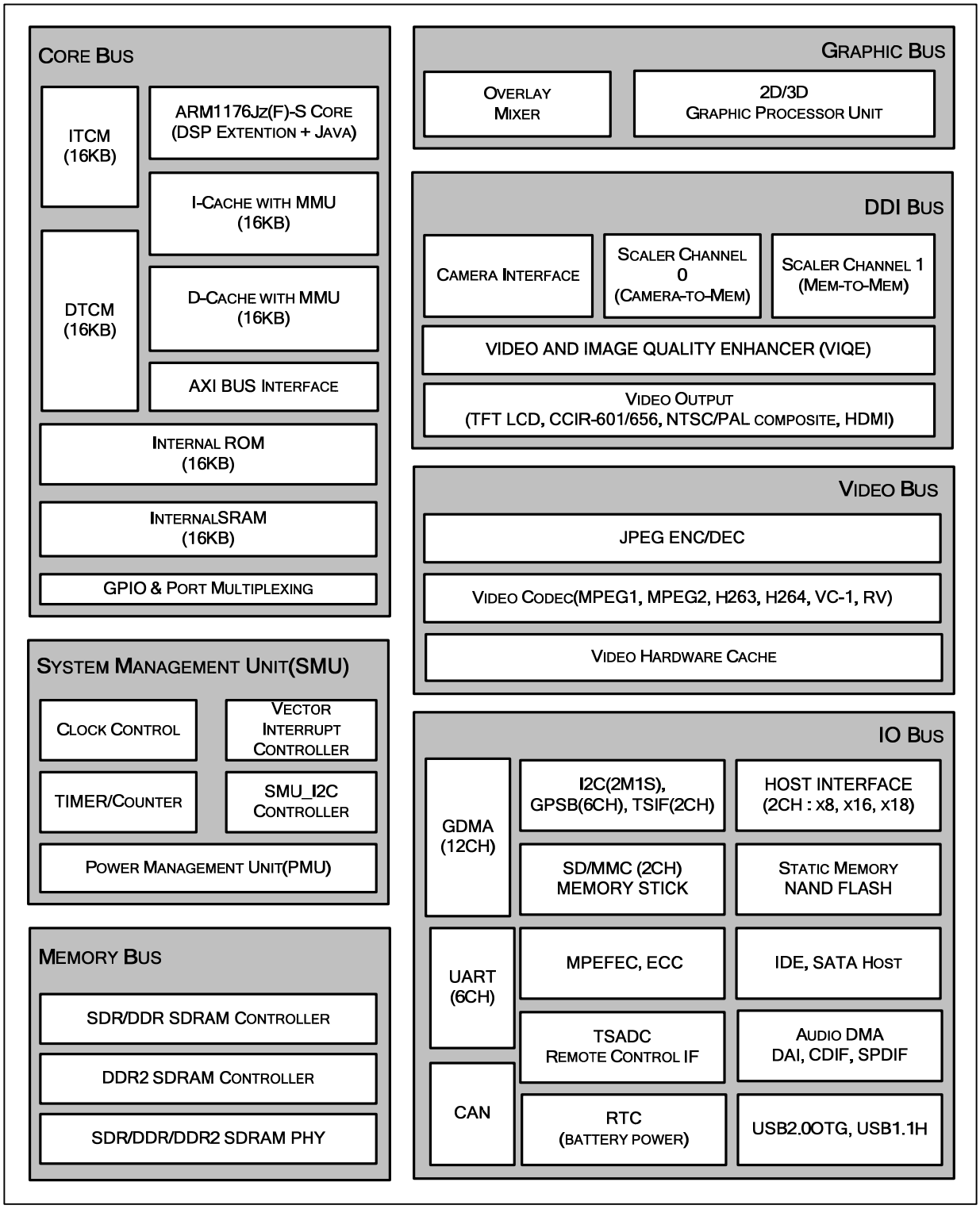


Figure 1.1 TCC8900 Functional Block Diagram

1.3 Pin Descriptions

The TCC8900 is a CMOS device. Floating level on input signals cause unstable device operation and abnormal current consumption. Pull-up or pull-down resistors should be used appropriately for input or bidirectional pins.

2 Address and Register Map

2.1 Address Map

The TCC8900 has fixed address maps for on-chip resources and off-chip resources. The following table represents overall address space of system.

Table 2.1 Remap & Address Map of TCC8900

Address Space	Region	Device Name
0x00000000 – 0x0FFFFFFF	R.0	This region is remapped to as follows. 1) If Remap is 00b, On-chip boot-ROM for region R.E 1) If Remap is 01b, On-chip memory for region R.1 2) If Remap is 10b, Off-chip SDRAM for region R.4 3) If Remap is 11b, this region is not remapped to any region.
0x00000000 – 0x00003FFF		Special Region for Instruction TCM
0x10000000 – 0x10003FFF	R.1	Assigned to on-chip 16kB memory Region
0x40000000 – 0x4FFFFFFF	R.4	Assigned to Off-chip SDRAM chip
0x50000000 – 0x5FFFFFFF	R.5	Assigned to Off-chip SDRAM chip
0x60000000 – 0x6FFFFFFF	R.6	Assigned to Off-chip SDRAM chip
0x70000000 – 0x7FFFFFFF	R.7	Assigned to Off-chip SDRAM chip
0xA0000000 – 0xA0003FFF	R.A	Assigned to DTCM memory region
0xE0000000 – 0xE0003FFF	R.E	Assigned to internal boot ROM
0xF0000000 – 0xFFFFFFFF	R.F	Assigned to on-chip peripherals

The address space (0x00000000 ~ 0x0FFFFFFF) is initially allocated to internal or external memory for booting procedure, and a special flag exists in system controller unit for remapping this space to other type of memories. Refer to the description of system controller for detailed operation.

The TCC8900 has various peripherals for specific interface controllers or on-chip hardware components. These peripherals can be configured appropriately by its own registers that can be accessed through specially allocated address. These address maps are represented in the following table. In case of memory controller, its space is separated for preventing illegal accessing.

Refer to corresponding sections for detail information of each peripheral.

Table 2.2 Base Address of Bus Components

Base Address	Peripherals
0xF0000000	Graphic Bus Components
0xF0100000	Memory Bus Components
0xF0200000	DDI Bus Components
0xF0300000	Memory Bus Components
0xF0400000	System Control and Management Registers
0xF0500000	IO Bus Peripherals
0xF0600000	MMU Table
0xF0700000	Video Bus Components

3 Bus Architecture

3.1 Overall Bus Architecture

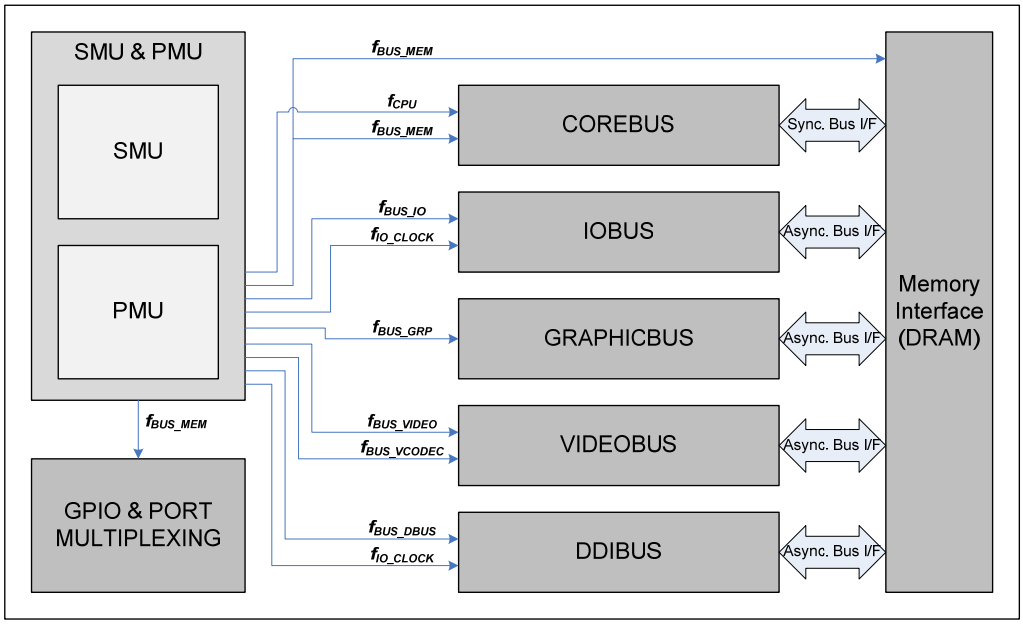


Figure 3.1 The TCC8900 Bus Architecture

The TCC8900 is composed of 9 hardware groups shown in the above figure. The SMU is the hardware blocks to manage the overall system such as the clock generators, interrupt controller, timer and etc and the PMU (Power-Management Unit) can controls the operating modes. The GPIO can configure the function and characteristics of all the ports.

The DDIBUS includes the video input/output hardwares such as LCD controller, camera interface. The main processor, boot-ROM, and internal SRAM are in the COREBUS. The VIDEOBUS has the compressor and decompressor for the various video standards, such as JPEG enc/dec, video codec. the GRAPHICBUS can handle the video image and RGB image. Finally, various interfaces to external devices are included in the IOBUS, such as SD/MMC controller, SPI hardware, and etc.

The following table shows the description of the clocks generated by the SMU in the above figure.

Table 3.1 Description of the Clocks from SMU

Clock Name	Description
f_{CPU}	* ARM1176JZF-S Operating Clock
f_{BUS_MEM}	* Bus Interface Clock for ARM1176JZF-S * GPIO Register Interface Clock * External DRAM Interface Clock
f_{BUS_IO}	* Bus Interface Clock of the I/O Bus
f_{BUS_GRP}	* Bus Interface Clock of the Graphic Bus
f_{BUS_VIDEO}	* Bus Interface Clock of the Video Bus
f_{BUS_DBUS}	* Bus Interface Clock of the Display Device Interface Bus
f_{IO_CLOCK}	* Peripheral Clocks for the Internal Hardware
f_{BUS_VCODEC}	* Core Clock for the Hardwired Video Codec

PART2 – SMU & PMU
TCC8900

High Performance and Low-Power Processor

For Digital Media Applications

Rev. 1.02

Oct 09, 2009

Telechips

Revision History

Date	Revision	Description
2008-12-11	0.00	* Initial Release
2009-02-10	0.01	* The number of interrupt sources and external interrupt sources has been changed from (32,4) to (64,12) * The SOURCES[31:0] was changed to SOURCE[63:0] in the Figure 2.1 * The function description for the POFF in PMU was modified with additional information in the PMU. * The section "Enter/Exit DEEP-POWER-DOWN Mode with ITCM Booting" was added in the PMU. * The section "Power On/Off Sequence for the HDMI PHY" in the PMU. * The section "Power On/Off Sequence for the SATA PHY" in the PMU. * The section "Power On/Off Sequence for the USB PHY" in the PMU. * The document for the BOOT PROCEDURE has been changed.
2009-02-25	0.02	* The description of the “PD” field of the PLLnCFG register was modified. * The base addresses for the PRI8 ~ PRI15 in the VIC were changed. * The section "Power On/Off Sequence for the LVDS PHY" in the PMU. * The bit field of the “VAIRQ, VAFIQ” was extended from “[6:0]” to “[8:0]” * The bit field of the “VNIRQ, VNFIQ” was extended from “[4:0]” to “[6:0]”
2009-05-28	0.03	* The register description from DPORT and ICLK(not including ICLK) was removed. * The bit-field of VIC corresponding to the JPEG Encoder was changed from 21 to 22. * The bit-field of the VIC corresponding to the JPEG Decoder was changed from 22 to 21. * The base address of the EINTSEL0 was changed from 0xf0102180 to 0xf0102184 * The base address of the EINTSEL1 was changed from 0xf0102184 to 0xf0102188 * The base address of the EINTSEL2 was changed from 0xf0102188 to 0xf010218C * The description for the WAKEUP sources were updated. * The description for the external interrupt sources numbered by 52, 53, 54, 55 was changed 52 : GPIOB[4] → GPIOE[4] 53 : GPIOB[5] → GPIOE[5] 54 : GPIOB[24] → GPIOE[24] 55 : GPIOB[25] → GPIOE[25] * The AISO was changed to ASION of the CONTROL register in the PMU. The active condition was changed.
2009-08-07	1.00	* Insert Nand V2 Boot mode * Delete memory initialize parameter * Add SD MMC Port *The description for the SWRESET register has been changed *The power management procedures for the video bus, graphic bus, DDI bus were added.
2009-09-04	1.01	* The timing parameter for PMU was added.
2009-10-09	1.02	* The more detailed information about the CKC, VPIC, Timer, Boot Procedure were added.

TABLE OF CONTENTS

Contents

1 CKC ... 1-1
1.1 Overview ... 1-1
1.2 Signal Descriptions ... 1-3
1.2.1 Global Signals ... 1-3
1.2.2 Clock Source Signals ... 1-3
1.2.3 CPU Related Signals ... 1-3
1.2.4 DDI Bus Related Signals ... 1-3
1.2.5 Memory Bus Related Signals ... 1-3
1.2.6 Graphic Bus Related Signals ... 1-4
1.2.7 I/O Bus Related Signals ... 1-4
1.2.8 Video Bus Related Signals ... 1-4
1.2.9 SMU Bus Related Signals ... 1-4
1.2.10 Peripheral Clock Output Signals ... 1-5
1.3 Register Descriptions ... 1-6
1.4 Operation & Timing Diagram ... 1-12
1.4.1 Clock Change Operation (Safe Clock Changer) ... 1-12
1.4.2 How to Generate "F_{CPU} " through BCLKOUT[0] ... 1-12
1.4.3 How to Generate BCLKOUT[1] ~ BCLKOUT[7] ... 1-13
1.4.4 How to Generate PCLKs[36:0] ... 1-14
1.4.5 Procedure for Configuration of Peripheral Clocks ... 1-15
1.4.6 Reference PMS Table for Target Frequency ... 1-16
2 VPIC(Vectored Priority Interrupt Controller) ... 2-25
2.1 Overview ... 2-25
2.2 Signal Descriptions ... 2-26
2.2.1 Global Signals ... 2-26
2.2.2 Interrupt Source Signals ... 2-26
2.2.3 Interrupt Output Signals (to ARM) ... 2-26
2.2.4 Timer Related Signals ... 2-26
2.3 Register Descriptions ... 2-27
2.3.1 Priority Interrupt Controller ... 2-28
2.3.2 Vectored Interrupt Controller ... 2-37
2.4 Operation & Timing Diagram ... 2-40
2.4.1 How to Configure the Interrupt Source ... 2-40
2.4.2 How to Enable Interrupt for IRQI ... 2-41
2.4.3 Recommended IRQI Configurations ... 2-41
2.4.4 How to Use Vectored Interrupts ... 2-43
3 Timer / Counter ... 3-45
3.1 Overview ... 3-45
3.2 Signal Descriptions ... 3-47
3.2.1 Global Signals ... 3-47
3.2.2 T-Timer/Counter(16/20bits Timer/Counter) Related Signals ... 3-47
3.2.3 X-Timer (Watchdog Timer) ... 3-47
3.2.4 Z-Timer (32bits Timer) ... 3-47
3.3 Register Description ... 3-48
3.4 Operation & Timing Diagram ... 3-54
3.4.1 How to Run 16bits Timer ... 3-54
3.4.2 How to Run 20bits Timer ... 3-55
3.4.3 How to Run External Clock Counter ... 3-55
3.4.4 How to Run Watchdog Timer ... 3-55
3.4.5 How to Run 32bits Timer ... 3-56
3.4.6 Interrupt Structure for IREQ[0] – 16/20bits/Watchdog Timer/Counter, ... 3-56
3.4.7 Interrupt Structure for IREQ[1] – 32bits Timer/Counter ... 3-56
4 PMU (power management unit) ... 4-57
4.1 Overview ... 4-57
4.2 Register Overview ... 4-58
4.3 Register Descriptions ... 4-59
4.4 Operation & Timing Diagram ... 4-64
4.4.1 Power Management Scheme ... 4-64
4.4.2 Operating Mode Definitions ... 4-65
4.4.3 Basic Power-Up Sequence ... 4-67
4.4.4 Basic Power-Off Sequence – Turn-Off ... 4-68
4.4.5 Enter POWER-ON State from Initial Boot-Up or RTC PMWKUP ... 4-69
4.4.6 Enter POWER-OFF and DEEP-POWER-DOWN Mode ... 4-70
4.4.7 WAKE-UP Event Configuration ... 4-71
4.4.8 Exit POWER-OFF and DEEP-POWER-DOWN Mode ... 4-72

4.4.9 Enter/Exit POWER-DOWN Mode 4-73
4.4.10 Enter and Exit DEEP-POWER-DOWN Mode with ITCM Booting 4-73
4.4.11 Power On/Off Sequence for the Video Bus 4-75
4.4.12 Power On/Off Sequence for the Graphic Bus 4-76
4.4.13 Power On/Off Sequence for the DDI Bus 4-77
4.4.14 Power On/Off Sequence for the HDMI PHY 4-78
4.4.15 Power On/Off Sequence for the SATA PHY 4-79
4.4.16 Power On/Off Sequence for the USB PHY 4-80
4.4.17 Power On/Off Sequence for the LVDS PHY 4-81
5 SMU_I2C 5-83
5.1 Overview 5-83
5.2 Register Descriptions 5-84
6 Boot Procedure 6-87
6.1 Power Up/Down Sequence for Core and I/O Power 6-87
6.2 Boot Mode 6-88
6.3 Overall Procedure 6-90
6.4 EHI Boot (BM[2:0] = 000b) 6-91
6.5 USB Boot (BM == 011b) 6-93
6.6 External NOR Boot (BM == 110b, GPIOE[4] == 0b) 6-94
6.7 NFC NAND Boot (BM == 111b, GPIOE[0] = 0b) 6-95
6.8 I2C Master Boot – EEPROM Boot (BM == 001b) 6-98
6.9 Serial Flash Boot (BM==010b) 6-99
6.10 SPI Slave Boot (BM==100b) 6-101
6.11 UART Boot (BM==110b, GPIOE[4]==1b) 6-102
6.12 SD/MMC Boot (BM==101b) 6-104
6.13 NAND_V2 Boot (BM==111b, GPIOE[0]= 1b) 6-106
6.14 Dual CRC Checking 6-108

Figures

Figure 1.1 CKC Block Diagram 1-1
Figure 1.2 Clock Change Timing Diagram 1-12
Figure 1.3 Clock Chain for BCLKOUT[0] 1-12
Figure 1.4 Clock Change Timing Diagram for BCLKOUT[0] 1-13
Figure 1.5 Clock Chain for BCLKOUT[0] 1-13
Figure 1.6 Clock Change Timing Diagram for BCLKOUT[7:1] 1-13
Figure 1.7 Clock Chain for PCLKs[36:0] 1-14
Figure 1.8 Clock Change Timing Diagram for BCLKOUT[7:1] 1-14
Figure 1.9 Peripheral Clock Configuration Procedure 1-15
Figure 2.1 Block Diagram of Vectored Interrupt Controller 2-25
Figure 2.2 Detailed Interrupt Flow 2-25
Figure 2.3 Active High vs. Active Low 2-40
Figure 2.4 Level Triggered vs. Edge Triggered 2-40
Figure 2.5 A Case of Missing the Edge-Triggered IRQI 2-40
Figure 2.6 The timing Relations between Original IRQI and Synchronized IRQI 2-41
Figure 2.7 Structure of the Non-Vectored Interrupt Handler 2-43
Figure 2.8 Structure of the Vectored Interrupt Handler 2-44
Figure 2.9 Example Code of the Vectored Interrupt Handler 2-44
Figure 3.1 Overall Timer/Counter Block Diagram 3-45
Figure 3.2 16-bit and 20bit Timer/Counter Block Diagram 3-45
Figure 3.3 Watchdog Timer Block Diagram 3-46
Figure 3.4 32-bit Counter Block Diagram 3-46
Figure 3.5 The Basic Timing Diagram of the Timer 3-54
Figure 3.6 The Waveform in Case of STOP being ‘1’ 3-54
Figure 3.7 Timing Diagram of Timer/Counter 3-55
Figure 3.8 IREQ 3-56
Figure 4.1 PMU Block Diagram 4-57
Figure 4.2 Overall Block Diagram 4-64
Figure 4.3 Operating Modes 4-65
Figure 4.4 Power-Up Sequence 4-67
Figure 4.5 Power-Off Sequence 4-68
Figure 4.6 Timing Diagram for Entering POWER-ON 4-69
Figure 4.7 Timing Diagram for Entering POWER-OFF and DEEP-POWER-DOWN Mode 4-70
Figure 4.8 Timing Diagram for Exiting POWER-OFF and DEEP-POWER-DOWN Mode 4-71
Figure 4.9 Timing Diagram for Exiting POWER-OFF and DEEP-POWER-DOWN Mode 4-72
Figure 4.10 Example Flow Chart to Enter Deep-Power-Down Mode 4-73
Figure 4.11 Example Flow Chart to Exit Deep-Power-Down Mode 4-74
Figure 4.12 Power-On/Off Sequency for the Video Bus 4-75

Figure 4.13 Video Bus Power Management for Entering/Exiting the POWER-OFF or DEEP-POWER-DOWN.... 4-75
Figure 4.14 Power-On/Off Sequency for the Graphic Bus 4-76
Figure 4.15 Graphic Bus Power Management for Entering/Exiting the POWER-OFF or DEEP-POWER-DOWN 4-76
Figure 4.16 Power-On/Off Sequency for the DDI Bus..... 4-77
Figure 4.17 DDI Bus Power Management for Entering/Exiting the POWER-OFF or DEEP-POWER-DOWN..... 4-77
Figure 4.18 Power-On/Off Sequency for the HDMI PHY..... 4-78
Figure 4.19 Power-On/Off Sequency for the SATA PHY 4-79
Figure 4.20 Power-On/Off Sequency for the USB PHY 4-80
Figure 4.21 Power-On/Off Sequency for the LVDS PHY..... 4-81
Figure 5.1 I2C Block Diagram..... 5-83
Figure 6.1 Power Up/Down Sequence for Core and I/O Power 6-87
Figure 6.2 Reset Sequence 6-89
Figure 6.3 Overall Flowchart of Boot Code 6-90
Figure 6.4 INITCFG Bit Field..... 6-91
Figure 6.5 BIP Data Structure 6-91
Figure 6.6 External Host Uploads Program when BIPEN = 1 6-91
Figure 6.7 EHI Boot Procedure..... 6-92
Figure 6.8 USB Boot Procedure..... 6-93
Figure 6.9 External NOR Boot Procedure..... 6-94
Figure 6.10 NAND Boot Procedure..... 6-96
Figure 6.11 Basic Structure and Boot Flow of NAND Boot..... 6-97
Figure 6.12 I2C Boot Procedure 6-98
Figure 6.13 Data Structure for Header Information Stored In Serial Flash 6-99
Figure 6.14 Pseudo Code for Serial Flash Booting Procedure..... 6-100
Figure 6.15 SPI Slave Boot Procedure 6-101
Figure 6.16 UART Boot Procedure 6-103
Figure 6.17 SD/MMC Boot Parameter 6-104
Figure 6.18 SD/MMC Boot Procedure 6-105
Figure 6.19 NAND_V2 Boot Procedure 6-106
Figure 6.20 NAND_V2 Golden Info. Structure 6-107

Tables

Table 1.1 Bus Clocks and Description..... 1-1
Table 1.2 CKC Register Map (Base Address = 0xF0400000) 1-2
Table 1.3 Global Signals 1-3
Table 1.4 Clock Source Signals 1-3
Table 1.5 CPU Related Signals..... 1-3
Table 1.6 DDI Bus Related Signals 1-3
Table 1.7 Memory Bus Related Signals 1-3
Table 1.8 Graphic Bus Related Signals..... 1-4
Table 1.9 I/O Bus Related Signals 1-4
Table 1.10 Video Bus Related Signals 1-4
Table 1.11 SMU Bus Related Signals 1-4
Table 1.12 Video Bus Related Signals 1-5
Table 1.13 PLL1/2/3 Configuration Examples 1-16
Table 1.14 PLL0 Configuration Examples 1-19
Table 2.1 Global Signals 2-26
Table 2.2 Clock Source Signals 2-26
Table 2.3 Interrupt Output Signals (to ARM) 2-26
Table 2.4 Interrupt Output Signals (to ARM) 2-26
Table 2.5 Priority Interrupt Controller Register Map (Base Address = 0xF0401000)..... 2-27
Table 2.6 Vectored Interrupt Controller Register Map (Base Address = 0xF0401200)..... 2-27
Table 2.7 Sources of External Interrupt..... 2-29
Table 2.8 Recommended IRQI Configurations..... 2-41
Table 2.9 Recommended External Interrupt Configurations..... 2-42
Table 3.1 Global Signals 3-47
Table 3.2 T-Timer/Counter(16/20bits Timer/Counter) Related Signals 3-47
Table 3.3 X-Timer (32bits Timer)..... 3-47
Table 3.4 Interrupt Output Signals (to ARM) 3-47
Table 3.5 Timer/Counter Register Map (Base Address = 0xF0403000) 3-48
Table 3.6 TC32 Count Mode 3-52
Table 4.1 PMU Register Map (Base Address = 0xF0404000)..... 4-58
Table 5.1 SMU_I2C Register Map (Base Address = 0xF0405000) 5-84
Table 6.1: Configuration Value 6-88
Table 6.2: Frequency of f_{MBUS}..... 6-88
Table 6.3 Supported NAND Flash Types..... 6-95
Table 6.4 Each Mode as NAND Types..... 6-96

1 CKC

1.1 Overview

The block diagram of CKC is shown in Figure 1.1.

The CKC block has 8 primary clock sources from 4 PLLs, XIN, XTIN, SATA XI, HDMI XI. Four PLL clock sources can generate four divided clock for corresponding clock divider. XIN and XTIN can generate divided clock. These 8 primary clock sources and various divided clock sources can be used for generating the CPU clock, the bus clock, and each peripheral clocks.

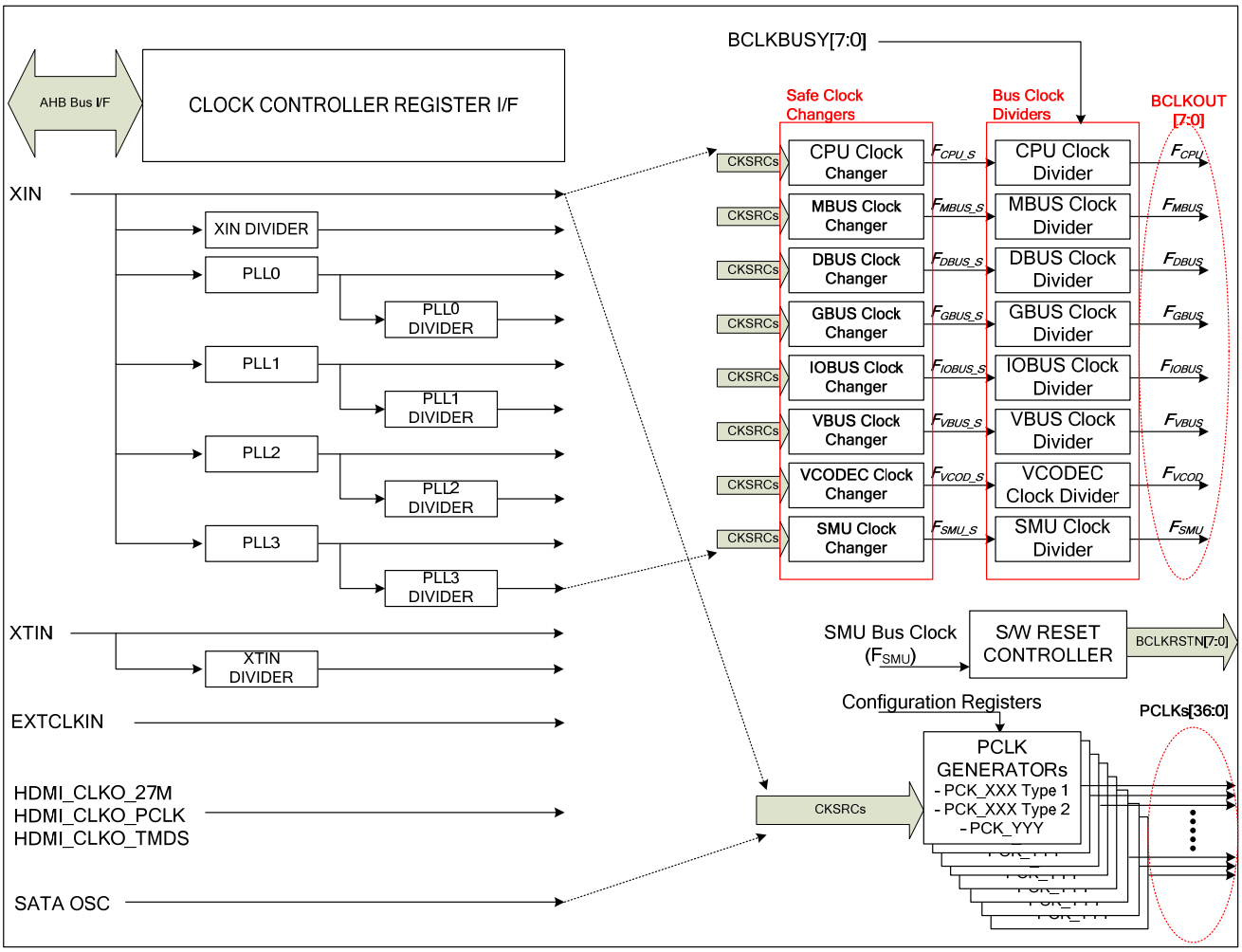


Figure 1.1 CKC Block Diagram

The output clock of safe clock changer, which generates glitch-free clock from 8 independent clock sources, can be used for making the main operating clocks (CPU clock and bus clocks). The one CPU clock and 7 bus clocks which are named by F_{CPU}, F_{MBUS}, F_{DBUS}, F_{GBUS}, F_{VBUS}, F_{VCOD}, F_{IOBUS}, F_{SMU}.

Table 1.1 Bus Clocks and Description

Name	Description
F_{CPU}	CPU Clock Frequency
F_{MBUS}	CPU Interface and Memory Bus Clock
F_{DBUS}	Bus Clock for Display Device (Video In/Out + Display Related Hardwares)
F_{GBUS}	Bus Clock for Graphic Device
F_{VBUS}	Bus Clock for Video Processing (Video Codec)
F_{VCOD}	Core Clock for Video Codec (Moving Video Only)
F_{IOBUS}	Bus Clock for I/O Interface Devices
F_{SMU}	Bus Clock for System Management Unit

The peripheral clock generator makes the corresponding hardware clock using 4 primary clock sources and 8 generated clocks.

Table 1.2 CKC Register Map (Base Address = 0xF0400000)

Name	Address	Type	Reset	Description	
CLKCTRL0	0x000	R/W	0x002FF014	CPU Clock Control Register	
CLKCTRL1	0x004	R/W	0x002FF014	Bus Clock Control Register for Display Devices	
CLKCTRL2	0x008	R/W	0x002FF014	Bus Clock Control Register for Memory Interface	
CLKCTRL3	0x00C	R/W	0x002FF014	Bus Clock Control Register for Graphic Devices	
CLKCTRL4	0x010	R/W	0x002FF014	Bus Clock Control Register for I/O Interface Devices	
CLKCTRL5	0x014	R/W	0x002FF014	Bus Clock Control Register for Video Codec	
CLKCTRL6	0x018	R/W	0x002FF014	Core Clock Control Register for Video Codec	
CLKCTRL7	0x01C	R/W	0x002FF014	Bus Clock Control Register for SMU Hardwares	
PLL0CFG	0x020	R/W	0x8100FA03	PLL0 Configuration Register	
PLL1CFG	0x024	R/W	0x80009603	PLL1 Configuration Register	
PLL2CFG	0x028	R/W	0x80007D03	PLL 2 Configuration Register	
PLL3CFG	0x02C	R/W	0x80009603	PLL 3 Configuration Register	
CLKDIVC0	0x030	R/W	0x81818181	PLL0/1/2/3 Divider Configuration Register	
CLKDIVC1	0x034	R/W	0x81818181	XIN/XTIN Divider Configuration Register	
CLKDIVC2	0x038	R/W	0x81818181	Reserved for Future Use	
CLKDIVC3	0x03C	R/W	0x81818181	Reserved for Future Use	
SWRSTPRD	0x040	R/W	0x000000FF	Software Reset Period Configuration Register	
SWRESET	0x044	R/W	0x00000000	Software Reset Control Register	
PCLK_TCX	0x080	R/W	0x14000000	Control Register for Timer Counter X Clock	PCK_XXX 1
PCLK_TCT	0x084	R/W	0x14000000	Control Register for Timer Counter T Clock	PCK_XXX 1
PCLK_TCZ	0x088	R/W	0x14000000	Control Register for Timer Counter Z Clock	PCK_XXX 1
PCLK_LCD0	0x08C	R/W	0x14000000	Control Register for LCD Channel 0	PCK_XXX 2
PCLK_LCD1	0x090	R/W	0x14000000	Control Register for LCD Channel 1	PCK_XXX 2
PCLK_LCDSI	0x094	R/W	0x14000000	Control Register for LCD System Interface	PCK_XXX 2
PCLK_CIFMC	0x098	R/W	0x14000000	Control Register for CIF Internal Clock	PCK_XXX 1
PCLK_CIFSC	0x09C	R/W	0x14000000	Control Register for CIF Scaler Clock	PCK_XXX 1
PCLK_OUT0	0x0A0	R/W	0x14000000	Control Register for External Clock Output 0	PCK_XXX 1
PCLK_OUT1	0x0A4	R/W	0x14000000	Control Register for External Clock Output 1	PCK_XXX 1
PCLK_HDMI	0x0A8	R/W	0x14000000	Control Register for HDMI PHY Input Clock	PCK_XXX 2
PCLK_USB11H	0x0AC	R/W	0x14000000	Control Register for USB 1.1 Host	PCK_XXX 1
PCLK_SDMMC0	0x0B0	R/W	0x14000000	Control Register for SD/MMC Channel 0	PCK_XXX 1
PCLK_MSTICK	0x0B4	R/W	0x0A000000	Control Register for Memory Stick Host Controller	PCK_XXX 1
PCLK_I2C	0x0B8	R/W	0x14000000	Control Register for I2C External Device Interface	PCK_XXX 1
PCLK_UART0	0x0BC	R/W	0x14000000	Control Register for UART Channel 0	PCK_XXX 1
PCLK_UART1	0x0C0	R/W	0x14000000	Control Register for UART Channel 1	PCK_XXX 1
PCLK_UART2	0x0C4	R/W	0x14000000	Control Register for UART Channel 2	PCK_XXX 1
PCLK_UART3	0x0C8	R/W	0x14000000	Control Register for UART Channel 3	PCK_XXX 1
PCLK_UART4	0x0CC	R/W	0x14000000	Control Register for UART Channel 4	PCK_XXX 1
PCLK_UART5	0x0D0	R/W	0x14000000	Control Register for UART Channel 5	PCK_XXX 1
PCLK_GPSB0	0x0D4	R/W	0x14000000	Control Register for GPSB Channel 0	PCK_XXX 1
PCLK_GPSB1	0x0D8	R/W	0x14000000	Control Register for GPSB Channel 1	PCK_XXX 1
PCLK_GPSB2	0x0DC	R/W	0x14000000	Control Register for GPSB Channel 2	PCK_XXX 1
PCLK_GPSB3	0x0E0	R/W	0x14000000	Control Register for GPSB Channel 3	PCK_XXX 1
PCLK_GPSB4	0x0E4	R/W	0x14000000	Control Register for GPSB Channel 4	PCK_XXX 1
PCLK_GPSB5	0x0E8	R/W	0x14000000	Control Register for GPSB Channel 5	PCK_XXX 1
PCLK_ADC	0x0EC	R/W	0x14000000	Control Register for ADC (Touch Screen)	PCK_YYY
PCLK_SPDIF	0x0F0	R/W	0x14000000	Control Register for SPDIF	PCK_YYY
PCLK_EHI0	0x0F4	R/W	0x14000000	Control Register for EHI Channel 0	PCK_XXX 1
PCLK_EHI1	0x0F8	R/W	0x14000000	Control Register for EHI Channel 1	PCK_XXX 1
Reserved	0x0FC	R/W	0x14000000	-	-
PCLK_CAN	0x100	R/W	0x14000000	Control Register for CAN	PCK_XXX 1
Reserved	0x104	R/W	0x14000000	-	-
PCLK_SDMMC1	0x108	R/W	0x14000000	Control Register for SD/MMC Channel 1	PCK_XXX 1
Reserved	0x10C	R/W	0x14000000	-	-
PCLK_DAI	0x110	R/W	0x14000000	Control Register for DAI (DAI Only)	PCK_YYY

1.2 Signal Descriptions

1.2.1 Global Signals

Table 1.3 Global Signals

Name	Direction	Descriptions	Related Blocks
PRESETn	Input	Reset Signal	

1.2.2 Clock Source Signals

Table 1.4 Clock Source Signals

Name	Direction	Descriptions	Related Blocks
XIN	Input	Clock Signal from PMU (12MHz Fixed)	PMU
XTIN	Input	Clock Signal from RTC	RTC
EXTCLKIN	Input	Clock Signal from External GPIO	GPIOA[13]
CLK_USB48M	Input	Clock Signal from USB OTG PHY (48MHz Fixed)	USB OTG Phy
SATA_CLKO_25M	Input	Clock Signal from SATA Oscillator (SATA_XI)	SATA Oscillator
HDMI_CLKO_27M	Input	Clock Signal from HDMI Oscillator (HDMI_XI)	HDMI Oscillator
HDMI_CLKO_PCLK	Input	Clock Signal from HDMI Phy (Video Clock)	HDMI Phy
HDMI_CLKO_TMDS	Input	Clock Signal from HDMI Phy (TMDS Clock)	HDMI Phy

1.2.3 CPU Related Signals

Table 1.5 CPU Related Signals

Name	Direction	Descriptions	Related Blocks
BCLKOUT[0] (F_{CPU})	Output	CPU Operating Clock Signal	CPU
BCLKRSTN[0]	Output	CPU Reset Signal	CPU
BCLKBUSY[0]	Input	*** It is internally generated signal.**	

1.2.4 DDI Bus Related Signals

Table 1.6 DDI Bus Related Signals

Name	Direction	Descriptions	Related Blocks
BCLKOUT[1] (F_{DBUS})	Output	DDI Bus Operating Clock Signal	DDI Bus
BCLKRSTN[1]	Output	DDI Bus Global Reset Signal	DDI Bus
BCLKBUSY[1]	Input	DDI Bus Busy Signal 0 : No traffic to the external memory from DDI Bus 1 : Traffic to the external memory from DDI Bus *** No traffic does not mean that the bus is idle.** *** It can be a case that there is internal operating without external memory access.** *** It is internally generated signal.**	DDI Bus

1.2.5 Memory Bus Related Signals

Table 1.7 Memory Bus Related Signals

Name	Direction	Descriptions	Related Blocks
BCLKOUT[2] (F_{MBUS})	Output	Memory Bus Operating Clock Signal	Memory Bus
BCLKRSTN[2]	Output	Memory Bus Global Reset Signal	Memory Bus
BCLKBUSY[2]	Input	Memory Bus Busy Signal * It has no mean and ignored internally.	

1.2.6 Graphic Bus Related Signals

Table 1.8 Graphic Bus Related Signals

Name	Direction	Descriptions	Related Blocks
BCLKOUT[3] (F_{GBUS})	Output	Graphic Bus Operating Clock Signal	Graphic Bus
BCLKRSTN[3]	Output	Graphic Bus Global Reset Signal	Graphic Bus
BCLKBUSY[3]	Input	Graphic Bus Busy Signal 0 : No traffic to the external memory from Graphic Bus 1 : Traffic to the external memory from Graphic Bus *** No traffic does not mean that the bus is idle.** *** It can be a case that there is internal operating without external memory access.** *** It is internally generated signal.**	Graphic Bus

1.2.7 I/O Bus Related Signals

Table 1.9 I/O Bus Related Signals

Name	Direction	Descriptions	Related Blocks
BCLKOUT[4] (F_{IOBUS})	Output	I/O Bus Operating Clock Signal	I/O Bus
BCLKRSTN[4]	Output	I/O Bus Global Reset Signal	I/O Bus
BCLKBUSY[4]	Input	I/O Bus Busy Signal 0 : No traffic to the external memory from I/O Bus 1 : Traffic to the external memory from I/O Bus *** No traffic does not mean that the bus is idle.** *** It can be a case that there is internal operating without external memory access.** *** It is internally generated signal.**	I/O Bus

1.2.8 Video Bus Related Signals

Table 1.10 Video Bus Related Signals

Name	Direction	Descriptions	Related Blocks
BCLKOUT[5] (F_{VBUS})	Output	Video Bus Operating Clock Signal	Video Bus
BCLKRSTN[5]	Output	Video Bus Global Reset Signal	Video Bus
BCLKBUSY[5]	Input	Video Bus Busy Signal 0 : No traffic to the external memory from Video Bus 1 : Traffic to the external memory from Video Bus *** No traffic does not mean that the bus is idle.** *** It can be a case that there is internal operating without external memory access.** *** It is internally generated signal.**	Video Bus
BCLKOUT[6] (F_{VCOD})	Output	Video Codec Operating Clock Signal	Video Bus
BCLKRSTN[6]	Output	Video Codec Global Reset Signal	Video Bus
BCLKBUSY[6]	Input	Video Codec Busy Signal 0 : No traffic to the external memory from Video Bus 1 : Traffic to the external memory from Video Bus *** No traffic does not mean that the bus is idle.** *** It can be a case that there is internal operating without external memory access.** *** It is internally generated signal.**	Video Bus

1.2.9 SMU Bus Related Signals

Table 1.11 SMU Bus Related Signals

Name	Direction	Descriptions	Related Blocks
BCLKOUT[7] (F_{SMU})	Output	SMU Bus Operating Clock Signal	SMU Bus
BCLKRSTN[7]	Output	SMU Bus Global Reset Signal	SMU Bus
BCLKBUSY[7]	Input	SMU Bus Busy Signal * It has no mean and ignored internally.	SMU Bus

1.2.10 Peripheral Clock Output Signals

Table 1.12 Video Bus Related Signals

Name	Direction	Descriptions	Related Blocks
PCLKs[0]	Output	Timer X Clock	TIMER
PCLKs[1]	Output	TimerT Clock	TIMER
PCLKs[2]	Output	TimerZ Clock	TIMER
PCLKs[3]	Output	LCD Operating Clock for Controller 0	LCD Controller 0
PCLKs[4]	Output	LCD Operating Clock for Controller 1	LCD Controller 1
PCLKs[5]	Output	LCD System Interface Clock	LCD System Interface
PCLKs[6]	Output	Camera Interface Block Output Clock	Camera Interface
PCLKs[7]	Output	Camera Interface Block Internal Operating Clock	Camera Interface
PCLKs[8]	Output	Clock Output Channel 0 to GPIO	Clock Output (GPIO)
PCLKs[9]	Output	Clock Output Channel 1 to GPIO	Clock Output (GPIO)
PCLKs[10]	Output	Operating Clock for HDMI Link Controller	HDMI
PCLKs[11]	Output	Operating Clock for USB 1.1 Host Controller	USB1.1 Host Controller
PCLKs[12]	Output	Operating Clock for SDMMC Controller 0	SDMMC Controller 0
PCLKs[13]	Output	Operating Clock for Memory Stick Controller	Memory Stick Controller
PCLKs[14]	Output	Operating Clock for I2C Controller	I2C Controller
PCLKs[15]	Output	Operating Clock for UART Controller 0	UART Controller 0
PCLKs[16]	Output	Operating Clock for UART Controller 1	UART Controller 1
PCLKs[17]	Output	Operating Clock for UART Controller 2	UART Controller 2
PCLKs[18]	Output	Operating Clock for UART Controller 3	UART Controller 3
PCLKs[19]	Output	Operating Clock for UART Controller 4	UART Controller 4
PCLKs[20]	Output	Operating Clock for UART Controller 5	UART Controller 5
PCLKs[21]	Output	Operating Clock for GPSB Controller 0	GPSB Controller 0
PCLKs[22]	Output	Operating Clock for GPSB Controller 1	GPSB Controller 1
PCLKs[23]	Output	Operating Clock for GPSB Controller 2	GPSB Controller 2
PCLKs[24]	Output	Operating Clock for GPSB Controller 3	GPSB Controller 3
PCLKs[25]	Output	Operating Clock for GPSB Controller 4	GPSB Controller 4
PCLKs[26]	Output	Operating Clock for GPSB Controller 5	GPSB Controller 5
PCLKs[27]	Output	Operating Clock for ADC (Including Touch Screen)	ADC (Touch Screen)
PCLKs[28]	Output	Operating Clock for SPDIF Controller	SPDIF
PCLKs[29]	Output	Operating Clock for EHI Controller 0	EHI Controller 0
PCLKs[30]	Output	Operating Clock for EHI Controller 1	EHI Controller 1
PCLKs[31]	Output	Reserved for Future Use	Reserved
PCLKs[32]	Output	Operating Clock for CAN Controller	CAN Controller
PCLKs[33]	Output	Reserved for Future Use	Reserved
PCLKs[34]	Output	Operating Clock for SDMMC Controller 1	SDMMC Controller 1
PCLKs[35]	Output	Reserved for Future Use	Reserved
PCLKs[36]	Output	Operating Clock for DAI Controller 1	Audio DMA

1.3 Register Descriptions

CLKCTRL0~7 Register **0xF0400000~0xF040001C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										EN	MD	CONFIG			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CONFIG												0	SEL		

Field	Name	RW	Reset	Description
2 ~ 0	SEL	RW	4	Clock source selection register for bus or CPU clocks 000b : Direct output from PLL0 001b : Direct output from PLL1 010b : Direct output from PLL2 011b : Direct output from PLL3 100b : Direct output from XIN 101b : Divided output from PLL0 110b : Divided output from PLL1 111b : Direct output from XTIN
3	SYNC	RW	0	Synchronous Mode Enable Register **** Do not change to '1'***
19 ~ 4	CONFIG	RW	1	Bus and CPU clock configuration register **** MD == '0' : Normal Mode*** CONFIG[3:0] : Divisor : 1/(DIVISOR+1) ***The value should not be 'ZERO'*** **** MD == '1' : Dynamic Control Mode*** CONFIG[3:0] : Current Divisor (Read-Only) CONFIG[7:4] : Max. Divisor CONFIG[11:8] : Min. Divisor CONFIG[15:12] : Update Cycle Period ***The max. divisor should be greater than Min. divisor and should not be zero.*** ***For the CPU clock (CLKCTRL0) in Normal Mode, the CONFIG is not divisor.*** ***The CONFIG[0] ~ [15] means clock output enable for the X'th cycle which makes 16 clock cycle repeatedly. For example, the CONFIG[15:0] = 1100101010110101b, the 1st, 2nd, 5th, 7th, 9th, 11th, 12th, 14th, 16th cycle are valid clock output.***
20	MD	RW	0	Clock Control Mode 0 for Normal Mode, 1 for Dynamic Control Mode ***The 'MD' of CLKCTRL2 (Memory Bus Clock) should be used with "Normal Mode"***
21	EN	RW	0	Clock Controller Enable Register '0' for Disable, '1' for Enable

**** The bit0 of the CONFIG value for the CLKCTRL2(Memory Bus Clock) should be '1' for clock duty. If the corresponding value is '0', the divisor is odd and the clock duty is not 50% which can be a cause of the unstable operation for the DDR memory interface.***

PLL0CFG/PLL1CFG/PLL2CFG/PLL3CFG Register **0xF0400020~0xF040002C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN					S								M		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
M								0		P					

Field	Name	RW	Reset	Description
5 ~ 0	P	RW	0x03	PLL P value (p = P) 1 ≤ P ≤ 63
18 ~ 8	M	RW	0xC0	PLL M value (m = M) 16 ≤ M ≤255
26 ~ 24	S	RW	0x01	PLL S value (s = S) 0 ≤ S ≤ 5
31	EN	RW	0x1	PLL Enable Register 0 : Disabled 1 : Enabled

The PLL has the following constraints.

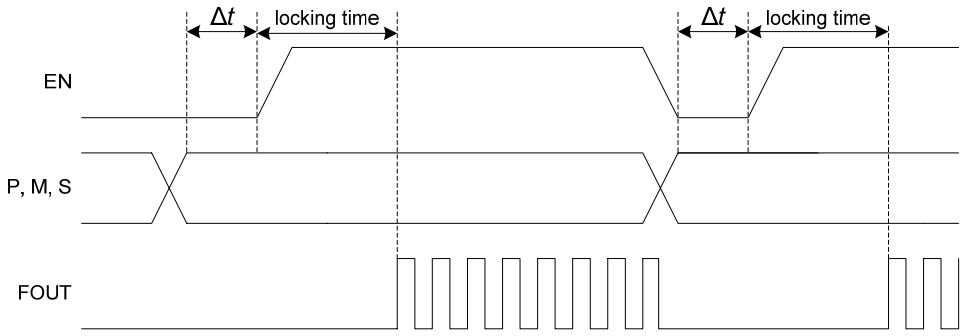
F_{VCO0} = (m * F_{IN}) / (p) ***: 800MHz ~ 1600MHz (F_{IN} is XIN oscillator)***

$F_{PLL0} = F_{VCO0} / (2 ^ s)$: F_{PLL0} should be less than 1GHz.

F_{VCO123} = (m * F_{IN}) / (p) ***: 250MHz ~ 600MHz (F_{IN} is XIN oscillator)***

$F_{PLL123} = F_{VCO123} / (2 ^ s)$: F_{PLL1} should be less than 600MHz

Whenever P, M, and S value are changed, the EN bit needs to be 0 for Δt (>5ns) and to be 1 again to restart the PLL with new setting values.



Δt > 5ns
locking time > 300us

CLKDIVC0 Register **0xF0400030**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P0E	-	P0DIV						P1E	-	P1DIV					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P2E	-	P2DIV						P3TE	-	P3DIV					

Field	Name	RW	Reset	Description
5 ~ 0	P3DIV	RW	0x3	PLL3 divisor value $F_{P3DIV} = F_{PLL3} / (P3DIV + 1)$ P3DIV should not be zero
6	-	-	-	Undefined
7	P3E	RW	0x1	PLL3 divider enable register
13 ~ 8	P2DIV	RW	0x3	PLL2 divisor value $F_{P2DIV} = F_{PLL2} / (P2DIV + 1)$ P2DIV should not be zero
14	-	-	-	Undefined
15	P2E	RW	0x1	PLL2 divider enable register
21 ~ 16	P1DIV	RW	0x2	PLL1 divisor value $F_{P1DIV} = F_{PLL1} / (P1DIV + 1)$ P1DIV should not be zero
22	-	-	-	Undefined
23	P1E	RW	0x1	PLL1 divider enable register
29 ~ 24	P0DIV	RW	0x1	PLL0 divisor value $F_{P0DIV} = F_{PLL0} / (P0DIV + 1)$ P0DIV should not be zero
30	-	-	-	Undefined
31	P0E	RW	0x1	PLL0 divider enable register

CLKDIVC1 Register **0xF0400034**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
XE	-	XDIV						XTE	-	XTDIV					

Field	Name	RW	Reset	Description
5 ~ 0	XTDIV	RW	0x0	XTIN divisor value $F_{XTDIV} = F_{XTIN} / (XTDIV + 1)$ XTDIV should not be zero
6	-	-	-	Undefined
7	XTE	RW	0x0	XTIN divider enable register
13 ~ 8	XDIV	RW	0x0	XIN divisor value $F_{XDIV} = F_{XIN} / (XDIV + 1)$ XDIV should not be zero
14	-	-	-	Undefined
15	XE	RW	0x0	XIN divider enable register
31 ~ 16	-	-	-	Undefined

SWRSTPRD Register **0xF0400040**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								PRD							

Field	Name	RW	Reset	Description
7~0	PRD	RW	0xFF	Software Reset Period ***This is used for the SWRESET register to reset the bus devices.***

SWRESET Register **0xF0400044**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								SWRST							

Field	Name	RW	Reset	Description
0	SWRST[0]	RW	0	Software Reset for CPU 0 for Not-reset, ‘1’ for Reset ***This is cleared automatically after pre-defined clock cycle which clock is SMU bus clock.***
1	SWRST[1]	RW	0	Software Reset for DDI Bus 0 for Not-reset, ‘1’ for Reset ***This is level sensitive register. This is not cleared automatically.***
2	SWRST[2]	RW	0	Software Reset for Memory Interface 0 for Not-reset, ‘1’ for Reset ***This is cleared automatically after pre-defined clock cycle which clock is SMU bus clock.***
3	SWRST[3]	RW	0	Software Reset for Graphic Bus 0 for Not-reset, ‘1’ for Reset ***This is level sensitive register. This is not cleared automatically.***
4	SWRST[4]	RW	0	Software Reset for I/O Bus 0 for Not-reset, ‘1’ for Reset ***This is level sensitive register. This is not cleared automatically.***
5	SWRST[5]	RW	0	Software Reset for Video Bus 0 for Not-reset, ‘1’ for Reset ***This is level sensitive register. This is not cleared automatically.***
6	SWRST[6]	RW	0	Software Reset for Video Core 0 for Not-reset, ‘1’ for Reset ***This is level sensitive register. This is not cleared automatically.***
7	SWRST[7]	RW	0	Software Reset for SMU Bus 0 for Not-reset, ‘1’ for Reset ***This is cleared automatically after pre-defined clock cycle which clock is SMU bus clock.***

PCK_XXX Type 1 Register **0xF0400080 + 4 * n (n=0~36, n != 3,4,5,10,27,28,31,36)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-			EN	SEL				-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-				DIV											

Field	Name	RW	Reset	Description
11 ~ 0	DIV	RW	0x0	Clock divisor for each hardware ***$F_{PCK\_xxx} = F_{CKS} / (DIV + 1)$*** F_{CKS} is selected clock by SEL In case of DIV being 0, the $F_{PCK\_xxx} = F_{CKS}$, F_{CKS} should be under 500MHz
23 ~ 12	-	-	-	Undefined
27 ~ 24	SEL	RW	0x4	Source clock selection register Refer to Clock Source Table
28	EN	RW	0x1	Clock divider enable register **** If you want to change, the value should be same level over 2 clock cycles for F_{CKS}.***
31 ~ 29	-	-	-	Undefined

Clock Source Table			
0	PLL0 direct output	1	PLL1 direct output
2	PLL2 direct output	3	PLL3 direct output
4	XIN direct	5	PLL0 divider output
6	PLL1 divider output	7	PLL2 divider output
8	PLL3 divider output	9	XTIN direct output
10	External clock[1]	11	XIN divider output
12	XTIN divider output	13	HDMI oscillator clock (27MHz)
14	SATA oscillator clock (25MHz)	15	USB PHY clock output (48MHz)

PCK_XXX Type 2 Register **0xF0400080 + 4 * n (n=0~36, n = 3,4,5,10)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-			EN	SEL				-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-				DIV											

Field	Name	RW	Reset	Description
11 ~ 0	DIV	RW	0x0	Clock divisor for each hardware ***$F_{PCK\_xxx} = F_{CKS} / (DIV + 1)$*** F_{CKS} is selected clock by SEL In case of DIV being 0, the $F_{PCK\_xxx} = F_{CKS}$, F_{CKS} should be under 500MHz
23 ~ 12	-	-	-	Undefined
27 ~ 24	SEL	RW	0x4	Source clock selection register Refer to Clock Source Table
28	EN	RW	0x1	Clock divider enable register **** If you want to change, the value should be same level over 2 clock cycles for F_{CKS}.***
31 ~ 29	-	-	-	Undefined

Clock Source Table			
0	PLL0 direct output	1	PLL1 direct output
2	PLL2 direct output	3	PLL3 direct output
4	XIN direct	5	PLL0 divider output
6	PLL1 divider output	7	PLL2 divider output
8	PLL3 divider output	9	XTIN direct output
10	External clock[2]	11	HDMI TMDS clock (Variable)
12	HDMI PCLK (Variable)	13	HDMI oscillator clock (27MHz)
14	SATA oscillator clock (25MHz)	15	USB PHY clock output (48MHz)

[1] GPIOA[13] can be configured as the external clock function.
[2] GPIOA[13] can be configured as the external clock function.

PCK_YYY Register **0xF0400080 + 4 * n, n = 27,28,31,36**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MD	-		EN	SEL				-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIV															

Field	Name	RW	Reset	Description
15 ~ 0	DIV	RW	0x0	Clock divisor for each hardware In Divider Mode, ***$F_{PCK\_YYY} = F_{CKS} / (DIV + 1)$*** In DCO Mode, ***DIV > 32768*** ***$F_{PCK\_DAI} = F_{CKS} * ((65536-DIV) / (65536))$*** ***DIV <= 32768*** ***$F_{PCK\_DAI} = F_{CKS} * (DIV / (65536))$*** FCKS is selected clock by SEL DIV should not be “ZERO”
23 ~ 16	-	-	-	Undefined
27 ~ 24	SEL	RW	0xA	Source clock selection register Refer to Clock Source Table
28	EN	RW	0x1	Clock divider enable register **** If you want to change, the value should be same level over 2 clock cycles for F_{CKS}.***
30 ~ 29	-	-	-	Undefined
31	MD	RW	0x0	Mode selection register 1 : DIVIDER mode 0 : DCO mode

* If you want the peripheral clock generator to be disabled, you should select the lowest frequency clock source for reducing the power consumption.

Clock Source Table			
0	PLL0 direct output	1	PLL1 direct output
2	PLL2 direct output	3	PLL3 direct output
4	XIN direct	5	PLL0 divider output
6	PLL1 divider output	7	PLL2 divider output
8	PLL3 divider output	9	XTIN direct output
10	External clock[3]	11	XIN divider output
12	XTIN divider output	13	HDMI oscillator clock (27MHz)
14	SATA oscillator clock (25MHz)	15	USB PHY clock output (48MHz)

[3] GPIOA[13] can be configured as the external clock function.

1.4 Operation & Timing Diagram

1.4.1 Clock Change Operation (Safe Clock Changer)

An example of the timing diagram for changing the clock is shown below, Figure 1.2.

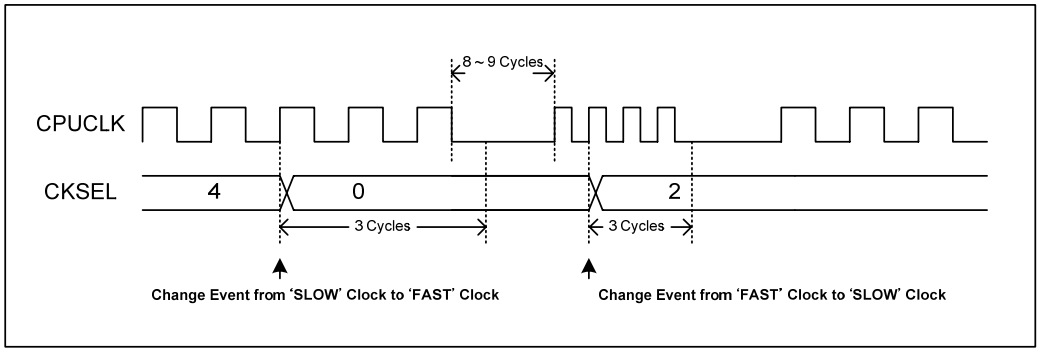


Figure 1.2 Clock Change Timing Diagram

An example shows two changing sequences. The first sequence is for changing from slower to faster frequency.

When the CPU write the 'SEL' register bits in the CLKCTRL0 ~ CLKCTRL7 with specified value ('0' in this example), the glitch-free circuit in the CKC hardware first stops the current clock after 3 clock cycles. And then in the 3 clock period, the clock multiplexor changes from the current clock to the next clock. Finally, the wake-up circuits enables the clock output. The changing sequence from 'faster' to 'slower' clocks is same procedure.

1.4.2 How to Generate "F_{CPU} " through BCLKOUT[0]

The following figure shows the clock chain from source to target which is BCLKOUT[0].
The clock selected by safe clock changer is "XIN" at the reset state.
Before changing the clock, the source and target clock should be alived. For example, assuming that the current clock is PLL1 and target clock is PLL0, the PLL1 should not be disabled and the PLL0 should be enabled before changing.

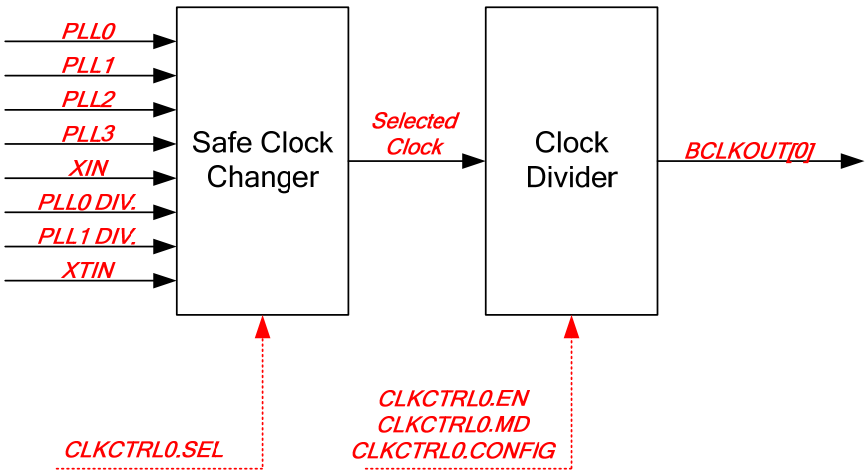


Figure 1.3 Clock Chain for BCLKOUT[0]

After selection of the source clock, you can change the divider configuration "CLKCTRL0.CONFIG" to make fast or slow for the target clock.

The following figure shows the waveform of the BCLKOUT[0] configured by CLKCTRL0.CONFIG.

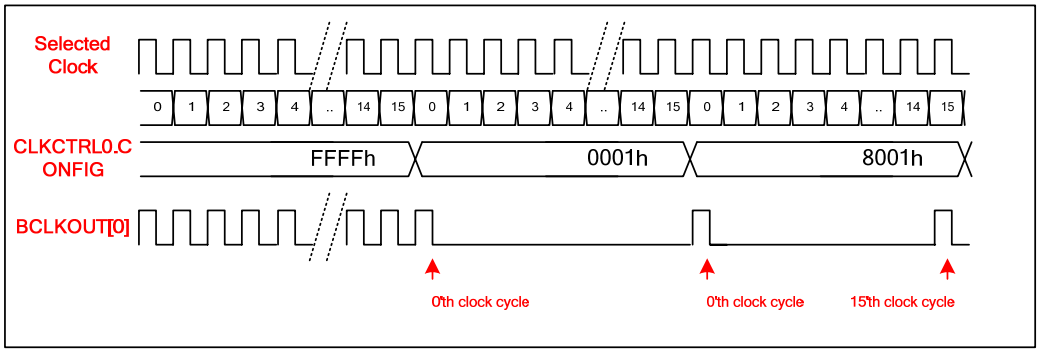


Figure 1.4 Clock Change Timing Diagram for BCLKOUT[0]

If the CLKCTRL0.CONFIG value is “FFFFh”, the output clock frequency is same as the input clock. But, in case of “0001h”, the valid clock cycle is only one which is first clock cycle and makes the frequency is 16 times slower than source clock. In the case of “8001h”, the output clock frequency is 8 times slower than source clock and the valid clocks are first and last clock cycle.

1.4.3 How to Generate BCLKOUT[1] ~ BCLKOUT[7]

The following figure shows the clock chain from source to target which is BCLKOUT[1] ~ BCLKOUT[7].
The clock selected by safe clock changer is “XIN” at the reset state.
Before changing the clock, the source and target clock should be alived. For example, assuming that the current clock is PLL1 and target clock is PLL0, the PLL1 should not be disabled and the PLL0 should be enabled before changing.

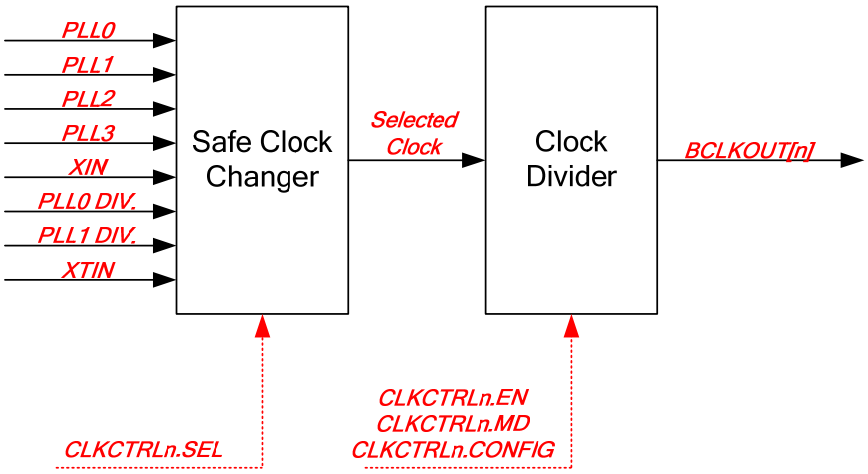


Figure 1.5 Clock Chain for BCLKOUT[0]

After selection of the source clock, you can change the divider configuration “CLKCTRLn.CONFIG” to make fast or slow for the target clock.

The following figure shows the waveform of the BCLKOUT[n] configured by CLKCTRLn.CONFIG.

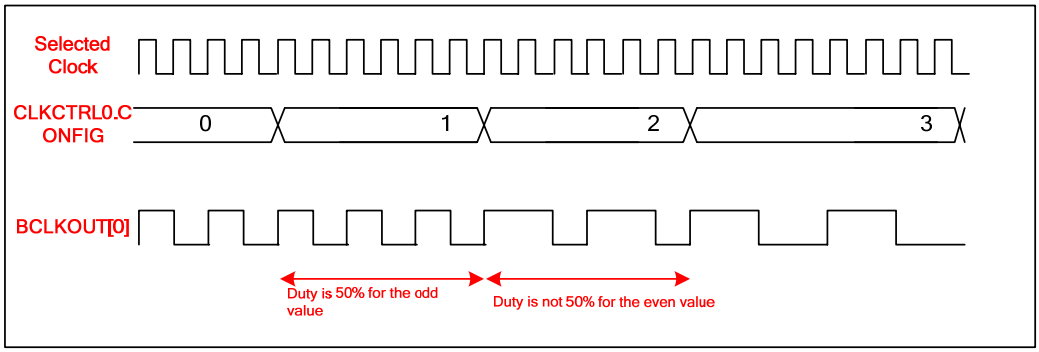


Figure 1.6 Clock Change Timing Diagram for BCLKOUT[7:1]

The divisor is the “config value + 1”. And for the odd value(even divisor), the duty of the output clock is 50%. But, the even

value(odd divisor) does not guarantee the 50% duty. The larger divisor, the closer to 50% duty. The duty of the BCLKOUT[2] for memory interface should be 50%, so the value for the CLKCTRL2.CONFIG should be odd which means even divisor such as 1/2, 1/4, 1/8 and etc.

1.4.4 How to Generate PCLKs[36:0]

The following figure shows the clock chain from source to target which is PCLKs[0] ~ PCLKs[36].
The input clock sources are 16 which are different from PCK_XXX type1, PCK_XXX type2 and PCK_YYY.
The clock selected by safe clock changer is “XIN” at the reset state.

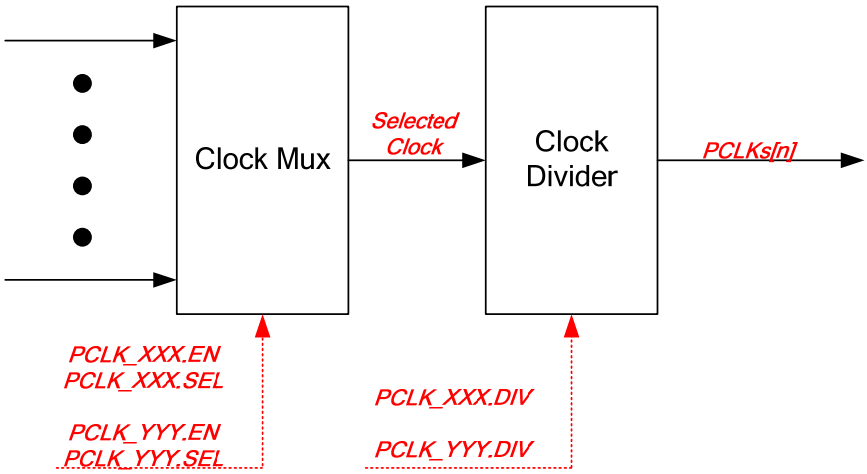


Figure 1.7 Clock Chain for PCLKs[36:0]

After selection of the source clock, you can change the divider configuration with “PCLK_XXX.DIV” or “PCLK_YYY.DIV” to make fast or slow for the target clock.

The following figure shows the waveform of the PCLKs[n] configured.

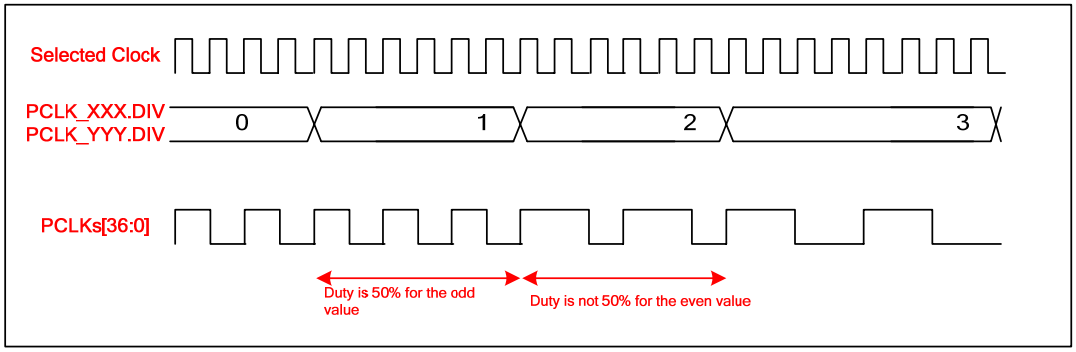


Figure 1.8 Clock Change Timing Diagram for BCLKOUT[7:1]

The divisor is the “config value + 1”. And for the odd value(even divisor), the duty of the output clock is 50%. But, the even value(odd divisor) does not guarantee the 50% duty. The larger divisor, the closer to 50% duty.

1.4.5 Procedure for Configuration of Peripheral Clocks

The Figure 1.9 shows the configuration procedure for peripheral clock.

The programmer should take care of some cautions in configuring the peripheral clock for corresponding hardware. (ex, MS, DAI, etc.)

The controller should be in reset state by setting the SWRESET bit before configuring the hardware clock and configuring the corresponding ports. If you configure the clock and ports in not-reset state, which is not-initialized state, the unexpected operations can occur.

The hardware starts operating by clearing the SWRESET bit after the configuration for the clock and ports of corresponding controller.

If you want to close the hardware operation, you should make the controller into reset-state by SWRESET bit and stop the clock and release the port.

If the controller is in reset-state (SWRESET='1'), the program can't access the control register for corresponding hardware.

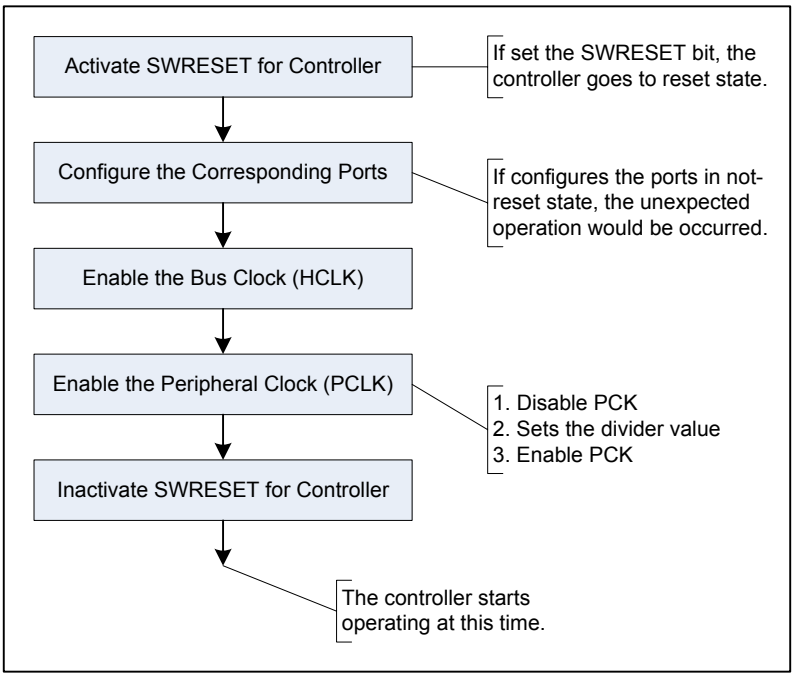


Figure 1.9 Peripheral Clock Configuration Procedure

1.4.6 Reference PMS Table for Target Frequency

PLLs embedded in the TCC8900 have the following constraints.

FVCO = M * (FIN / P), FPLL = FVCO / (2 ^ S)

Where FIN is the input(XIN) frequency of oscillator, FPLL is final output of PLL.

About FVCO, it is restriced to 800MHz ~ 1600MHz in PLL1/2/3 and 250MHz ~ 600MHz in PLL1/2/3. As a result, FPLL is restriced to 40MHz ~ 1600MHz in PLL1/2/3 and 8MHz ~ 600MHz in PLL1/2/3.

You must choose the proper divider values from the following table. If you want to use other divider value than that in following, consult with Telechips. To set the selected divider value, refer to PLLnCFG.

Table 1.13 PLL1/2/3 Configuration Examples[4]

FIN (MHz)	P(1~63)	M	S(0~5)	FVCO(MHz)	FPLL(MHz)
12	3	128	5	512	16
12	3	99	4	396	24.75
12	3	140	3	560	70
12	3	144	3	576	72
12	3	86	2	344	86
12	3	101	2	404	101
12	3	132	2	528	132
12	3	73	1	292	146
12	3	90	1	360	180
12	3	92	1	368	184
12	3	95	1	380	190
12	3	96	1	384	192
12	3	102	1	408	204
12	3	104	1	416	208
12	3	105	1	420	210
12	3	107	1	428	214
12	3	108	1	432	216
12	3	111	1	444	222
12	3	113	1	452	226
12	3	114	1	456	228
12	3	115	1	460	230
12	3	117	1	468	234
12	3	118	1	472	236
12	3	122	1	488	244
12	3	123	1	492	246
12	3	124	1	496	248
12	3	126	1	504	252
12	3	127	1	508	254
12	3	129	1	516	258
12	3	130	1	520	260
12	3	131	1	524	262
12	1	44	1	528	264
12	3	133	1	532	266
12	3	138	1	552	276
12	3	141	1	564	282
12	3	145	1	580	290

[4] The ranges of M and FIN/P in PLL0 are different from those in others.
PLL0 : M = 64~1023, FIN/P = 2~6
PLL1/2/3: M = 15~255, FIN/P = no restriction

FIN (MHz)	P(1~63)	M	S(0~5)	FVCO(MHz)	FPLL(MHz)
12	3	147	1	588	294
12	3	76	0	304	304
12	3	77	0	308	308
12	2	52	0	312	312
12	3	79	0	316	316
12	3	80	0	320	320
12	3	81	0	324	324
12	3	82	0	328	328
12	2	55	0	330	330
12	3	83	0	332	332
12	3	84	0	336	336
12	3	85	0	340	340
12	3	87	0	348	348
12	3	88	0	352	352
12	3	89	0	356	356
12	1	30	0	360	360
12	6	181	0	362	362
12	3	91	0	364	364
12	4	123	0	369	369
12	6	185	0	370	370
12	1	31	0	372	372
12	3	95	0	380	380
12	4	127	0	381	381
12	1	32	0	384	384
12	3	97	0	388	388
12	2	65	0	390	390
12	3	98	0	392	392
12	4	131	0	393	393
12	1	33	0	396	396
12	4	133	0	399	399
12	3	100	0	400	400
12	3	101	0	404	404
12	5	169	0	405.6	405.6
12	6	203	0	406	406
12	1	34	0	408	408
12	6	205	0	410	410
12	3	103	0	412	412
12	3	104	0	416	416
12	5	174	0	417.6	417.6
12	1	35	0	420	420
12	3	106	0	424	424
12	3	107	0	428	428
12	4	143	0	429	429
12	6	215	0	430	430
12	1	36	0	432	432
12	3	109	0	436	436
12	3	110	0	440	440
12	6	221	0	442	442
12	1	37	0	444	444
12	3	112	0	448	448
12	2	75	0	450	450
12	3	113	0	452	452

FIN (MHz)	P(1~63)	M	S(0~5)	FVCO(MHz)	FPLL(MHz)
12	6	227	0	454	454
12	1	38	0	456	456
12	3	115	0	460	460
12	5	193	0	463.2	463.2
12	3	116	0	464	464
12	6	233	0	466	466
12	1	39	0	468	468
12	6	235	0	470	470
12	3	118	0	472	472
12	2	79	0	474	474
12	3	119	0	476	476
12	1	40	0	480	480
12	3	121	0	484	484
12	2	81	0	486	486
12	3	122	0	488	488
12	6	245	0	490	490
12	1	41	0	492	492
12	3	124	0	496	496
12	5	207	0	496.8	496.8
12	2	83	0	498	498
12	3	125	0	500	500
12	1	42	0	504	504
12	3	127	0	508	508
12	3	128	0	512	512
12	1	43	0	516	516
12	3	130	0	520	520
12	3	131	0	524	524
12	1	44	0	528	528
12	3	133	0	532	532
12	3	134	0	536	536
12	1	45	0	540	540
12	3	136	0	544	544
12	3	137	0	548	548
12	1	46	0	552	552
12	3	139	0	556	556
12	3	140	0	560	560
12	1	47	0	564	564
12	3	142	0	568	568
12	3	143	0	572	572
12	1	48	0	576	576
12	3	145	0	580	580
12	3	146	0	584	584
12	1	49	0	588	588
12	3	148	0	592	592
12	3	149	0	596	596
12	1	50	0	600	600

Table 1.14 PLL0 Configuration Examples[5]

FIN (MHz)	P(1~63)	M	S(0~5)	FVCO(MHz)	FPLL(MHz)
12	3	202	3	808	101
12	2	176	3	1056	132
12	3	280	3	1120	140
12	2	192	3	1152	144
12	3	292	3	1168	146
12	3	344	3	1376	172
12	2	240	3	1440	180
12	3	368	3	1472	184
12	3	380	3	1520	190
12	2	256	3	1536	192
12	3	202	2	808	202
12	2	136	2	816	204
12	3	208	2	832	208
12	2	140	2	840	210
12	2	140	2	840	210
12	3	214	2	856	214
12	2	144	2	864	216
12	2	148	2	888	222
12	3	226	2	904	226
12	2	152	2	912	228
12	3	230	2	920	230
12	2	156	2	936	234
12	3	236	2	944	236
12	3	244	2	976	244
12	2	164	2	984	246
12	3	248	2	992	248
12	2	168	2	1008	252
12	3	254	2	1016	254
12	2	172	2	1032	258
12	3	260	2	1040	260
12	3	262	2	1048	262
12	2	176	2	1056	264
12	3	266	2	1064	266
12	2	184	2	1104	276
12	2	188	2	1128	282
12	3	290	2	1160	290
12	3	292	2	1168	292
12	2	196	2	1176	294
12	3	304	2	1216	304
12	3	308	2	1232	308
12	2	208	2	1248	312
12	3	316	2	1264	316
12	3	320	2	1280	320
12	2	216	2	1296	324
12	3	328	2	1312	328
12	2	220	2	1320	330

[5] The ranges of M and FIN/P in PLL0 are different from those in others.
PLL0 : M = 64~1023, FIN/P = 2~6
PLL1/2/3: M = 15~255, FIN/P = no restriction

FIN (MHz)	P(1~63)	M	S(0~5)	FVCO(MHz)	FPLL(MHz)
12	3	332	2	1328	332
12	2	224	2	1344	336
12	3	340	2	1360	340
12	2	232	2	1392	348
12	3	352	2	1408	352
12	3	356	2	1424	356
12	2	240	2	1440	360
12	3	362	2	1448	362
12	3	364	2	1456	364
12	3	368	2	1472	368
12	2	246	2	1476	369
12	3	370	2	1480	370
12	2	248	2	1488	372
12	3	380	2	1520	380
12	3	380	2	1520	380
12	2	254	2	1524	381
12	2	256	2	1536	384
12	3	388	2	1552	388
12	2	260	2	1560	390
12	3	392	2	1568	392
12	2	262	2	1572	393
12	2	264	2	1584	396
12	2	266	2	1596	399
12	3	400	2	1600	400
12	3	202	1	808	404
12	5	338	1	811	406
12	3	203	1	812	406
12	2	136	1	816	408
12	3	205	1	820	410
12	3	206	1	824	412
12	3	208	1	832	416
12	5	348	1	835	418
12	2	140	1	840	420
12	3	212	1	848	424
12	3	214	1	856	428
12	2	143	1	858	429
12	3	215	1	860	430
12	2	144	1	864	432
12	3	218	1	872	436
12	3	220	1	880	440
12	3	221	1	884	442
12	2	148	1	888	444
12	3	224	1	896	448
12	2	150	1	900	450
12	3	226	1	904	452
12	3	227	1	908	454
12	2	152	1	912	456
12	3	230	1	920	460
12	5	386	1	926	463
12	3	232	1	928	464
12	3	233	1	932	466
12	2	156	1	936	468

FIN (MHz)	P(1~63)	M	S(0~5)	FVCO(MHz)	FPLL(MHz)
12	3	235	1	940	470
12	3	236	1	944	472
12	2	158	1	948	474
12	3	238	1	952	476
12	2	160	1	960	480
12	3	242	1	968	484
12	2	162	1	972	486
12	3	244	1	976	488
12	3	245	1	980	490
12	2	164	1	984	492
12	3	248	1	992	496
12	5	414	1	994	497
12	2	166	1	996	498
12	3	250	1	1000	500
12	2	168	1	1008	504
12	3	254	1	1016	508
12	3	256	1	1024	512
12	2	172	1	1032	516
12	3	260	1	1040	520
12	3	262	1	1048	524
12	2	176	1	1056	528
12	3	266	1	1064	532
12	3	268	1	1072	536
12	2	180	1	1080	540
12	3	272	1	1088	544
12	3	274	1	1096	548
12	2	184	1	1104	552
12	3	278	1	1112	556
12	3	280	1	1120	560
12	2	188	1	1128	564
12	3	284	1	1136	568
12	3	286	1	1144	572
12	2	192	1	1152	576
12	3	290	1	1160	580
12	3	292	1	1168	584
12	2	196	1	1176	588
12	3	296	1	1184	592
12	3	298	1	1192	596
12	2	200	1	1200	600
12	3	304	1	1216	608
12	3	308	1	1232	616
12	2	208	1	1248	624
12	3	316	1	1264	632
12	3	320	1	1280	640
12	2	216	1	1296	648
12	3	328	1	1312	656
12	2	220	1	1320	660
12	3	332	1	1328	664
12	2	224	1	1344	672
12	3	340	1	1360	680
12	2	232	1	1392	696
12	3	280	4	1120	70
12	3	352	1	1408	704
12	3	356	1	1424	712
12	2	192	4	1152	72
12	2	240	1	1440	720

FIN (MHz)	P(1~63)	M	S(0~5)	FVCO(MHz)	FPLL(MHz)
12	3	362	1	1448	724
12	3	364	1	1456	728
12	2	246	1	1476	738
12	3	370	1	1480	740
12	2	248	1	1488	744
12	3	380	1	1520	760
12	2	254	1	1524	762
12	2	256	1	1536	768
12	3	388	1	1552	776
12	2	260	1	1560	780
12	3	392	1	1568	784
12	2	262	1	1572	786
12	2	264	1	1584	792
12	2	266	1	1596	798
12	3	400	1	1600	800
12	3	202	0	808	808
12	5	338	0	811	811
12	3	203	0	812	812
12	2	136	0	816	816
12	3	205	0	820	820
12	3	206	0	824	824
12	3	208	0	832	832
12	5	348	0	835	835
12	2	140	0	840	840
12	3	212	0	848	848
12	3	214	0	856	856
12	2	143	0	858	858
12	3	344	4	1376	86
12	3	215	0	860	860
12	2	144	0	864	864
12	3	218	0	872	872
12	3	220	0	880	880
12	3	221	0	884	884
12	2	148	0	888	888
12	3	224	0	896	896
12	2	150	0	900	900
12	3	226	0	904	904
12	3	227	0	908	908
12	2	152	0	912	912
12	3	230	0	920	920
12	5	386	0	926	926
12	3	232	0	928	928
12	3	233	0	932	932
12	2	156	0	936	936
12	3	235	0	940	940
12	3	236	0	944	944
12	2	158	0	948	948
12	3	238	0	952	952
12	2	160	0	960	960
12	3	242	0	968	968
12	2	162	0	972	972
12	3	244	0	976	976
12	3	245	0	980	980
12	2	164	0	984	984
12	3	248	0	992	992
12	5	414	0	994	994
12	2	166	0	996	996
12	3	250	0	1000	1000
12	2	168	0	1008	1008
12	3	254	0	1016	1016
12	3	256	0	1024	1024
12	2	172	0	1032	1032
12	3	260	0	1040	1040
12	3	262	0	1048	1048

FIN (MHz)	P(1~63)	M	S(0~5)	FVCO(MHz)	FPLL(MHz)
12	2	176	0	1056	1056
12	3	266	0	1064	1064
12	3	268	0	1072	1072
12	2	180	0	1080	1080
12	3	272	0	1088	1088
12	3	274	0	1096	1096
12	2	184	0	1104	1104
12	3	278	0	1112	1112
12	3	280	0	1120	1120
12	2	188	0	1128	1128
12	3	284	0	1136	1136
12	3	286	0	1144	1144
12	2	192	0	1152	1152
12	3	290	0	1160	1160
12	3	292	0	1168	1168
12	2	196	0	1176	1176
12	3	296	0	1184	1184
12	3	298	0	1192	1192
12	2	200	0	1200	1200

2 VPIC(Vectored Priority Interrupt Controller)

2.1 Overview

The following figure represents the block diagram of interrupt controller. The interrupt controller can manage up to 64 interrupt sources. In the TCC8900, there are 12 external interrupt sources that can be detected various kind of method that is a rising edge/ falling edge / level high / level low. The external interrupt sources can be selected among the various GPIO ports and fed reliably into interrupt controller with dedicated noise filters.

There are two types of interrupt in ARM family processor; IRQ type, FIQ type. Interrupt controller can select these two types for each interrupt sources separately.

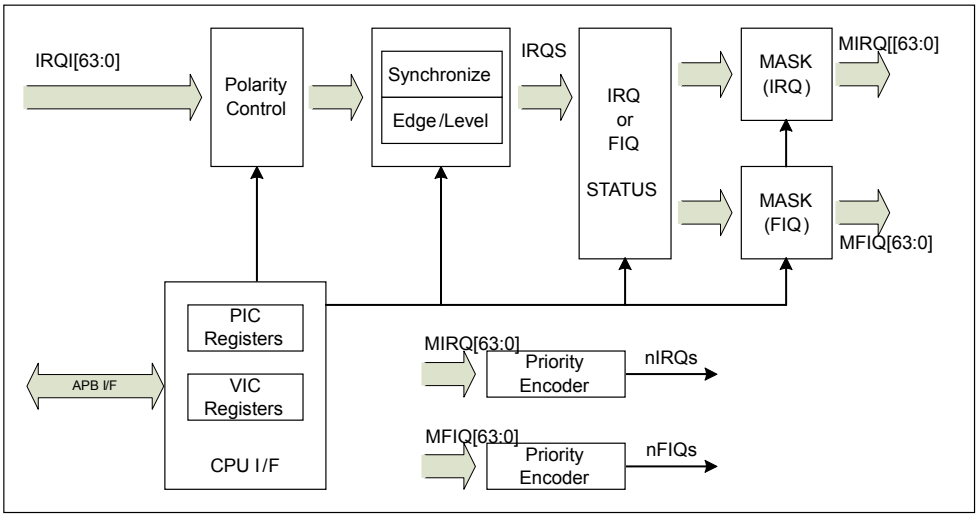


Figure 2.1 Block Diagram of Vectored Interrupt Controller

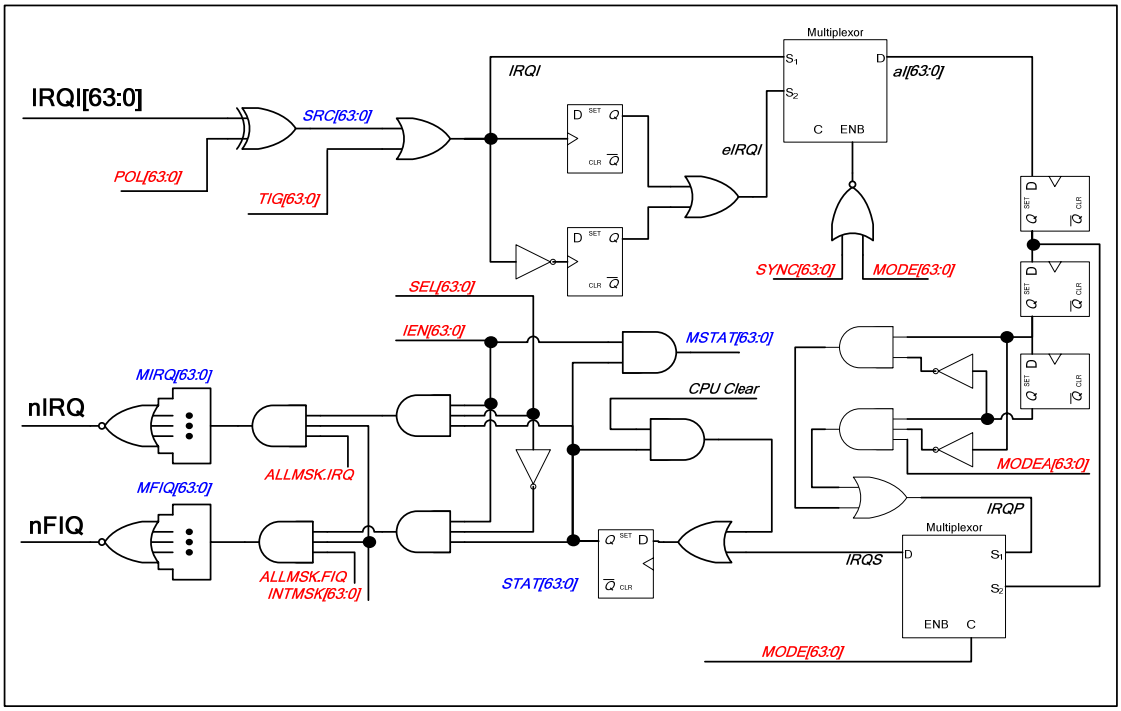


Figure 2.2 Detailed Interrupt Flow

The above figure shows the detailed interrupt flow from IRQI[63:0] to nIRQ or nFIQ. The red colored signals are configuration register fields and the blue colored ones are status registers describe in the register description. **The MIRQ and MFIQ are used to encode the vector number and vector address with pre-defined priorities in the vectored interrupt controller**.

2.2 Signal Descriptions

2.2.1 Global Signals

Table 2.1 Global Signals

Name	Direction	Descriptions	Related Blocks
PCLK	Input	SMU Bus Clock	CKC
PRESETn	Input	Reset Signal	CKC

2.2.2 Interrupt Source Signals

Table 2.2 Clock Source Signals

Name	Direction	Descriptions	Related Blocks
IRQI[63:0]	Input	Interrupt Sources from on-chip devices or external interrupts.	Each Block or GPIOs

2.2.3 Interrupt Output Signals (to ARM)

Table 2.3 Interrupt Output Signals (to ARM)

Name	Direction	Descriptions	Related Blocks
nFIQ	Output	This is FIQ signal to ARM processor and is active low and level sensitive. This goes to LOW just after interrupt occurred and goes to HIGH just after clearing the interrupt status.	ARM
nIRQ	Output	This is IRQ signal to ARM processor and is active low and level sensitive. This goes to LOW just after interrupt occurred and goes to HIGH just after clearing the interrupt status.	ARM

2.2.4 Timer Related Signals

Table 2.4 Interrupt Output Signals (to ARM)

Name	Direction	Descriptions	Related Blocks
IRQP[0]	Output	This is the interrupt pulse signal to counting in the timer. The corresponding source is IRQI[0]. The pulse can be generated at the rising edge or falling edge or both edges.	TIMER
IRQP[1]	Output	This is the interrupt pulse signal to counting in the timer. The corresponding source is IRQI[1]. The pulse can be generated at the rising edge or falling edge or both edges.	TIMER
IRQP[2]	Output	This is the interrupt pulse signal to counting in the timer. The corresponding source is IRQI[2]. The pulse can be generated at the rising edge or falling edge or both edges.	TIMER
IRQP[3]	Output	This is the interrupt pulse signal to counting in the timer. The corresponding source is IRQI[3]. The pulse can be generated at the rising edge or falling edge or both edges.	TIMER

2.3 Register Descriptions

Table 2.5 Priority Interrupt Controller Register Map (Base Address = 0xF0401000)

Name	Address	Type	Reset	Description
IEN0	0x000	R/W	0x00000000	Interrupt Enable0 Register
IEN1	0x004	R/W	0x00000000	Interrupt Enable1 Register
CLR0	0x008	R/W	0x00000000	Interrupt Clear0 Register
CLR1	0x00C	R/W	0x00000000	Interrupt Clear1 Register
STS0	0x010	R	Unknown	Interrupt Status0 Register
STS1	0x014	R	Unknown	Interrupt Status1 Register
SEL0	0x018	R/W	0x00000000	IRQ or FIQ Selection0 Register
SEL1	0x01C	R/W	0x00000000	IRQ or FIQ Selection1 Register
SRC0	0x020	R	Unknown	Source Interrupt Status0 Register
SRC1	0x024	R	Unknown	Source Interrupt Status1 Register
MSTS0	0x028	R	0x00000000	Masked Status0 Register
MSTS1	0x02C	R	0x00000000	Masked Status1 Register
TIG0	0x030	R/W	0x00000000	Test Interrupt Generation0 Register
TIG1	0x034	R/W	0x00000000	Test Interrupt Generation1 Register
POL0	0x038	R/W	0x00000000	Interrupt Polarity0 Register
POL1	0x03C	R/W	0x00000000	Interrupt Polarity1 Register
IRQ0	0x040	R	0x00000000	IRQ Raw Status0 Register
IRQ1	0x044	R	0x00000000	IRQ Raw Status1 Register
FIQ0	0x048	R	Unknown	FIQ Status0 Register
FIQ1	0x04C	R	Unknown	FIQ Status1 Register
MIRQ0	0x050	R	0x00000000	Masked IRQ Status0 Register
MIRQ1	0x054	R	0x00000000	Masked IRQ Status1 Register
MFIQ0	0x058	R	0x00000000	Masked FIQ Status0 Register
MFIQ1	0x05C	R	0x00000000	Masked FIQ Status1 Register
MODE0	0x060	R/W	0x00000000	Trigger Mode0 Register – Level or Edge
MODE1	0x064	R/W	0x00000000	Trigger Mode1 Register – Level or Edge
SYNC0	0x068	R/W	0xFFFFFFFF	Synchronization Enable0 Register
SYNC1	0x06C	R/W	0xFFFFFFFF	Synchronization Enable1 Register
WKEN0	0x070	R/W	0x00000000	Wakeup Event Enable0 Register
WKEN1	0x074	R/W	0x00000000	Wakeup Event Enable1 Register
MODEA0	0x078	R/W	0x00000000	Both Edge or Single Edge0 Register
MODEA1	0x07C	R/W	0x00000000	Both Edge or Single Edge1 Register
INTMSK0	0x100	R/W	0xFFFFFFFF	Interrupt Output Masking0 Register
INTMSK1	0x104	R/W	0xFFFFFFFF	Interrupt Output Masking1 Register
ALLMSK	0x108	R/W	0x00000003	All Mask Register

Table 2.6 Vectored Interrupt Controller Register Map (Base Address = 0xF0401200)

Name	Address	Type	Reset	Description
VAIRQ	0x200	R	0x800000XX	IRQ Vector Register
VAFIQ	0x204	R	0x800000XX	FIQ Vector Register
VNIRQ	0x208	R	0x800000XX	IRQ Vector Number Register
VNFIQ	0x20C	R	0x800000XX	FIQ Vector Number Register
VCTRL	0x210	R/W	0x00000000	Vector Control Register
PRIO0	0x220	R/W	0x03020100	Priorities for Interrupt 0 ~ 3
PRIO1	0x224	R/W	0x07060504	Priorities for Interrupt 4 ~ 7
PRIO2	0x228	R/W	0x0B0A0908	Priorities for Interrupt 8 ~ 11
PRIO3	0x22C	R/W	0x0F0E0D0C	Priorities for Interrupt 12 ~ 15
PRIO4	0x230	R/W	0x13121110	Priorities for Interrupt 16 ~ 19
PRIO5	0x234	R/W	0x17161514	Priorities for Interrupt 20 ~ 23
PRIO6	0x238	R/W	0x1B1A1918	Priorities for Interrupt 24 ~ 27
PRIO7	0x23C	R/W	0x1F1E1D1C	Priorities for Interrupt 28 ~ 31
PRIO8	0x240	R/W	0x23222120	Priorities for Interrupt 32 ~ 35
PRIO9	0x244	R/W	0x27262524	Priorities for Interrupt 36 ~ 39
PRIO10	0x248	R/W	0x2B2A2928	Priorities for Interrupt 40 ~ 43
PRIO11	0x24C	R/W	0x2F2E2D2C	Priorities for Interrupt 44 ~ 47
PRIO12	0x250	R/W	0x33323130	Priorities for Interrupt 48 ~ 51
PRIO13	0x254	R/W	0x37363534	Priorities for Interrupt 52 ~ 55
PRIO14	0x258	R/W	0x3B3A3938	Priorities for Interrupt 56 ~ 59
PRIO15	0x25C	R/W	0x3F3E3D3C	Priorities for Interrupt 60 ~ 63

2.3.1 Priority Interrupt Controller

Interrupt Enable Register (IEN0) **0xF0401000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Interrupt Enable Register (IEN1) **0xF0401004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: For each bit, ‘1’ means that corresponding interrupt is enabled and ‘0’ for disabled

Field	Name	RW	Reset	Description
0	TC0	RW	0x0	Timer 0 interrupt enable
1	TC1	RW	0x0	Timer 1 interrupt enable
2	SMUI2C	RW	0x0	SMU_I2C interrupt enable
3	EI0	RW	0x0	External interrupt 0 enable
4	EI1	RW	0x0	External interrupt 1 enable
5	EI2	RW	0x0	External interrupt 2 enable
6	EI3	RW	0x0	External interrupt 3 enable
7	EI4	RW	0x0	External interrupt 4 enable
8	EI5	RW	0x0	External interrupt 5 enable
9	EI6	RW	0x0	External interrupt 6 enable
10	EI7	RW	0x0	External interrupt 7 enable
11	EI8	RW	0x0	External interrupt 8 enable
12	EI9	RW	0x0	External interrupt 9 enable
13	EI10	RW	0x0	External interrupt 10 enable
14	EI11	RW	0x0	External interrupt 11 enable
15	SC0	RW	0x0	Mem-to-Mem scaler 0 interrupt enable
16	SC1	RW	0x0	Mem-to-Mem scaler 1 interrupt enable
17	CAM	RW	0x0	Camera interrupt enable
18	LCD0	RW	0x0	LCD controller 0 interrupt enable
19	LCD1	RW	0x0	LCD controller 1 interrupt enable
20	VIPET	RW	0x0	VIPET controller interrupt enable Note: the interrupt request signal is active low.
21	JPGD	RW	0x0	JPEG Decoder interrupt enable
22	JPGE	RW	0x0	JPEG Encoder interrupt enable
23	VCDC	RW	0x0	Video CODEC interrupt enable
24	3DPP	RW	0x0	3D Pixel Processor interrupt enable
25	3DGP	RW	0x0	3D Geometry Processor interrupt enable
26	3DMMU	RW	0x0	3D MMU interrupt enable
27	G2D	RW	0x0	Graphic Engine 2D Hardware Interrupt Enable
28	TSADC	RW	0x0	TSADC interrupt enable
29	DMA	RW	0x0	DMA controller interrupt enable
30	ECC	RW	0x0	ECC interrupt enable
31	EHI0	RW	0x0	External interrupt 0 enable
32	EHI1	RW	0x0	External interrupt 1 enable
33	CAN	RW	0x0	CAN interrupt enable
34	HDMI	RW	0x0	HDMI interrupt enable
35	SATA	RW	0x0	SATA Host interrupt enable
36	GPSB	RW	0x0	GPSB Interrupt Enable
37	HDD	RW	0x0	HDD controller interrupt enable

38	I2C	RW	0x0	I2C interrupt enable
39	MPEFEC	RW	0x0	MPEFEC interrupt enable
40	MS	RW	0x0	Memory Stick interrupt enable
41	NFC	RW	0x0	Nand flash controller interrupt enable
42	RMT	RW	0x0	Remote Control interrupt enable
43	RTC	RW	0x0	RTC interrupt enable
44	SD0	RW	0x0	SD/MMC 0 interrupt enable
45	SD1	RW	0x0	SD/MMC 1 interrupt enable
46	SPDTX	RW	0x0	SPDIF transmitter interrupt enable
47	UART	RW	0x0	UART interrupt enable
48	UOTG	RW	0x0	USB 2.0 OTG interrupt enable
49	U11H	RW	0x0	USB 1.1 host interrupt enable
50	-	RW	0x0	Reserved
51	-	RW	0x0	Reserved
52	-	RW	0x0	Reserved
53	TSIF0	RW	0x0	TS interface 0 interrupt enable
54	TSIF1	RW	0x0	TS interface 1 interrupt enable
55	CDRX	RW	0x0	CDIF receive interrupt enable
56	DAIRX	RW	0x0	DAI receive interrupt enable
57	DAITX	RW	0x0	DAI transmit interrupt enable
58	ADMA	RW	0x0	AUDIO DMA interrupt enable
59	AUDIO	RW	0x0	AUDIO interrupt enable
60	APMU	RW	0x0	ARM System Metrics interrupt enable Note: the interrupt request signal is active low.
61	AIRQ	RW	0x0	Non secure ARM DMA interrupt enable Note: the interrupt request signal is active low.
62	ASIRQ	RW	0x0	Secure ARM DMA select interrupt enable Note: the interrupt request signal is active low.
63	AEIRQ	RW	0x0	Not maskable error ARM DMA interrupt enable Note: the interrupt request signal is active low.

The interrupt controller can receive up to 12 external interrupts simultaneously, which are EI0 ~ EI11. And each external interrupt can become one of the following 64 interrupt sources. EINTSEL0, EINTSEL1 and EINTSEL2 registers are used for selecting these interrupt sources.

Table 2.7 Sources of External Interrupt

NUM	External Interrupt Sources	NUM	External Interrupt Sources
0	GPIOA[0]	32	GPIOF[25]
1	GPIOA[1]	33	GPIOF[26]
2	GPIOA[2]	34	GPIOF[27]
3	GPIOA[3]	35	GPIOF[20]
4	GPIOA[4]	36	GPIOF[17]
5	GPIOA[5]	37	GPIOF[13]
6	GPIOA[6]	38	GPIOF[10]
7	GPIOA[7]	39	GPIOF[8]
8	GPIOA[8]	40	GPIOC[28]
9	GPIOA[9]	41	GPIOC[29]
10	GPIOA[10]	42	GPIOC[30]
11	GPIOA[11]	43	GPIOC[31]
12	GPIOA[12]	44	GPIOC[9]
13	GPIOA[13]	45	GPIOC[13]
14	GPIOA[14]	46	GPIOB[28]
15	GPIOA[15]	47	GPIOB[29]
16	GPIOD[5]	48	GPIOB[30]
17	GPIOD[6]	49	GPIOB[31]
18	GPIOD[7]	50	GPIOB[8]
19	GPIOD[8]	51	GPIOB[12]
20	GPIOD[9]	52	GPIOE[4]
21	GPIOD[10]	53	GPIOE[5]

22	GPIOD[13]	54	GPIOE[24]
23	GPIOD[14]	55	GPIOE[25]
24	GPIOD[15]	56	TSWKU
25	GPIOD[16]	57	TSSTOP
26	GPIOD[17]	58	TSUPDN
27	GPIOD[18]	59	-
28	GPIOD[19]	60	-
29	GPIOD[20]	61	PMKUP
30	GPIOF[23]	62	USB_VBON
31	GPIOF[24]	63	USB_VBOFF

EINTSEL0 **0xF0102184**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		EINT[3]SEL								EINT[2]SEL					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		EINT[1]SEL								EINT[0]SEL					

EINTSEL1 **0xF0102188**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		EINT[7]SEL								EINT[6]SEL					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		EINT[5]SEL								EINT[4]SEL					

EINTSEL2 **0xF010218C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		EINT[11]SEL								EINT[10]SEL					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		EINT[9]SEL								EINT[8]SEL					

Interrupt Clear Register (CLR0) **0xF0401008**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: For each bit, the interrupt status is cleared by writing '1' for corresponding bit.

Interrupt Clear Register (CLR1) **0xF040100C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: For each bit, the interrupt status is cleared by writing '1' for corresponding bit.

Interrupt Status Register (STS0) **0xF0401010**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: For each bit, if it is '1', the corresponding interrupt was activated.

Interrupt Status Register (STS1) **0xF0401014**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: For each bit, if it is '1', the corresponding interrupt was activated.

Interrupt Select Register (SEL0) **0xF0401018**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: For each bit, if it is '1', the corresponding interrupt is propagated to nIRQ otherwise to nFIQ.

Interrupt Select Register (SEL1) **0xF040101C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: For each bit, if it is '1', the corresponding interrupt is propagated to nIRQ otherwise to nFIQ.

Interrupt Source Status Register (SRC0) **0xF0401020**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: This represents the status for each interrupt source by XOR with interrupt input and polarity register for the corresponding interrupt source.

Interrupt Source Status Register (SRC1) **0xF0401024**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: This represents the status for each interrupt source by XOR with interrupt input and polarity register for the corresponding interrupt source.

Masked Interrupt Status Register (MSTS0) **0xF0401028**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: This register represents the interrupt status for enabled sources. If a interrupt is not enabled, the corresponding bit is '0'.

Masked Interrupt Status Register (MSTS1) **0xF040102C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: This register represents the interrupt status for enabled sources. If a interrupt is not enabled, the corresponding bit is '0'.

Test Interrupt Source Register (TIG0) **0xF0401030**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: In the case of the corresponding bit in SRC register is '1', the interrupt is activated by writing '1'.

Test Interrupt Source Register (TIG1) **0xF0401034**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: In the case of the corresponding bit in SRC register is '1', the interrupt is activated by writing '1'.

Interrupt Polarity Control Register (POL0) **0xF0401038**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: If the interrupt signal is active-high, the corresponding bit should be '0', otherwise '1' – active-low.

Interrupt Polarity Control Register (POL1) **0xF040103C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: If the interrupt signal is active-high, the corresponding bit should be '0', otherwise '1' – active-low.

IRQ Raw Status Register (IRQ0) **0xF0401040**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: This represents the raw status for IRQ register.

IRQ Raw Status Register (IRQ1) **0xF0401044**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: This represents the raw status for IRQ register.

FIQ Raw Status Register (FIQ0) **0xF0401048**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: This represents the raw status for FIQ register.

FIQ Raw Status Register (FIQ1) **0xF040104C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: This represents the raw status for FIQ register.

Masked IRQ Status Register (MIRQ0) **0xF0401050**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: This represents the masked status for IRQ register.

Masked IRQ Status Register (MIRQ1) **0xF0401054**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: This represents the masked status for IRQ register.

Masked FIQ Status Register (MFIQ0) **0xF0401058**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: This represents the masked status for FIQ register.

Masked FIQ Status Register (MFIQ1) **0xF040105C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: This represents the masked status for FIQ register.

Mode Register (MODE0) **0xF0401060**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: If the corresponding bit is ‘1’, the interrupt acts as level-triggered mode, otherwise edge-triggered mode.

Mode Register (MODE1) **0xF0401064**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: If the corresponding bit is ‘1’, the interrupt acts as level-triggered mode, otherwise edge-triggered mode.

Synchronization Enable Register (SYNC0) **0xF0401068**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: If the corresponding bit is ‘1’, the controller synchronizes the corresponding interrupt source with SMU clock.

Synchronization Enable Register (SYNC1) **0xF040106C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: If the corresponding bit is ‘1’, the controller synchronizes the corresponding interrupt source with SMU clock.

Wakeup Enable Register (WKEN0) **0xF0401070**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: If the corresponding bit is ‘0’, the CPU clock or bus clock can be waked up by the corresponding interrupt source.

Wakeup Enable Register (WKEN1) **0xF0401074**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

Note: If the corresponding bit is ‘0’, the CPU clock or bus clock can be waked up by the corresponding interrupt source.

Edge Trigger Mode Register (MODEA0) **0xF0401078**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: If the corresponding bit is ‘0’ in case of the edge-triggered mode, the interrupt propagated to nFIQ or nIRQ in single edge, otherwise in both edge.

Edge Trigger Mode Register (MODEA1) **0xF040107C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

IRQ Output Mask Register (INTMSK0) **0xF0401100**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHI0	ECC	DMA	TSADC	G2D	3DMMU	3DGP	3DPP	VCDC	JPGE	JPGD	VIPET	LCD1	LCD0	CAM	SC1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC0	EI11	EI10	EI9	EI8	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	SMUI2C	TC1	TC0

Note: If the corresponding bit is ‘1’, interrupt are passed to IRQ or FIQ.

IRQ Output Mask Register (INTMSK1) **0xF0401104**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEIRQ	ASIRQ	AIRQ	APMU	AUDIO	ADMA	DAITX	DAIRX	CDRX	TSIF1	TSIF0	0	0	0	U11H	UOTG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART	SPDTX	SD1	SD0	RTC	RMT	NFC	MS	MPEFEC	I2C	HDD	GPSB	SATA	HDMI	CAN	EHI1

All Register (ALLMSK) **0xF0401108**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-														FIQ	IRQ

Field	Name	RW	Reset	Description
0	IRQ	RW	0x1	IRQ mask register When it is 0, IRQ interrupt is masked.
1	FIQ	RW	0x1	FIQ mask register When it is 0, FIQ interrupt is masked.
31 ~ 2	-	-	-	Undefined

2.3.2 Vectored Interrupt Controller

The vectored interrupt controller can make it possible to service the corresponding interrupt with any other unnecessary comparison operation. The vectored interrupt controller has the 4 registers for this purpose, VAIRQ, VAFIQ, VNIRQ, and VNIRQ.

The VAIRQ and VAFIQ represent the vectored address offset for current activated interrupt in word-based address. For example, if the 2nd priority interrupt has been activated to IRQ, the VAIRQ represents the '4' in VA field below with '0' INV field. The VAFIQ is same as VAIRQ.

The VNIRQ and VNFIQ represent the vectored number offset for current interrupt. For example, if the 2nd priority interrupt has been activated to IRQ, the VNIRQ represents the '1' in VA field below. You can use these features for fast handler service.

IRQ Vector Register (VAIRQ) **0xF0401200**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INV	0														
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VA															

INV [31]	Type	Invalid Field
0	R	Valid for Current Interrupt Source.
1		Invalid for Current Interrupt Source.

VA [8:0]	Type	Invalid Field
n	R	Interrupt Vector Address Offset in Word-Address. This is one of '0', '4', '8', …

FIQ Vector Register (VAFIQ) **0xF0401204**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INV	0														
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VA															

INV [31]	Type	Invalid Field
0	R	Valid for Current Interrupt Source.
1		Invalid for Current Interrupt Source.

VA [8:0]	Type	Invalid Field
n	R	Interrupt Vector Address Offset in Word-Address. This is one of '0', '4', '8', …

IRQ Number Register (VNIRQ) **0xF0401208**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INV	0														
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VN															

INV [31]	Type	Invalid Field
0	R	Valid for Current Interrupt Source.
1		Invalid for Current Interrupt Source.

VN [6:0]	Type	Invalid Field
n	R	Interrupt Vector Address Offset in Word-Address. This is one of '0', '1', '2', …

FIQ Number Register (VNFIQ) 0xF040120C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INV	0														
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VN															

INV [31]	Type	Invalid Field
0	R	Valid for Current Interrupt Source.
1		Invalid for Current Interrupt Source.

VN [6:0]	Type	Invalid Field
n	R	Interrupt Vector Address Offset in Word-Address. This is one of ‘0’, ‘1’, ‘2’, …

Vector Control Register (VCTRL) 0xF0401210

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCL	FPOL	FFLG	IFLG	FHD	IHD	0									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0															

RCL [31]	Type	Clear Interrupt Status
0	R/W	The write operation is needed for clearing interrupt status.
1		The reading the VN or VA register clears the interrupt status.

FPOL [30]	Type	Valid Flag Polarity
0	R/W	The INV field means valid for ‘0’, invalid for ‘1’.
1		The INV field means valid for ‘1’, invalid for ‘0’.

FFLG [29]	Type	Valid Flag Enable
0	R/W	Invalid flag enable for FIQ vector registers.
1		Valid flag enable for FIQ vector registers.

IFLG [28]	Type	Valid Flag Enable
0	R/W	Invalid flag enable for IRQ vector registers.
1		Valid flag enable for IRQ vector registers.

FHD [27]	Type	Hold Enable
0	R/W	Disable the holding vector for FIQ until cleared.
1		Enable the holding vector for FIQ until cleared.

IHD [26]	Type	Hold Enable
0	R/W	Disable the holding vector for IRQ until cleared.
1		Enable the holding vector for IRQ until cleared.

Priority Configuration Register (PRIOn) 0xF0401220+4*n, n=0~15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI[4*n+3]								PRI[4*n+2]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI[4*n+1]								PRI[4*n+0]							

After resetting the system, the priority of each interrupt controller is determined with default value. The default priority of interrupt source 0 is 0 – highest priority and the default priority of interrupt source 63 is 63 – lowest priority. If you want to change the priority of each interrupt sources, you have to write the priority value to the corresponding field.

Be advised that the controller can do the unpredicted operation in case the priorities of different interrupt sources are set with the same value.

PRIOn [7:0]	Type	Priority Configuration Register
PRI[4*n+0]	R/W	Priority for Interrupt [4*n], n=0~15

PRIOn[15:8]	Type	Priority Configuration Register
PRI[4*n+1]	R/W	Priority for Interrupt [4*n+1], n=0~15

PRIOn[23:16]	Type	Priority Configuration Register
PRI[4*n+2]	R/W	Priority for Interrupt [4*n+2], n=0~15

PRIOn[31:24]	Type	Priority Configuration Register
PRI[4*n+3]	R/W	Priority for Interrupt [4*n+3], n=0~15

2.4 Operation & Timing Diagram

2.4.1 How to Configure the Interrupt Source

The following figure shows the active condition and corresponding IRQI waveform. The active condition can be configured by the POL0 and POL1 registers. If the IRQI is low during the active period, the corresponding bit-field of the POL0 and POL1 registers should be '1'.

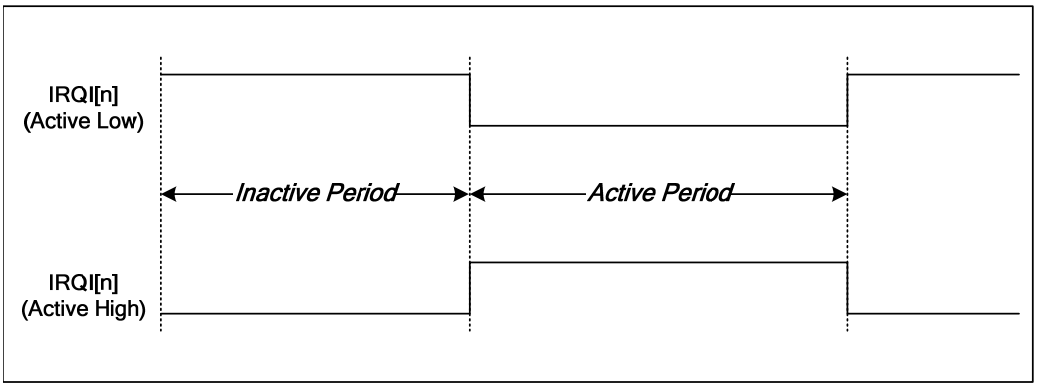


Figure 2.3 Active High vs. Active Low

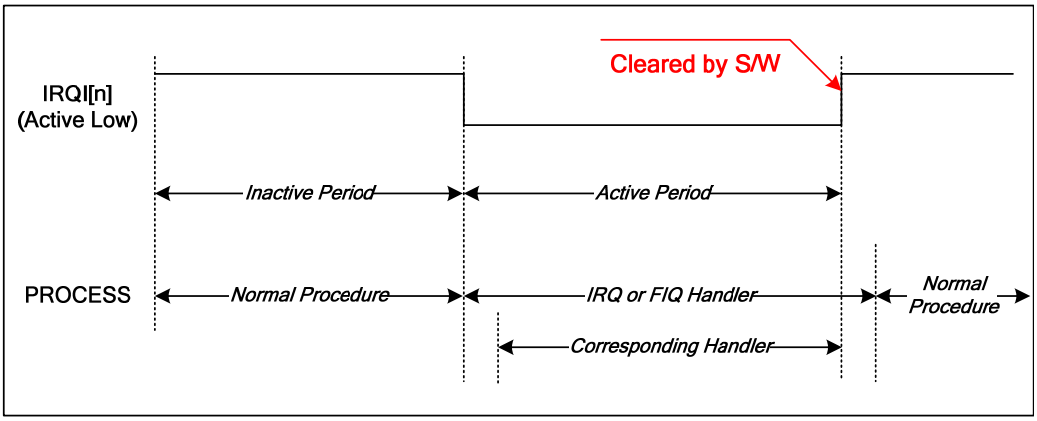


Figure 2.4 Level Triggered vs. Edge Triggered

The above figure shows the IRQI timing diagram and the corresponding S/W flow.
At the falling edge of the IRQI, the nIRQ or nFIQ will be "LOW" to process the interrupt handler. If the IRQI can be cleared during the IRQ or FIQ handler, the level-triggered type is preferred. But, if the clearing the interrupt can not be finished during the IRQ or FIQ handler, the corresponding IRQI should be configured by edge-triggered type because the IRQI can make unexpected IRQ or FIQ handler called continuously until the IRQI changed to inactive state.

But, in the edge-triggered type, the IRQI can be missed in the special case. The following shows the possibility to miss the edge-triggered IRQI signal.

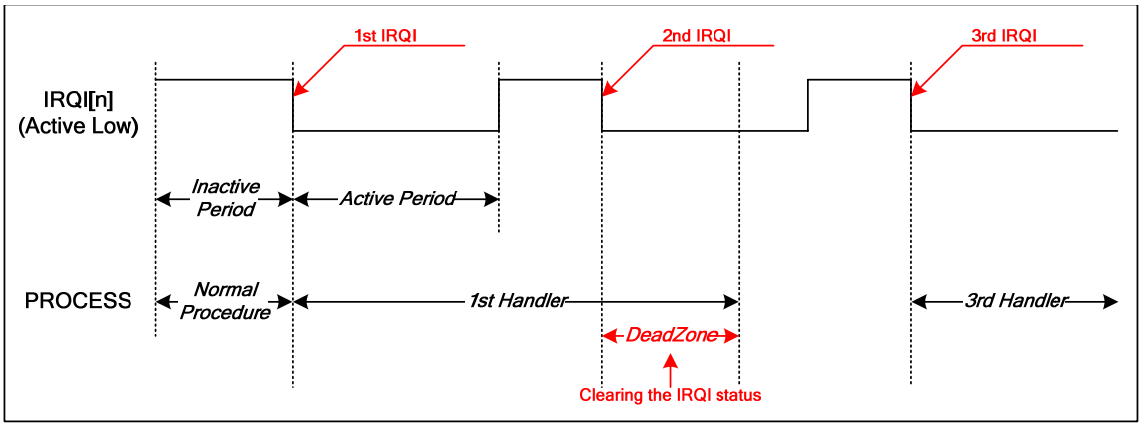


Figure 2.5 A Case of Missing the Edge-Triggered IRQI

If the edge-triggered interrupt occurred, the interrupt controller can not receive the next interrupt until the corresponding interrupt status being cleared. In the above figure, if the "1st IRQI' would be cleared in the "DeadZone", the 2nd IRQI interrupt can not be received by the interrupt controller. After than, the controller can accept the "3rd IRQI" event in the next falling edge. For the above case, the edge-triggered interrupt should be taken care of.

The following paragraphs and figure are about the difference between sync and async mode.

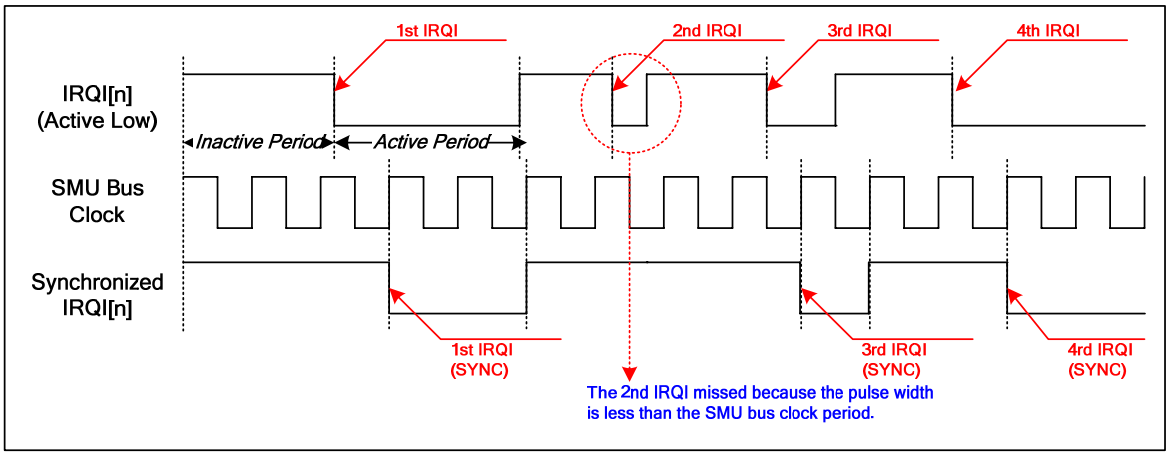


Figure 2.6 The timing Relations between Original IRQI and Synchronized IRQI

As shown in the above figure, if the active pulse width of the IRQI is greater than the SMU bus clock period, the interrupt controller can accept the corresponding IRQI with timing of the synchronized IRQI. But, if the pulse width is shorter than the clock period such as "2nd IRQI", the interrupt controller can't recognize the corresponding IRQI because of the corresponding interrupt can be registered by the SMU bus clock as shown in the "Synchronized IRQI[n]" in the above figure.

Finally, the INTMSK0, INTMSK1 and ALLMSK register can be used for masking the interrupt temporarily. The IEN0 and IEN1 register can also be used to mask. But, major difference is the possibility of accepting the interrupt source during the masked period. The INTMSK0, INTMSK1 and ALLMSK can recognize the interrupt during the masked period but IEN0 and IEN1 can not accept the interrupt during the masked period.

2.4.2 How to Enable Interrupt for IRQI

Before enabling the interrupt for the corresponding IRQI, the followings should be determined.

1. Is the IRQI level-triggered or edge-triggered ? (MODE0 and MODE1)
2. If edge-triggered case, which edge(s) is(are) used ? (MODEA0 and MODEA1)
3. Which type of the corresponding interrupt be used ? IRQ or FIQ (SEL0 and SEL1)
4. Whether the synchronization function is required or not? The recommended is to use this function. (SYNC0 and SYNC1)

After the above configurations decided, you have only to enable the interrupt with IEN0 and IEN1.

2.4.3 Recommended IRQI Configurations

The following table is the recommended configurations for the corresponding IRQI. ***But, the following configurations can be changed according to the system or software platform.*** The SYNC field in the following table is recommended to '1', but if the SMU bus clock is disabled, the interrupt controller can not receive the interrupt.

Table 2.8 Recommended IRQI Configurations

Field	Name	MODE	MODEA	POL	SYNC	Related Hardware(s)
0	TC0	1	-	0	1	TIMER
1	TC1	1	-	0	1	TIMER
2	SMUI2C	1	-	0	1	SMU_I2C
3~14	EI0~EI11	-	-	-	1	External Interrupts * Refer to the next table.
15	SC0	1		1	1	Mem-to-Mem Scaler Controller 0
16	SC1	1		1	1	Mem-to-Mem Scaler Controller 1
17	CAM	1	-	0	1	Camera Interface
18	LCD0	1		1	1	LCD Controller 0

19	LCD1	1		1	1	LCD Controller 1
20	VIPET	1		0	1	VIQE (Video Enhancer)
21	JPGD	1	-	0	1	JPEG Decoder
22	JPGE	1	-	0	1	JPEG Encoder
23	VCDC	1	-	0	1	Video Codec
24	3DPP	1	-	0	1	3D Pixel Processor Interrupt
25	3DGP	1	-	0	1	3D Geometry Processor Interrupt
26	3DMMU	1	-	0	1	3D MMU Interrupt
27	G2D	1	-	0	1	Overlay Mixer (2D Graphic Engine)
28	TSADC	1	-	0	1	Touch Screen Controller
29	DMA	1	-	0	1	DMA
30	ECC	1	-	0	1	ECC
31	EHI0	1	-	0	1	External Host Interface Controller 0
32	EHI1	1	-	0	1	External Host Interface Controller 1
33	CAN	1	-	0	1	CAN Controller
34	HDMI	1	-	0	1	HDMI Link Controller
35	SATA	1	-	0	1	SATA Link Controller
36	GPSB	1	-	0	1	GPSB Controller
37	HDD	1	-	0	1	IDE Controller
38	I2C	1	-	0	1	I2C Controller
39	MPEFEC	1	-	0	1	MPEFEC Accelerator
40	MS	1	-	1	1	Memory Stick Controller
41	NFC	1	-	0	1	NAND Flash Controller
42	RMT	1	-	0	1	Remocon Controller
43	RTC	1	-	0	1	RTC(Real-Time Clock)
44	SD0	1	-	0	1	SDMMC Controller 0
45	SD1	1	-	0	1	SDMMC Controller 1
46	SPDTX	0	0	0	0	SPDIF Transmitter
47	UART	1	-	0	1	UART Controller
48	UOTG	1	-	0	1	USB 2.0 OTG
49	U11H	1	-	0	1	USB 1.1 Host Controller
53	TSIF0	0	0	0	0	TS Interface Controller 0
54	TSIF1	0	0	0	0	TS Interface Controller 1
55	CDRX	0	0	0	0	CDIF
56	DAIRX	0	0	0	0	I2S Receiver
57	DAITX	0	0	0	0	I2S Transmitter
58	ADMA	0	0	0	0	Audio DMA Controller
59	AUDIO	0	0	0	0	Audio DMA Controller
60	APMU	1	-	1	1	ARM System Metrics interrupt enable
61	AIRQ	1	-	1	1	Non secure ARM DMA interrupt enable
62	ASIRQ	1	-	1	1	Secure ARM DMA select interrupt enable
63	AEIRQ	1	-	1	1	Not maskable error ARM DMA interrupt enable

The following sources can be mapped to external interrupt source. And the recommended configurations are shown in the following table.

Table 2.9 Recommended External Interrupt Configurations

Field	Name	MODE	MODEA	POL	SYNC	Related Hardware(s)
1	GPIOs	-	-	-	1	The configurations can be changed to the external components.
2	TSWKU	1	-	0	1	Wake-up interrupt from the Touch Screen Controller.
3	TSSTOP	0	0	0	0	Stop interrupt from Touch Screen Controller
4	TSUPDN	0	1	1	0	Up/Down interrupt from Touch Screen Controller

5	PMWKUP	0	0	0	0	RTC Wakeup Interrupt * The polarity can be changed to active low.
6	USB_VBON	0	0	0	0	
7	USB_VBOFF	0	0	0	0	

For the GPIO case, the values of MODE, MODEA, POL should be determined appropriatly according the external component or any type of interrupt source,

2.4.4 How to Use Vectored Interrupts

In the real-time system, the interrupt latency and interrupt processing time is very important. The vectored interrupt controller can make them faster than that of non-vectored system. To use of the vectored interrupt controller, the interrupt priorities according to the IRQI should be pre-defined. For the safe system management, the priorities is preferred not to change in run-time operation.

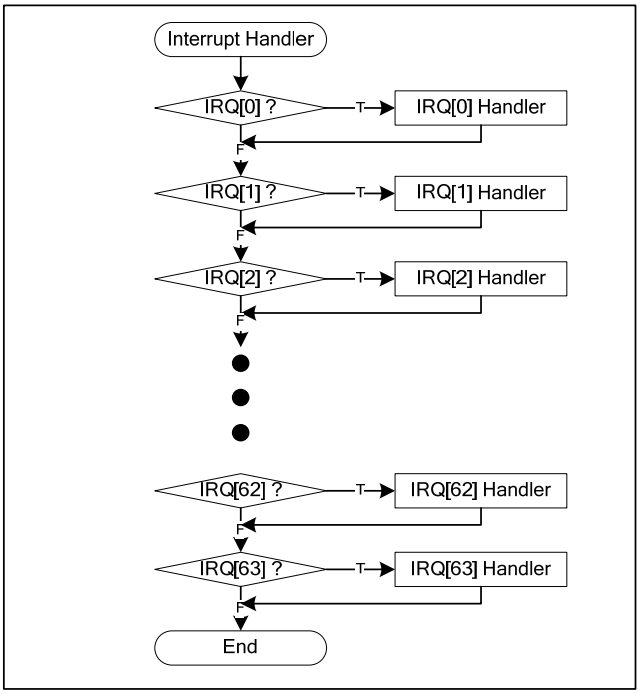


Figure 2.7 Structure of the Non-Vectored Interrupt Handler

The above figure shows the software structure of the non-vectored interrupt. In the above figure, the IRQ[0] has the highest priority and IRQ[63] has the lowest priority which can be determined and changed by software. The software searches from IRQ[0] to IRQ[63] consequently to find which interrupt occurred. The IRQ[0], which has highest priority, can be processed lowest latency. But for the IRQ[63], lowest priority, even if there is no interrupt in the IRQ[0] ~ IRQ[62], the large latency for IRQ[63] is required because of searching routine.

The following figure shows that the structure of the vectored interrupt.

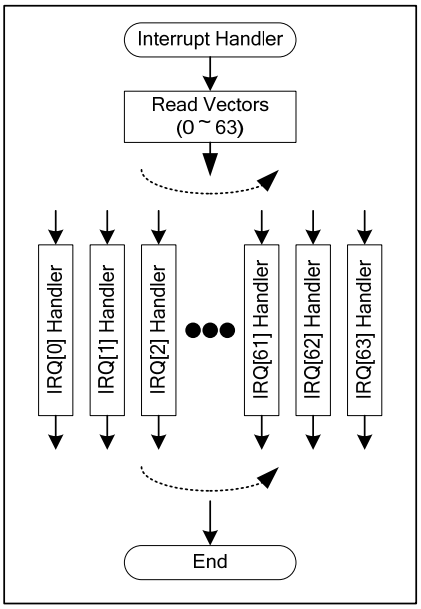


Figure 2.8 Structure of the Vectored Interrupt Handler

All the interrupt handler has the same latency as you can see in the above figure because the vector indicating which interrupt occurred can be read. The software can branch to the corresponding handler directly with function pointer array.

```
typedef void Handler(void);                          // Handler Function Prototype

unsigned int gnIntHandler[64];                       // Vector Table
                                                     // should be initialized at the system boot-up


__irq IRQHandler (void)
{
  Handler *fUserHandler;
  unsigned int nVectorID;

  while (1) {                                        // Looping Until All the interrupts processed
    nVectorID = VNIRQ;                               // Reading Vector
    if ( nVectorID & 0x80000000) break;              // Checking Invalid Interrupt or Not
    fUserHandler = (Handler *) (gnIntHandler[nVectorID]);
    fUserHandler ();                                 // Branching to the handler
    VNIRQ = nVectorID;                               // Clearing the corresponding interrupt
  }
}
```

Figure 2.9 Example Code of the Vectored Interrupt Handler

The above figure shows the example code for the IRQ handler. In the FIQ case, the main structure is same as IRQ. The bold, italic and underlined indicates the register value.
The “gnIntHandler” arrays should have the base address for all interrupts and the handler function should have same prototype.

3 Timer / Counter

3.1 Overview

The TCC8900 has four 16-bits, two 20-bits, and one 32-bits timers/counters. 16-bits and 20-bits timer/counters have three registers for basic operation modes. Refer to register description table for details. When operating in counter modes, External interrupt pin is used as counting clock for that counter.

The main clock frequency of timer counter is determined by TCLK, XCLK, ZCLK frequency. With the 12bit internal basic counter, the timer counter can generate various intervals from microseconds to seconds unit.

The TCLK is for 16/20bits timer/couter named T-Timer, the XCLK is for watchdog timer named X-Timer, and finally ZCLK is for 32bits timer named Z-Timer. The following figure shows the overall timer/counter block diagram.

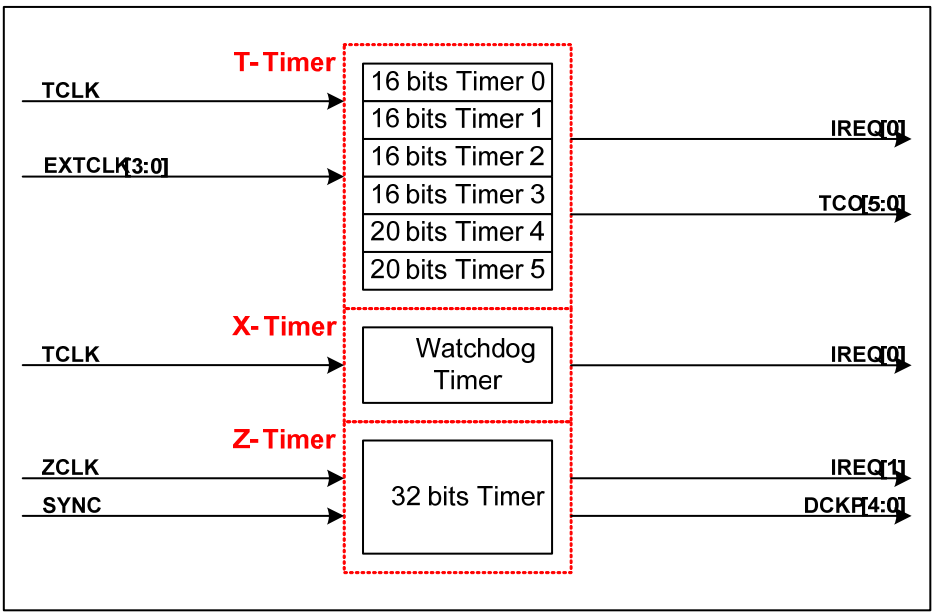


Figure 3.1 Overall Timer/Counter Block Diagram

The following figure represents the block diagram of T-Timer.

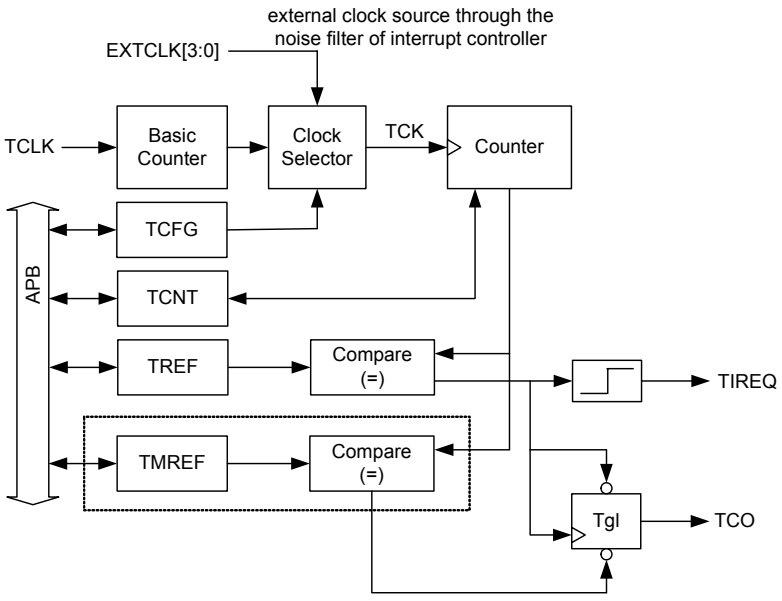


Figure 3.2 16-bit and 20bit Timer/Counter Block Diagram

The following figure shows block diagram of the watchdog timer.

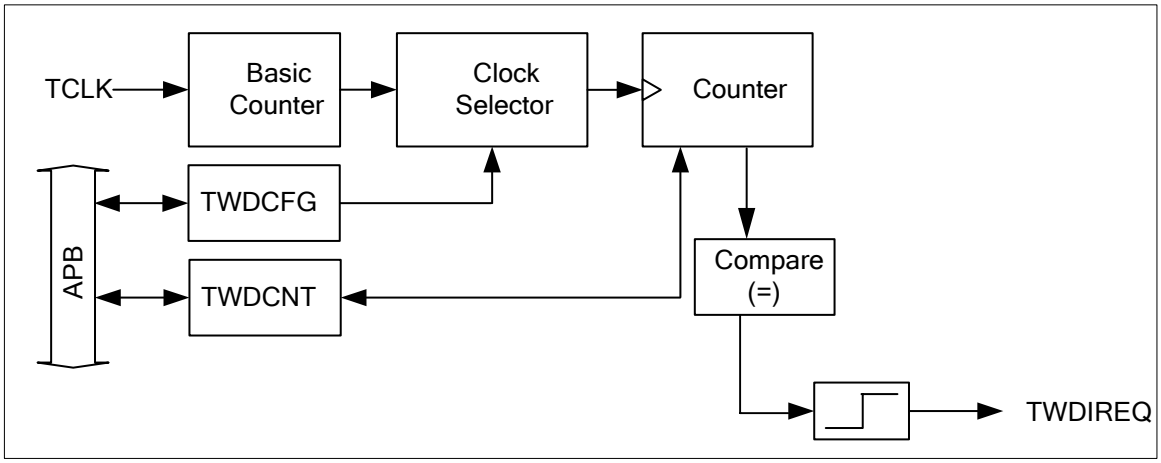


Figure 3.3 Watchdog Timer Block Diagram

As illustrated in the figure below, TC32 consists of a pre-scale counter, main counter and two comparators. The pre-scale counter is a simple 24-bit up-counter which always counts from zero to PRESCALE value programmed in TC32EN register. The 32-bit main counter is incremented only when the prescale counter reaches PRESCALE value.

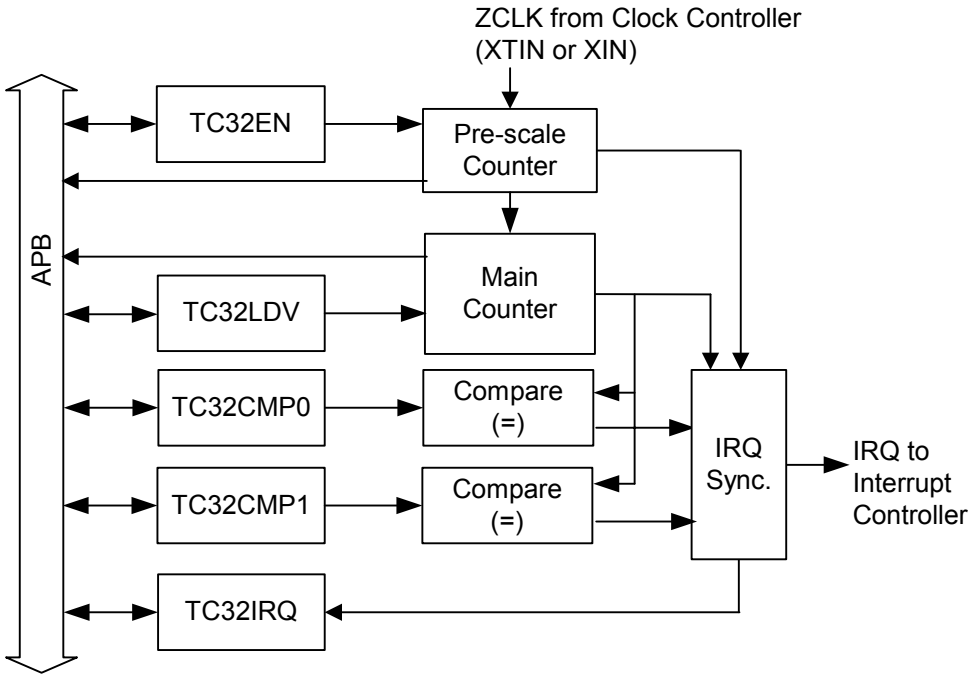


Figure 3.4 32-bit Counter Block Diagram

3.2 Signal Descriptions

3.2.1 Global Signals

Table 3.1 Global Signals

Name	Direction	Descriptions	Related Blocks
PCLK	Input	Timer Bus Clock (SMU Bus Clock)	CKC
PRESETn	Input	Reset Signal	CKC

3.2.2 T-Timer/Counter(16/20bits Timer/Counter) Related Signals

Table 3.2 T-Timer/Counter(16/20bits Timer/Counter) Related Signals

Name	Direction	Descriptions	Related Blocks
TCLK	Input	Timer Operating Clock	PCLKs[1] of CKC
EXTCLK[3:0]	Input	External Counter Clock Sources	IRQP[3:0] of VPIC
TCO[5:0]	Output	Timer Counter Outputs [0] : 16bits Timer 0 [1] : 16bits Timer 1 [2] : 16bits Timer 2 [3] : 16bits Timer 3 [4] : 20bits Timer 4 [5] : 20bits Timer 5	GPIO
IREQ[0]	Output	T-Timer Interrupt Output	IRQI[0] of VPIC

3.2.3 X-Timer (Watchdog Timer)

Table 3.3 X-Timer (32bits Timer)

Name	Direction	Descriptions	Related Blocks
TCLK	Input	Timer Operating Clock	PCLKs[0] of CKC
WDTRST	Output	Watchdog Reset but **Not Used**	-
IREQ[0]	Output	T-Timer Interrupt Output	IRQI[0] of VPIC

3.2.4 Z-Timer (32bits Timer)

Table 3.4 Interrupt Output Signals (to ARM)

Name	Direction	Descriptions	Related Blocks
ZCLK	Input	Timer Operating Clock	PCLKs[2] of CKC
DCKP[4:0]	Output	Pulse Outputs but **Not used**	-
SYNC	Input	Stuck at HIGH	-
IREQ[1]	Output	Z-Timer Interrupt Output	IRQI[1] of VPIC

3.3 Register Description

The following table explains the registers of each timer counter. The address of each timer counter is 16bytes aligned. The base address of timer counter is 0xF0403000.

The number n represents for each timer/counter. In case of timer/counter 4, 5 (that is n = 4 or 5) the TREF, TCNT register has 20bit resolution. It can be used for generating long time events.

Table 3.5 Timer/Counter Register Map (Base Address = 0xF0403000)

Name	Address	Type	Reset	Description
TCFG0	0x00	R/W	0x00	Timer/Counter 0 Configuration Register
TCNT0	0x04	R/W	0x0000	Timer/Counter 0 Counter Register
TREF0	0x08	R/W	0xFFFF	Timer/Counter 0 Reference Register
TMREF0	0x0C	R/W	0x0000	Timer/Counter 0 Middle Reference Register
TCFG1	0x10	R/W	0x00	Timer/Counter 1 Configuration Register
TCNT1	0x14	R/W	0x0000	Timer/Counter 1 Counter Register
TREF1	0x18	R/W	0xFFFF	Timer/Counter 1 Reference Register
TMREF1	0x1C	R/W	0x0000	Timer/Counter 1 Middle Reference Register
TCFG2	0x20	R/W	0x00	Timer/Counter 2 Configuration Register
TCNT2	0x24	R/W	0x0000	Timer/Counter 2 Counter Register
TREF2	0x28	R/W	0xFFFF	Timer/Counter 2 Reference Register
TMREF2	0x2C	R/W	0x0000	Timer/Counter 2 Middle Reference Register
TCFG3	0x30	R/W	0x00	Timer/Counter 3 Configuration Register
TCNT3	0x34	R/W	0x0000	Timer/Counter 3 Counter Register
TREF3	0x38	R/W	0xFFFF	Timer/Counter 3 Reference Register
TMREF3	0x3C	R/W	0x0000	Timer/Counter 3 Middle Reference Register
TCFG4	0x40	R/W	0x00	Timer/Counter 4 Configuration Register
TCNT4	0x44	R/W	0x00000	Timer/Counter 4 Counter Register
TREF4	0x48	R/W	0xFFFFF	Timer/Counter 4 Reference Register
TCFG5	0x50	R/W	0x00	Timer/Counter 5 Configuration Register
TCNT5	0x54	R/W	0x00000	Timer/Counter 5 Counter Register
TREF5	0x58	R/W	0xFFFFF	Timer/Counter 5 Reference Register
TIREQ	0x60	R/W	0x0000	Timer/Counter n Interrupt Request Register
TWDCFG	0x70	R/W	0x0000	Watchdog Timer Configuration Register
TWDCLR	0x74	W	-	Watchdog Timer Clear Register
TWDCNT	0x78	R/W	0xFFE0	Watchdog Timer Counter Register
TC32EN	0x80	R/W	0x00007FFF	32-bit Counter Enable / Pre-scale Value
TC32LDV	0x84	R/W	0x00000000	32-bit Counter Load Value
TC32CMP0	0x88	R/W	0x00000000	32-bit Counter Match Value 0
TC32CMP1	0x8C	R/W	0x00000000	32-bit Counter Match Value 1
TC32PCNT	0x90	R/W	-	32-bit Counter Current Value (pre-scale counter)
TC32MCNT	0x94	R/W	-	32-bit Counter Current Value (main counter)
TC32IRQ	0x98	R/W	0x0000----	32-bit Counter Interrupt Control

The registers prefixed by “TC32” are for the Z-Timer and the registers prefixed by “TWD” are for the X-Timer. Others are for T-Timers.

Timer/Counter n Configuration Register (TCFG*n*) **0xF04030*n*0**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						STOP	CC	POL	TCKSEL			IEN	PWM	CON	EN

STOP [9]	STOP Mode
0	Continuous counting mode.
1	If TCNTn is equal to the TREFn, the TCNTn counter stop to increment.

CC [8]	Clear Count
0	TCNTn is not cleared.
1	TCNTn is cleared to zero.

POL [7]	TCK Polarity
0	TCNTn is incremented at rising edge of the selected counting clock
1	TCNTn is incremented at falling edge of the selected counting clock

TCKSEL [6:4]	TCK Select
k = 0 ~ 4	TCK is internally generated from divider circuit. It is driven by TCLK, and this value determines the division factor of this circuit. Division factor is $2^{(k+1)}$.
k = 5, 6	TCK is internally generated from divider circuit. It is driven by TCLK, and this value determines the division factor of this circuit. Division factor is 2^{2k}
k = 7	TCK is the external pin shared by external interrupt signal. In TCC8900, there are 4 external pins for this purpose, so this configuration is valid only for timer/counter 3 ~ 0. (not for timer/counter 5, 4)

IEN [3]	Interrupt Enable
1	Enable Timer/Counter interrupt
0	Disable Timer/Counter interrupt

PWM [2]	PWM Mode Enable
1	Enable PWM mode Timer/Counter output is changed at every time the TCNTn is equal to TREFn and TMREFn value. It can be used to generate PWM waveform, by changing TMREFn while fixing TREFn. (where, TREFn > TMREFn)
0	Disable PWM mode Timer/Counter output can be changed only when the TCNTn is equal to TREFn. It can be used to generate a rectangular pulse of variable frequency.

The output of Timer/Counter 0, 1, 2, and 3 can be monitored through the ports. Refer to the GPIO & Port Multiplexor.

CON [1]	Continue Counting
0	When the TCNTn is reached to TREFn, TCNTn restarts counting from 0 at the next pulse of selected clock source.
1	The TCNTn continues counting from the TREFn.

If the STOP bit is set, this bit is meaningless.

EN [0]	Timer/Counter Enable
0	Timer counter is disabled.
1	Timer counter is enabled.

Following figure illustrates the basic behavior of timer / counter.

Timer/Counter n Counting Register (TCNTn) **0xF04030n4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0												TCNTn[19:16]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCNTn[15:0]															

TCNTn is increased by 1 at every pulse of selected clock source. TCNTn can be set to any value by writing to this register. In case of timer 4 and timer 5, it has 20 bits, otherwise it has 16 bits.

Timer/Counter n Counting Reference Register (TREFn) **0xF04030n8**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0												TREFn[19:16]			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TREFn[15:0]															

When TCNTn is reached at TREFn and the CON flag of TCFGn register is set to 1, the TCNTn is cleared to 0 at the next pulse of selected clock source. According to the TCFGn settings, various kinds of operations may be done. In case of timer 4 and timer 5, it has 20 bit, otherwise it has 16 bit.

Timer/Counter n Middle Reference Register (TMREFn) **0xF04030nC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TMREFn[15:0]															

When TCNTn is reached at TMREFn and the PWM flag of TCFGn register is set to 1, the timer output of TCOn is cleared to 0 at the negative edge of that pulse of selected clock source. The TCOn is set to 1 when the TCNTn is reached at TREFn. (refer TREFn). So you can generate PWM signal by modifying TMREFn(n=0~3) between 0 ~ (TREFn-1).

Timer/Counter Interrupt Request Register (TIREQ) **0xF0403060**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	TWF	TF5	TF4	TF3	TF2	TF1	TF0	0	TWI	TI5	TI4	TI3	TI2	TI1	TI0

TWF [14]	Watchdog Timer Flag
1	Watchdog timer has reached to its reference value.

TFn [13:8]	Timer/Counter n Flag
1	Timer/counter n has reached to its reference value.

TWI [6]	Type	Watchdog Timer Interrupt Request Flag
1	Read	Watchdog timer has generated its interrupt.
1	Write	Watchdog timer interrupt is cleared.

TIn [5:0]	Type	Timer/Counter n Interrupt Request Flag
1	Read	Timer/counter n has generated its interrupt.
1	Write	Timer/counter n interrupt flag is cleared.

If a timer n has reached its reference value, the TFn is set. (bit n represents for Timer n). If its interrupt request is enabled by set bit 3 of TCFGn register, then the TIn is set. If the TC bit of IEN register is set, the timer interrupt is really generated and this TIREQ register can be used to determine which timer has requested the interrupt. After checking these flags, user can clear these TFn and TIn field by writing “1” to corresponding TFn or TIn bit field.

Watchdog Timer Configuration Register (TWDCFG) 0xF0403070

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0									TCLKSEL			IEN	0		EN

TCKSEL [6:4]	TCK Select
k = 4	TCK is internally generated from divider circuit. It is driven by TCLK, and this value determines the division factor of this circuit. Division factor is 25.
k = 5, 6	TCK is internally generated from divider circuit. It is driven by TCLK, and this value determines the division factor of this circuit. Division factor is 22k.
k = 0~3,7	Undefined. Should not be used.

IEN [3]	Interrupt Enable
1	Watchdog Timer Interrupt is enabled. This field is valid only if RST field is set to 0.

EN [0]	Watchdog Timer Enable
1	Watchdog timer is enabled. If the watchdog timer is disabled, its counter goes to 0xE0, so when it is first enabled, user must clear the counter by writing to TWDCLR register.

Watchdog timer is used for the system not to be stuck by generating a reset pulse automatically when the watchdog timer counter overflows to zero. It has 8bit counter and when this counter overflows from 0xFF to 0x00, the reset or interrupt is generated.
The programmer must clear the watchdog counter before it overflows by writing any value to TWDCLR register. The duration can be chosen by selecting TCKSEL field appropriately.

Watchdog Timer Clear Register (TWDCLR) 0xF0403074

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Any Value															

The watchdog timer counter can be cleared to 0 by writing any value to this register. If it is not cleared before it overflows, the watchdog timer generate reset signal to the entire component of chip.

Watchdog Timer Counting Register (TWDCNT) 0xF0403078

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TWDCNT [15:0]															

TWDCNT is increased by 1 at every pulse of selected clock source. TWDCNT can be set to any value by writing to this register.

TC32 Enable / Pre-scale Value Register (TC32EN) 0xF0403080

Bit	Name	Default	R/W	Description
31:30	Reserved	0	R	
29	LDM1	0	R/W	Re-load counter when the counter value is matched with CMP1. LOADZERO bit below selects the counter load(start) value.
28	LDM0	0	R/W	Re-load counter when the counter value is matched with CMP0 LOADZERO bit below selects the counter load(start) value.
27	Reserved	0	R	
26	STOPMODE	0	R/W	0 = Free Running Mode, 1 = Stop Mode.
25	LOADZERO	0	R/W	By default, counter starts from LOADVAL. When this bit is enabled (1), the counter is forced to count from “0” to “LOADVAL – 1”.
24	ENABLE	0	R/W	Counter Enable 0 = Disable, 1 = Enable
23:0	PRESCALE	0x007FFF	R/W	Pre-scale counter load value. The pre-scale counter always runs from “0” up to PRESCALE. The default value is for 1Hz counter when ZCLK = XTIN (32.768kHz).

Possible counter modes are described in the table below.

Table 3.6 TC32 Count Mode

Mode	TC32EN Register Bits			Main Counter Operation	
	LOADZERO	LDM1	LDM0	Start Count Value	End Count Value
0	0	0	0	LOADVAL	0xFFFFFFFF
1	0	0	1	LOADVAL	CMP0 (if LOADVAL < CMP0)
2	0	1	0	LOADVAL	CMP1 (if LOADVAL < CMP1)
3	0	1	1	LOADVAL	CMP0 (if LOADVAL < CMP0 ≤ CMP1) or CMP1 (if LOADVAL < CMP1 ≤ CMP0)
4	1	0	0	0	LOADVAL – 1
5	1	0	1	0	CMP0 (if LOADVAL > CMP0)
6	1	1	0	0	CMP1 (if LOADVAL > CMP1)
7	1	1	1	0	CMP0 (if LOADVAL > CMP1 ≥ CMP0) or CMP1 (if LOADVAL > CMP0 ≥ CMP1)

Refer to register descriptions below for CMP0, CMP1 and LOADVAL.
Mode0 can be used as 1Hz counter mode, if PRESCALE = 0x007FFF, STOPMODE = 0, ZCLK = XTIN (32.768kHz)

TC32 Load Value Register (TC32LDV) **0xF0403084**

Bit	Name	Default	R/W	Description
31: 0	LOADVAL	0x00000000	R/W	Counter Load Value.

The counter is restarted whenever one of the TC32En and TC32LDV is written.

TC32 Match Value 0 Register (TC32CMP0) **0xF0403088**

Bit	Name	Default	R/W	Description
31: 0	CMP0	0x00000000	R/W	Counter Match Value

TC32 Match Value 1 Register (TC32CMP1) **0xF040308C**

Bit	Name	Default	R/W	Description
31: 0	CMP1	0x00000000	R/W	Counter Match Value

TC32 Pre-scale Counter Current Value Register (TC32PCNT) **0xF0403090**

Bit	Name	Default	R/W	Description
31:24	Reserved	0x00	R	
23: 0	PCNT	0x000000	R	Pre-scale counter current value. The AHB system clock must be three times faster than the frequency of ZCLK to read valid value.

TC32 Main Counter Current Value Register (TC32MCNT) **0xF0403094**

Bit	Name	Default	R/W	Description
31: 0	MCNT	0x00000000	R	Main counter current value. When RSYNC is enabled, the AHB system clock must be faster than the frequency calculated below. (ZCLK frequency) / (PRESCALE + 1) * 3

TC32 Interrupt Control Register (TC32IRQ) **0xF0403098**

Bit	Name	Default	R/W	Description
31	IRQCLR	0	R/W	Interrupt Clear Control. When this bit is 0, interrupt raw status bits (IRQRSTAT) are cleared by reading this register. When this bit is set, IRQSTAT bits are cleared only if written with non-zero value.
30	RSYNC	0	R/W	Synchronization control for Counter Current Value Registers (TC32PCNT and TC32MCNT). 0 = Enable, 1 = Disable.
29:24	BITSEL	0x00	R/W	Counter bit selection value for interrupt generation. Any one of the counter bits {MCNT[31:0], PCNT[23:0]} selected by BITSEL is used to generate an interrupt. 0x00 ~ 0x17 : PCNT[0] ~ PCNT[23] 0x18 ~ 0x38: MCNT[0] ~ MCNT[31]
23:21	Reserved	0	R/W	
20	IRQEN[4]	0	R/W	Enable Interrupt at the rising edge of a counter bit selected by BITSEL.
19	IRQEN[3]	0	R/W	Enable Interrupt at the end of pre-scale count
18	IRQEN[2]	0	R/W	Enable Interrupt at the end of count
17	IRQEN[1]	0	R/W	Enable Interrupt when the counter value matched with CMP1
16	IRQEN[0]	0	R/W	Enable Interrupt when the counter value matched with CMP0
15:13	Reserved	0	R/W	
12:8	IRQRSTAT	0x00	R/W	Interrupt Raw Status. Refer to the description for IRQEN above.
7:5	Reserved	0	R/W	
4:0	IRQMSTAT	0x00	R/W	Masked Interrupt Status = IRQRSTAT & IRQEN

The IRQEN[n] is active high, ‘1’ means enabled and ‘0’ means disabled. And the IRQRSTAT[n] is active high, ‘1’ for valid and ‘0’ for invalid.

3.4 Operation & Timing Diagram

3.4.1 How to Run 16bits Timer

The Timer/Counter 0 ~ 3 are belongs to 16-bits timer group. The 16bits timer uses not only the bus clock but also TCLK. The TCO configuration is required to indicate the end of timer counting after clock configuration. After the CFG, CNT, REF registers are configured, you have only to set enable bit in the CFG to start counting. The MREF or PWM bit can be configured if required.

The TCK in the following figure starts to working if the enable bit in the CFG register. As shown in the following figure, the TCO can be inverted at the time in which TCNT and TREF are same and makes the interrupt to CPU.

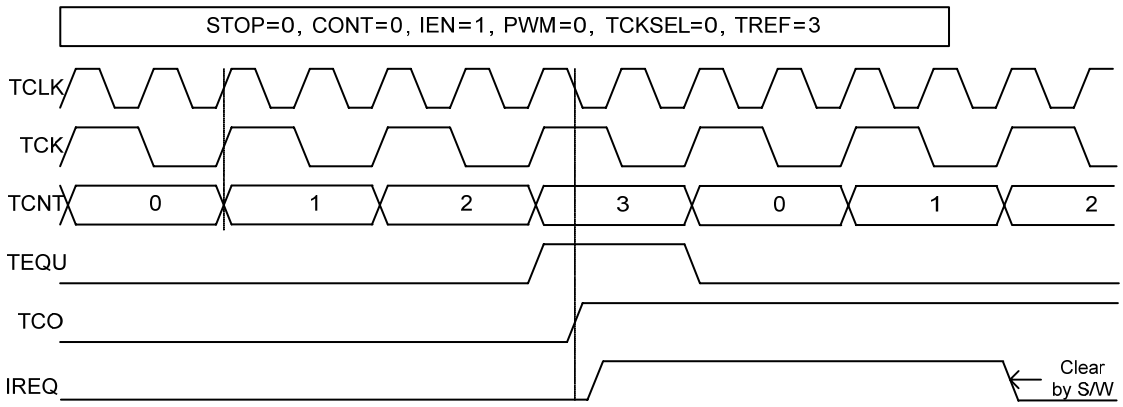


Figure 3.5 The Basic Timing Diagram of the Timer

You should take the cautions in using TCO into consideration in the case that the STOP bit is ‘1’. The TCO will be inverted in the TEQU being ‘1’ in the following figure. So, if the STOP was ‘1’, the TCNT will be stopped after it is same as the TREF and it makes the TCO will be toggled because the TEQU is always ‘1’. To avoid this toggling of the TCO, the enable bit of the CFG register should be cleared.

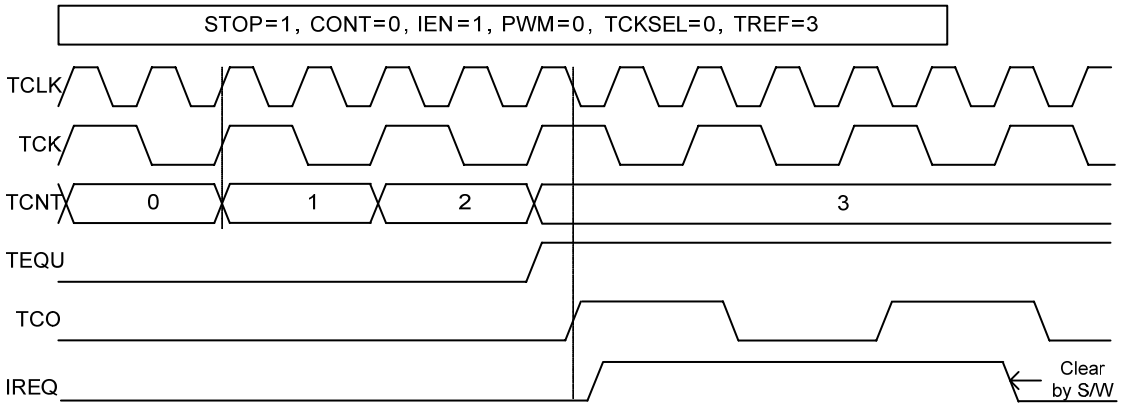


Figure 3.6 The Waveform in Case of STOP being ‘1’

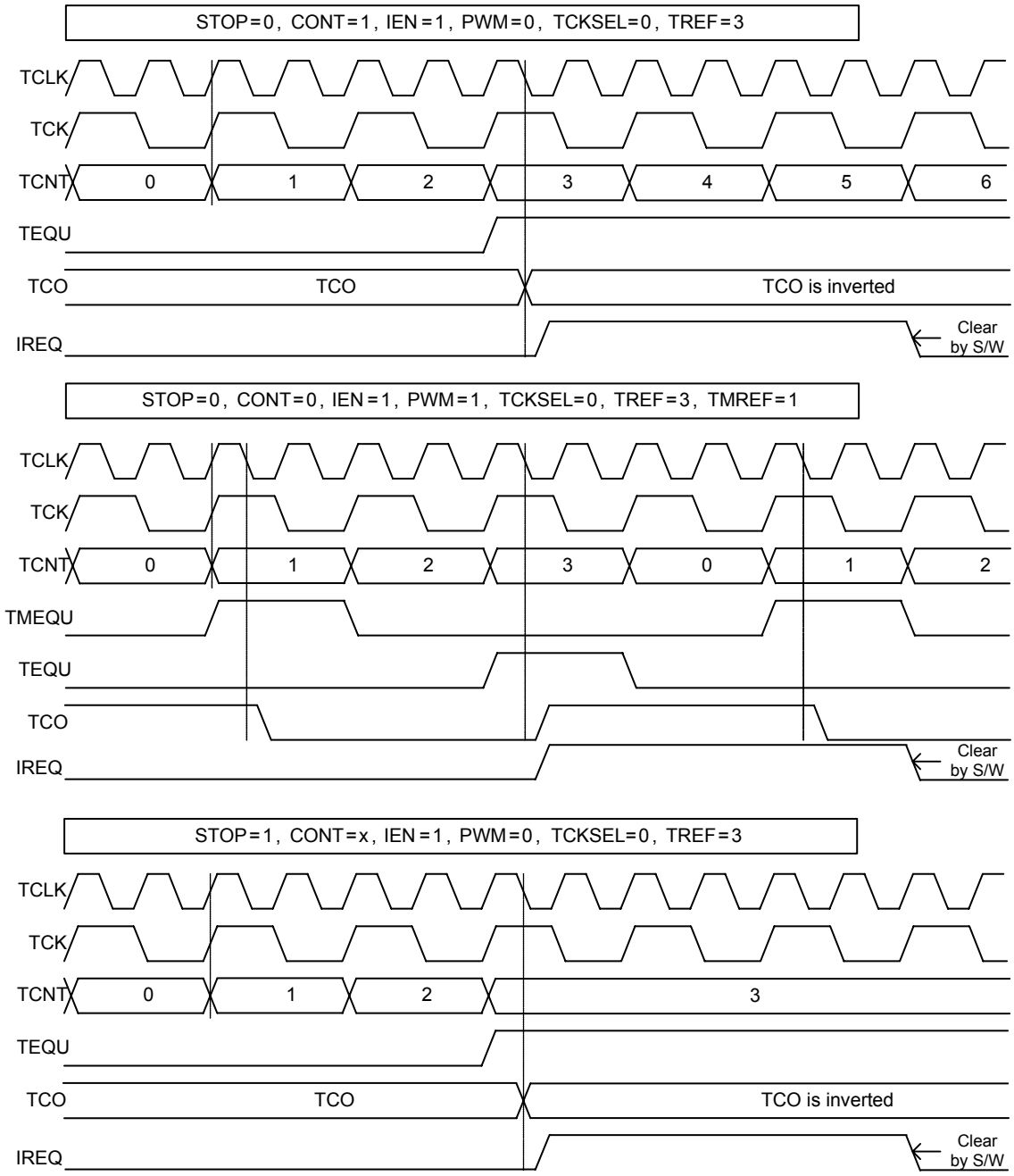


Figure 3.7 Timing Diagram of Timer/Counter

3.4.2 How to Run 20bits Timer

The Timer/Counter 4 and 5 are belongs to 20bits timer. Basically, the 20bits timer operation is same as the 16bits, but the 20bits timer does not have the MREF register and bitwidth of the CNT and REF counter is 20bits. The TCO output is same as the 16bits timer.

3.4.3 How to Run External Clock Counter

The external clocks can be used as the input clock to the 16bits timer. If the TCLK field of the CFG register for the 16bits timer is ‘7’, the 16bits timer uses the external clock as the operating clock.

3.4.4 How to Run Watchdog Timer

The watchdog timer uses the bus clock and TCLK. The bus clock is used to interface control or configuration registers and TCLK is used as the main operationg clock. The watchdog timer does not have the TCO output, so only interrupt can be used. The TCK selection bits should be one of the 4,5,6. In the another case, the clock could not be generated and does not work properly.

3.4.5 How to Run 32bits Timer

The 32bits timer uses the bus clock and ZCLK. Same as the watchdog timer, there is no TCO output and only interrupt can indicate the end of the timer operation. The basic operation is same as the other timers, but there are many counting modes which can make more flexible to use in the system. Refer to 3.6 Table

3.4.6 Interrupt Structure for IREQ[0] – 16/20bits/Watchdog Timer/Counter,

The following figure shows the interrupt structure of the 16bits, 20bits, watchdog timer and how to be transferred to the main interrupt. There is no timing relation between various interrupt sources, the IREQ[0] should be used as the level-sensitive interrupt type and each interrupt status should be cleared in the handler routine as soon as possible.

3.4.7 Interrupt Structure for IREQ[1] – 32bits Timer/Counter

The IREQ[1] is dedicated to the 32bits timer.

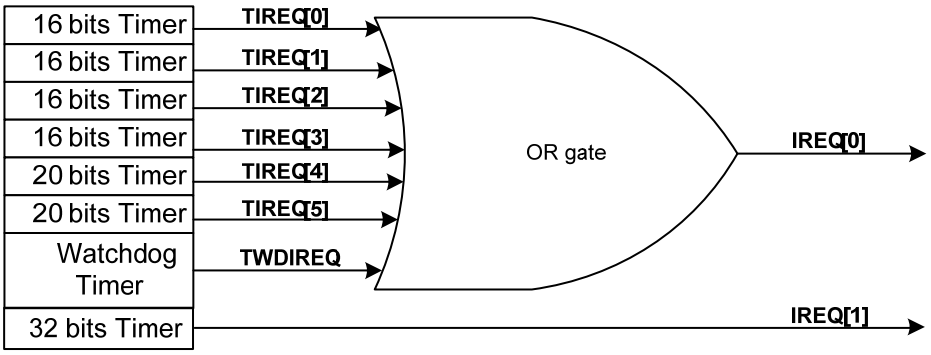


Figure 3.8 IREQ

4 PMU (power management unit)

4.1 Overview

The block diagram of PMU is shown in the following figure.

The PMU operated by XI/XO makes the signals to turn on or to turn-off the sub blocks or overall system.

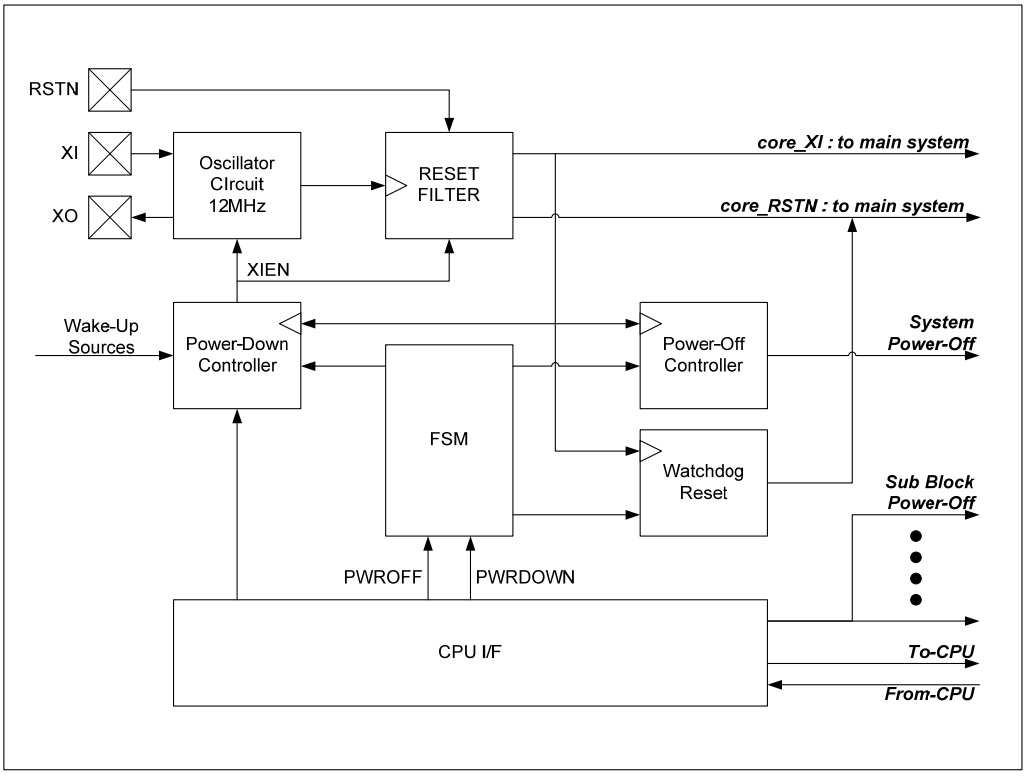


Figure 4.1 PMU Block Diagram

The PMU of the TCC8900 also can support the wake-up sources from external or internal devices. In the power-down mode, if one of the wake-up sources be activated, the Power-Down Controller makes the XIEN high and then the main oscillator starts oscillation. The watchdog reset controller in the PMU can generate reset only.

4.2 Register Overview

Table 4.1 PMU Register Map (Base Address = 0xF0404000)

Name	Address	Type	Reset	Description
CONTROL	0x00	R/W		PMU Control Register
WKUPEN	0x04	R/W		Wakeup Enable Configuration Register
WKUPPOL	0x08	R/W		Wakeup Polarity Configuration Register
WATCHDOG	0x0C	R/W		Watchdog Control Register
CONFIG0	0x10	R/W		Boot Configuration Register
USERSTS	0x14	R/W		Status Register
PWROFF	0x18	R/W		Power-Off Control Register

4.3 Register Descriptions

CONTROL Register **0xF0404000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IOR	FWKU	FPO											AISON	ASTM	APEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
											INITR	DPDN	PDN	POFF	XIEN

Field	Name	RW	Reset	Description
0	XIEN	RW	1	Main Oscillator Enable Register 0 for Disable, 1 for Enable **** Do not write '0'***
1	POFF	RW	0	Power Off Mode Register 0 for Normal, 1 for Power-Off **** Should check the power-up/down sequence*** **** Writing '1' with DPDN '0' makes the power-off state and Writing '1' with DPDN '1' makes the deep-power-down state.***
2	PDN	RW	0	Power Down Mode Register 0 for Normal, 1 for Power-Down **** Refer to enter/exit power-down sequence***
3	DPDN	RW	0	Deep Power Down Mode Register 0 for Normal, 1 for Deep Power-Down **** Refer to enter/exit deep power-down sequence***
4	INITR	RW	0	Boot Memory Configuration Register with ITCM. 0 for BOOT-ROM, 1 for ITCM * When waking up from POWER-OFF, DEEP-POWER-DOWN mode.
16	APEN	RW	0x0	Touch Screen ADC Power Enable Register 0 for Disable, 1 for Enable
17	ASTM	RW	0x0	Touch Screen ADC Stop Mode Register 0 for Normal, 1 for Stop Mode
18	AISON	RW	0x0	Touch Screen ADC Isolation Enable Register 1 for Not-Isolated, 0 for Isolated **** Before entering power-off mode or deep-power-down mode, this should be cleared***.
29	FPO	RW	0x0	Fast Power-Off 0 for Normal, 1 for Fast **** This is only for test mode***
30	FWKU	RW	0x0	Fast Wakeup Enable Register 0 for Normal, 1 for Fast **** This is only for test mode.***
31	IOR	RW	0x0	I/O Retension Enable Register 0 for Not-Retension, 1 for Retension **** Refer to the power-up/down sequence***

WKUPEN Register **0xF0404004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCS															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCS															

Field	Name	RW	Reset	Description
31~0	SRCS	RW	1	Wakeup Enable Register for Each Wakeup Sources SRCS[0] : RTCWKUP[6] SRCS[1] : GPIOC[28] SRCS[2] : GPIOC[29] SRCS[3] : GPIOC[30] SRCS[4] : GPIOC[31] SRCS[5] : GPIOF[27] SRCS[6] : GPIOF[26] SRCS[7] : GPIOF[25] SRCS[8] : GPIOF[24] SRCS[9] : GPIOF[23] SRCS[10] : TSC_WKU[7] SRCS[11] : GPIOD[18] SRCS[12] : TSC_STOP_WKU[8] SRCS[13] : TSC_UPDOWN[9] SRCS[14] : GPIOA[2] SRCS[15] : GPIOA[3] SRCS[16] : GPIOA[4] SRCS[17] : GPIOA[5] SRCS[18] : GPIOA[6] SRCS[19] : GPIOA[7] SRCS[20] : GPIOA[10] SRCS[21] : GPIOA[11] SRCS[22] : GPIOA[12] SRCS[23] : GPIOA[13] SRCS[24] : GPIOA[14] SRCS[25] : GPIOA[15] SRCS[26] : GPIOB[30] SRCS[27] : GPIOB[31] SRCS[28] : GPIOE[04] SRCS[29] : GPIOE[05] SRCS[30] : GPIOE[24] SRCS[31] : GPIOE[25]

[6] RTC Wakeup Output Signal
[7] Touch Screen Wake-Up Signal
[8] Touch Screen Stop Wakeup Signal
[9] Touch Screen Up/Down Signal

WKUPPOL Register **0xF0404008**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCS															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCS															

Field	Name	RW	Reset	Description
31~0	SRCS	RW	1	Wakeup Enable Register for Each Wakeup Sources 0 for Active High, 1 for Active Low SRCS[0] : RTCWKUP[10] SRCS[1] : GPIOC[28] SRCS[2] : GPIOC[29] SRCS[3] : GPIOC[30] SRCS[4] : GPIOC[31] SRCS[5] : GPIOF[27] SRCS[6] : GPIOF[26] SRCS[7] : GPIOF[25] SRCS[8] : GPIOF[24] SRCS[9] : GPIOF[23] SRCS[10] : TSC_WKU[11] SRCS[11] : GPIOD[18] SRCS[12] : TSC_STOP_WKU[12] SRCS[13] : TSC_UPDOWN[13] SRCS[14] : GPIOA[2] SRCS[15] : GPIOA[3] SRCS[16] : GPIOA[4] SRCS[17] : GPIOA[5] SRCS[18] : GPIOA[6] SRCS[19] : GPIOA[7] SRCS[20] : GPIOA[10] SRCS[21] : GPIOA[11] SRCS[22] : GPIOA[12] SRCS[23] : GPIOA[13] SRCS[24] : GPIOA[14] SRCS[25] : GPIOA[15] SRCS[26] : GPIOB[30] SRCS[27] : GPIOB[31] SRCS[28] : GPIOE[04] SRCS[29] : GPIOE[05] SRCS[30] : GPIOE[24] SRCS[31] : GPIOE[25]

[10] RTC Wakeup Output Signal
[11] Touch Screen Wake-Up Signal Can't be used in the Power-Off or Deep-Power-Down Mode.
[12] Touch Screen Stop Wakeup Signal Can't be used in the Power-Off or Deep-Power-Down Mode.
[13] Touch Screen Up/Down Signal

WATCHDOG Register **0xF040400C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN	CLR							STR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT															

Field	Name	RW	Reset	Description
15 ~ 0	CNT	RW	0xffff	Watchdog Count Register The frequency of input clock is XI This is initialized by external RSTN.
23 ~ 16	STR	RW	0x0	User Status Register for Software This is not used by H/W. This is initialized by external RSTN.
30	CLR	RW	0x0	Watchdog Clear Register If write ‘1’, the watchdog counter restarts. This is cleared automatically.
31	EN	RW	0x0	Watchdog Enable Register ‘1’ for Enable, ‘0’ for Disable If watchdog reset activated, this will be cleared.

CONFIG0 Register **0xF0404010**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
							TM		PKG				BM		

Field	Name	RW	Reset	Description
2 ~ 0	BM	RW	-	Boot Mode Port Status **** Don’t write***
6~4	PKG	RW	-	Package Port Status **** Don’t write***
8	TM	RW	0x0	Test Mode Pin **** Don’t write***

USERSTS Register **0xF0404014**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STATUS															

Field	Name	RW	Reset	Description
15 ~ 0	STATUS	RW	0x0	This register is initialized by RSTN port

PWROFF Register **0xF0404018**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						IOB	GB	DB	VB	MB	SP	UP	LV	HD	DAC

Field	Name	RW	Reset	Description
0	DAC	RW	0	Video DAC Power-Off Control Register '1' for Power-Off, '0' for Power-On **** This is only used to isolate the core interface signals. The DAC power should not be turned-off for leakage path between internal core power and DAC power.***
1	HD	RW	0	HDMI Phy Power-Off Control Register '1' for Power-Off, '0' for Power-On **** This is only used to isolate the core interface signals. If you want to reduce power consumption for the HDMI phy, the external HDMI power should be turned-off.***
2	LV	RW	0	LVDS Phy Power-Off Control Register '1' for Power-Off, '0' for Power-On **** This is only used to isolate the core interface signals. If you want to reduce power consumption for the LVDS phy, the external LVDS power should be turned-off.***
3	UP	RW	0	USB Nano Phy Power-Off Control Register '1' for Power-Off, '0' for Power-On **** This is only used to isolate the core interface signals. If you want to reduce power consumption for the USB 2.0 phy, the external USB 2.0 power should be turned-off. The USB2.0 power is composed of PWRUSB12, PWRUSB33.***
4	SP	RW	0	SATA Phy Power-Off Control Register '1' for Power-Off, '0' for Power-On **** It is only used to isolate the core interface signals. If you want to reduce power consumption for the SATA phy, the external SATA power should be turned-off.***
5	MB	RW	0	Memory Bus Power-Off Control Register '1' for Power-Off, '0' for Power-On
6	VB	RW	0	Video Bus Power-Off Control Register '1' for Power-Off, '0' for Power-On **** It is only used to turn-off the core power for the video bus.***
7	DB	RW	0	DDI Bus Power-Off Control Register '1' for Power-Off, '0' for Power-On **** It is only used to turn-off the core power for the display device bus.***
8	GB	RW	0	Graphic Bus Power-Off Control Register '1' for Power-Off, '0' for Power-On **** It is only used to turn-off the core power for the graphic bus.***
9	IOB	RW	0	I/O Bus Power-Off Control Register '1' for Power-Off, '0' for Power-On **** This is only used to turn-off the core power for the memories in the I/O bus.***

4.4 Operation & Timing Diagram

4.4.1 Power Management Scheme

The following figure shows the overall block diagram of the TCC8900.

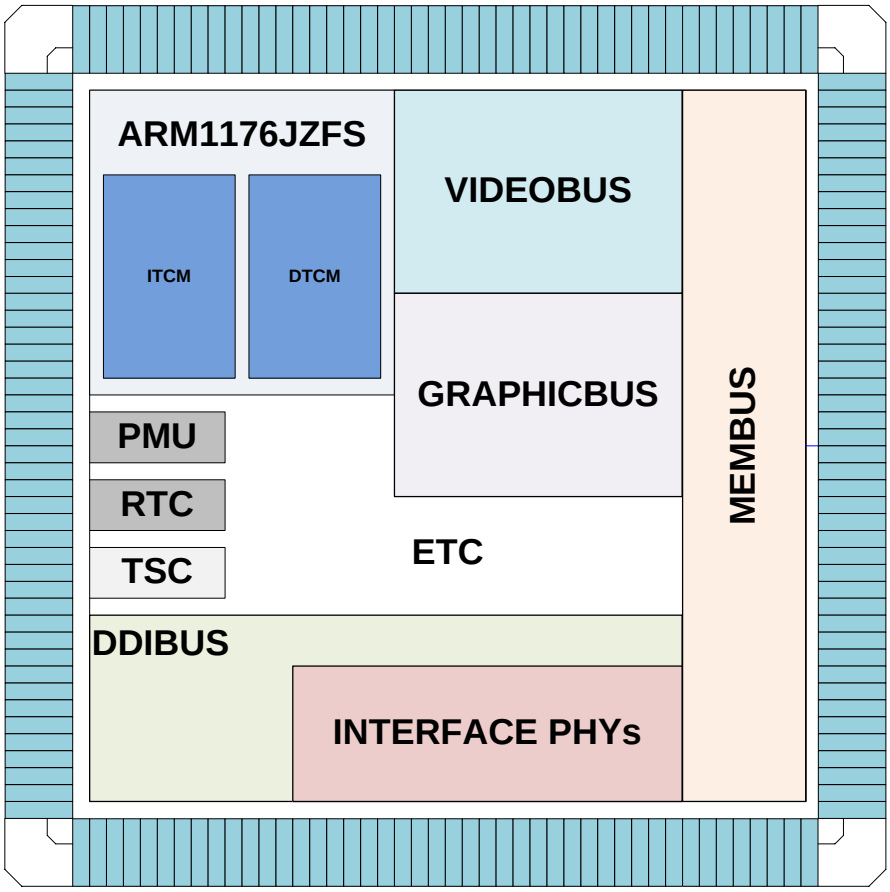


Figure 4.2 Overall Block Diagram

The TCC8900 has the 10 component group for power management. The PMU can control the power state of each block. But, the RTC block is always alive if PWRRTC is supplied. The high speed serial interface phys are belongs to “INTERFACE PHYs” group.

The ITCM & DTCM can retend the data in the corresponding mode described in the next section. If the self-refresh mode is required, the MEMBUS should be alive in the corresponding mode.

If the ARM1176JZF-S has been in power-off state, the boot-up procedure should be executed after waking up.

Basically, the GPIOs in the TCC8900 can keep the state such as input or output direction and output value in the output mode. It means that the register values for GPIOs before turning-off the internal power should reside in the TCM memories or external SDRAM to restore the state of the GPIOs and the memories are alive during the power off state.

4.4.2 Operating Mode Definitions

The state diagram for power down modes is shown in the following figure.

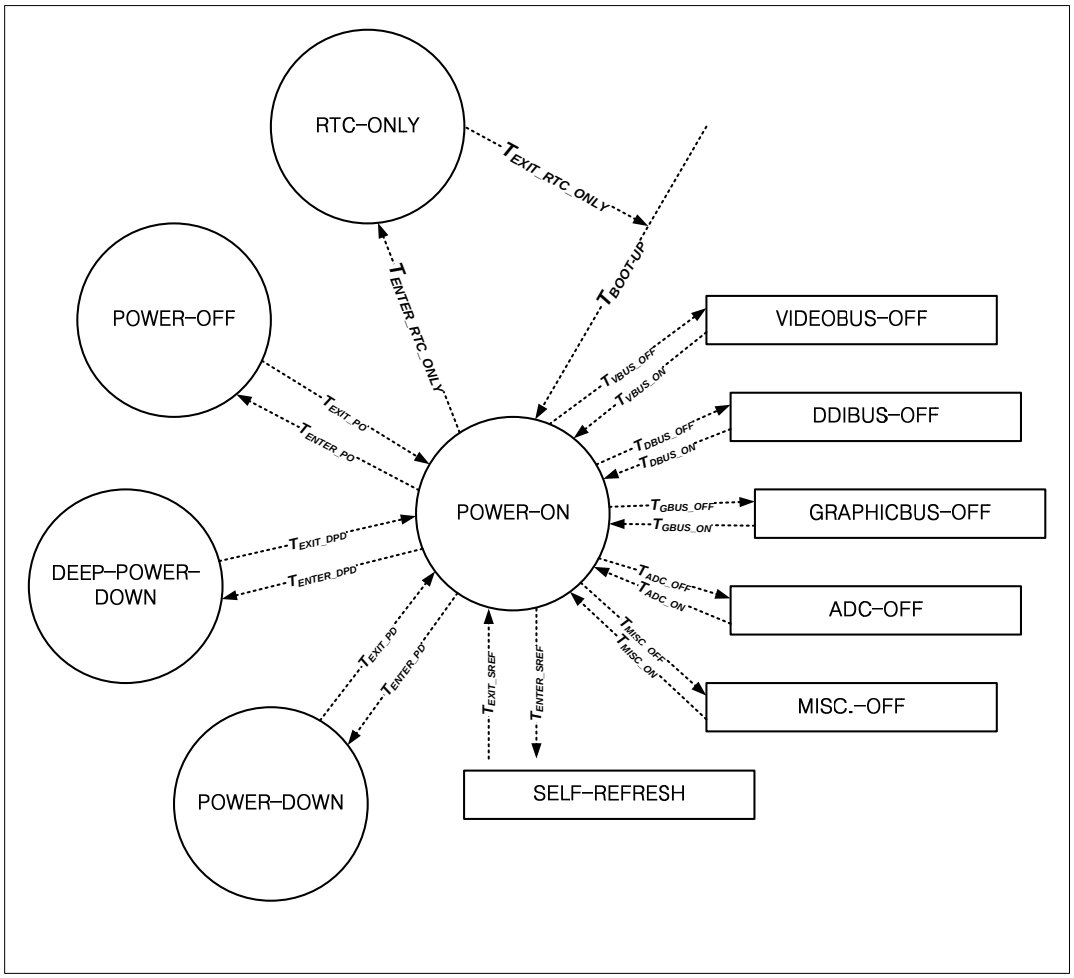


Figure 4.3 Operating Modes

Conceptually, the states described in circle in the above figure are the power-down mode defined in the TCC8900. And the rectangular block can be turned-on or turned-off freely in the “POWER-ON” state which means that the ARM1176JZF-S is in working. All the power-down modes except for the “POWER-ON” makes the ARM1176JZF-S turned-off.

RTC-ONLY mode

This mode is defined that all the powers except for the RTC are turned-off which means the power consumption is lowest. The only way to wake up is to be turned on by the power circuit through external events such as key or the internal “PMWKUP”. Before waking up, the RSTN should have “LOW” to “HIGH” transition. And then the main boot procedure can start processing waking up.

POWER-OFF mode

This mode is defined that internal core power and I/O power are alive and the RSTN should be “HIGH”. In this mode, the GPIO states can be retended by software. The main oscillator doesn’t work any more and can restart oscillation by the wake-up events which can be activated externally or internally. The touch screen event is belongs to the internally generated event such as UP/DOWN event described in the TSC(Touch Screen Controller). Also, the power of ITCM and DTCM is turned off which means that ITCM and DTCM can not preserve the data any more.

DEEP-POWER-DOWN mode

This mode is very similar to the “POWER-OFF” mode. The major difference is the power state of the ITCM and DTCM. In this mode, the data stored in the ITCM and DTCM can be preserved and can be used to restore the states of the whole system after waking up. Also the main oscillator does not work any more and can be waked up by the external wake-up events.
For ITCM being alive during the DEEP-POWER-DOWN mode, the ITCM can be used as the wake-up boot memory to make

INITR field “1” before entering the DEEP-POWER-DOWN mode.

POWER-DOWN mode

In this mode, all the powers except for the rectangular blocks in the above figure are alive. Also, the main oscillator can be in not-working state and the wake-up procedure is same as one of the “DEEP-POWER-DOWN” mode.

POWER-ON mode

Finally, the “POWER-ON” state is defined that the main oscillator is in working. Also, the rectangular boxes can be turned on and off.

Modes	Ext. I/O Power	Ext. Core Power	Inside Core Power	Int. I/O Retension	ENTER /WAKEUP	ALIVE BLOCKS
RTC_ONLY	OFF	OFF	OFF	X	GPIO/ PMWKUP	RTC
POWER-OFF[14]	ON	ON	OFF	O/X[15]	CPU/ WAKEUP	RTC, PMU[16]
DEEP-POWER-DOWN[14]	ON	ON	OFF (Partial)	O/X[15]	CPU/ WAKEUP	RTC, PMU, TCM[16]
POWER-DOWN[14]	ON	ON	OFF (Partial)	O/X[15]	CPU/ WAKEUP	RTC, PMU, ARM w/ TCMs
POWER-ON	ON	ON	ON (Partial)[17]	O/X[15]	-	-

[14] The RSTN port should be HIGH.
[15] The I/O retension function can be controlled by CPU(Program) before entering the corresponding mode.
[16] The touch screen controller(ADC) can be lived.
[17] The sub block can be in power-on or off state

4.4.3 Basic Power-Up Sequence

An example of the timing diagram for power-up sequence is shown below, Figure 1.2.

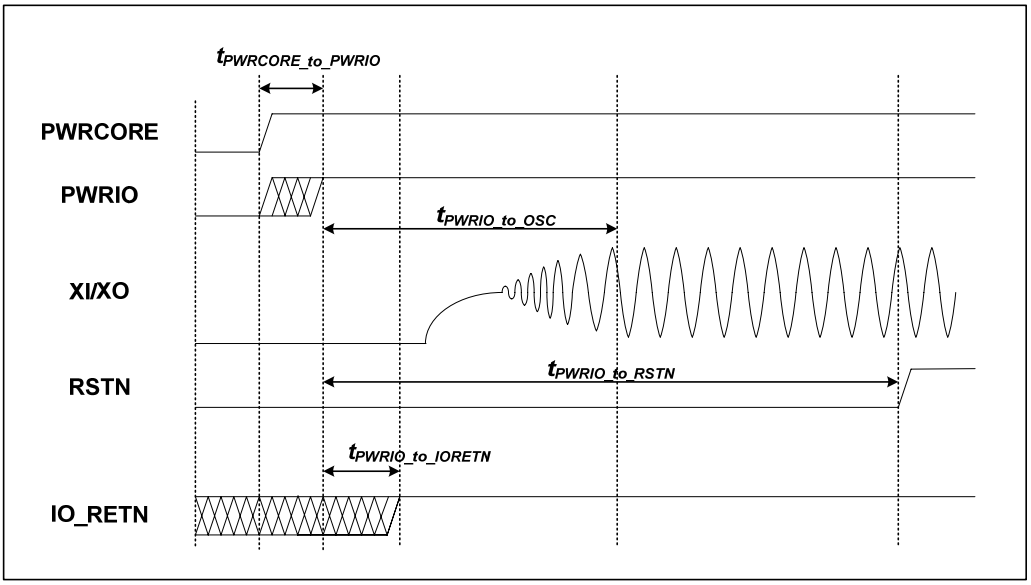


Figure 4.4 Power-Up Sequence

- $t_{PWRCORE\_to\_PWRIO}$: PWRIO turn-on time after PWRCORE asserted.
  0 < $t_{PWRCORE\_TO\_PWRIO}$
- $t_{PWRIO\_to\_OSC}$: The main oscillator stable time after PWRIO(OSC) asserted.
  $t_{PWRIO\_TO\_OSC}$ < 10ms
- $t_{PWRIO\_to\_RSTN}$: The RSTN time after PWRIO asserted.
  20ms < $t_{PWRIO\_TO\_RSTN}$
- $t_{PWRIO\_to\_IORETN}$: The I/O release time after PWRIO asserted when RSTN LOW.
  $t_{PWRIO\_TO\_IORETN}$ < 1us
- ***Power State*** : HIGH : VDD * 0.9, LOW : VDD * 0.1

The function of I/O retension is defined that the I/O direction control signal and output level were kept during the PWRCORE off. In the TCC8900, the internal core power can be controlled by internal power management circuit.

4.4.4 Basic Power-Off Sequence – Turn-Off

The sequence in this section is invalid for the pre-defined power-down modes such as POWER-OFF, DEEP-POWER-DOWN, POWER-DOWN, and etc. An example of the timing diagram for power-off sequence is shown below, Figure 1.2.

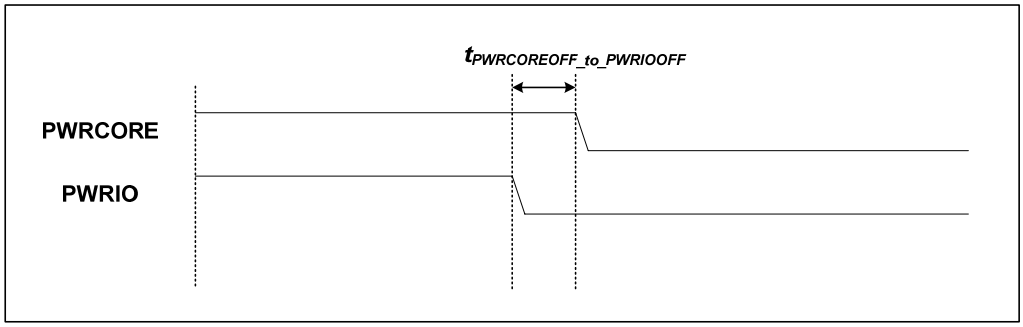


Figure 4.5 Power-Off Sequence

- $t_{PWRCOREOFF\_to\_PWRIOOFF}$: PWRCORE turn-off time after PWRIO de-asserted.
  $0 < t_{PWRCOREOFF\_TO\_PWRIOOFF}$
- ***Power State*** : HIGH : VDD * 0.9, LOW : VDD * 0.1

4.4.5 Enter POWER-ON State from Initial Boot-Up or RTC PMWKUP

If there is rising transition on RSTN port, the TCC8900 starts to initialize the PMU and enter the POWER-ON state. In this section, the timing diagram from the RSTN rising transition to POWER-ON state will be described.

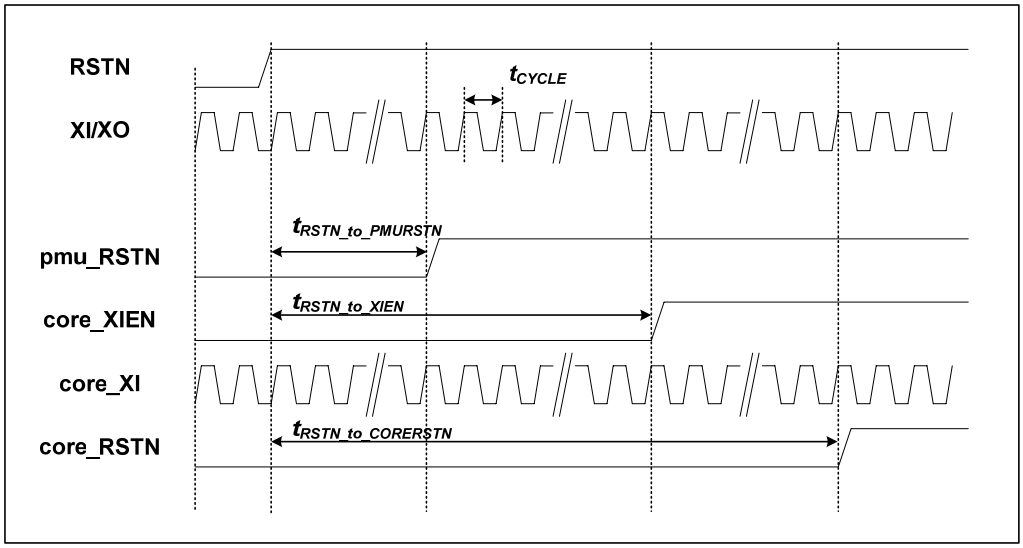


Figure 4.6 Timing Diagram for Entering POWER-ON

- t_{CYCLE} : Clock cycle period, about 84ns for 12MHz.
- $t_{RSTN\_to\_PMURSTN}$: The cycle period from external RSTN to internal RSTN for PMU. $t_{RSTN\_to\_PMURSTN}$ is about 11ms.
- $T_{RSTN\_to\_XIEN}$: The XI/XO stable indicator. $T_{RSTN\_to\_XIE}$ is about 12ms.
- $T_{RSTN\_to\_CORERSTN}$: The cycle period from external RSTN to internal RSTN for core. $T_{RSTN\_to\_XIE}$ is about 16ms.

4.4.6 Enter POWER-OFF and DEEP-POWER-DOWN Mode

In the POWER-OFF mode, the XI/XO oscillator would be disabled, which requires the following restrictions before entering this mode.

- All the clocks using PLL should be changed to XI or disabled.
- The wakeup event should be assigned to the corresponding wakeup source.
- The I/O retension function should be enabled.

The POWER-OFF mode can be entered by writing '1' to POFF in the CONTROL register. After that, the POWER-OFF sequence is as following figure.

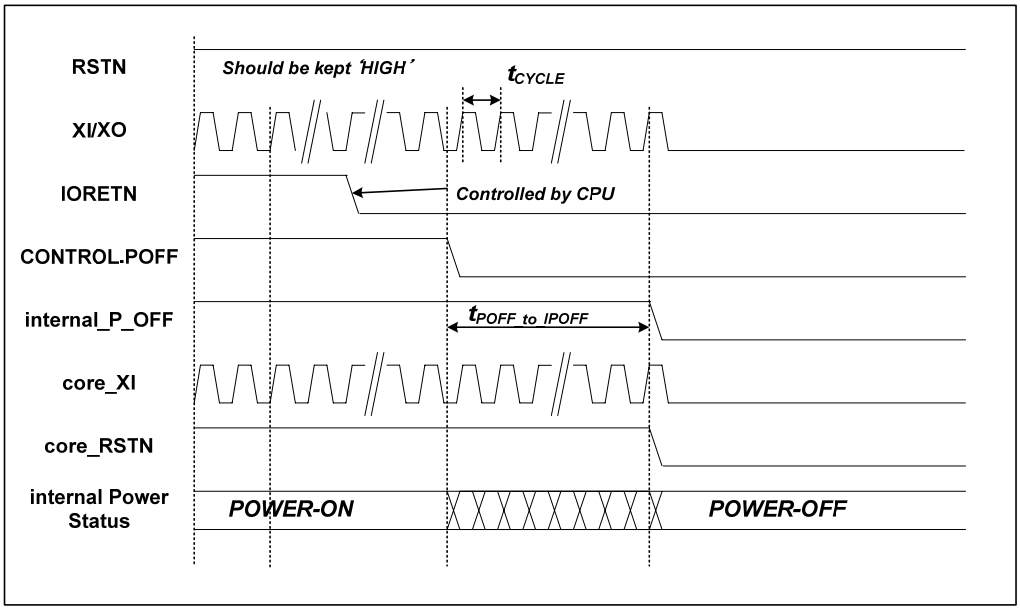


Figure 4.7 Timing Diagram for Entering POWER-OFF and DEEP-POWER-DOWN Mode

- t_{CYCLE} : Clock cycle period, about 84ns for 12MHz.
- $t_{POFF\_to\_IPOFF}$: The cycle period from POFF(register) to internal power-off signal.
  $t_{POFF\_to\_IPOFF}$ < 1ms

4.4.7 WAKE-UP Event Configuration

The WAKE-UP event can be made by the various sources which are defined in the register description. The following figure shows how to make WAKE-UP event with polarity control register and enable register.

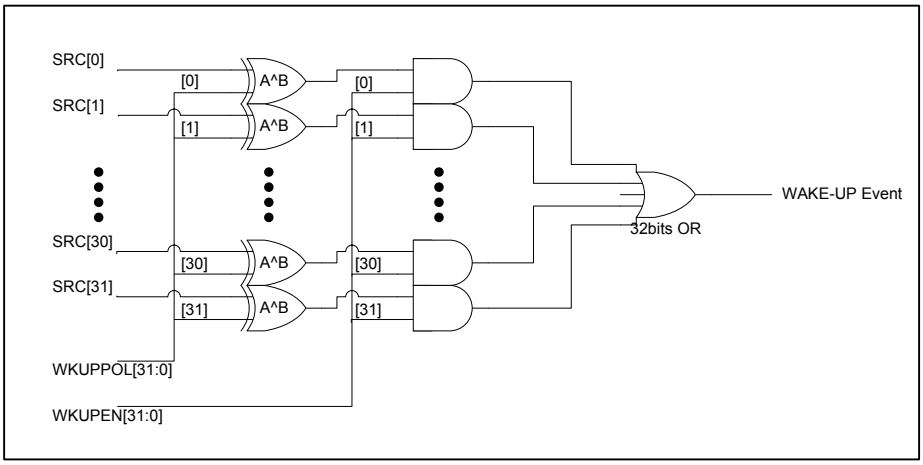


Figure 4.8 Timing Diagram for Exiting POWER-OFF and DEEP-POWER-DOWN Mode

4.4.8 Exit POWER-OFF and DEEP-POWER-DOWN Mode

To wake-up from the POWER-OFF mode, the wake-up sources should be configured.

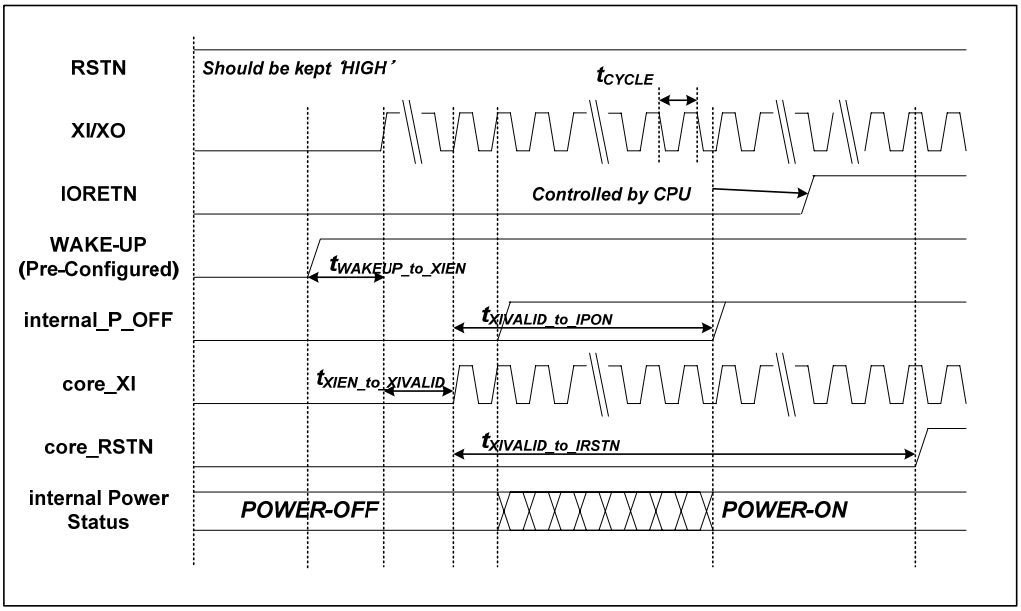


Figure 4.9 Timing Diagram for Exiting POWER-OFF and DEEP-POWER-DOWN Mode

- t_{CYCLE} : Clock cycle period, about 84ns for 12MHz.
- $t_{WAKEUP\_to\_XIEN}$: The maximum delay from WAKEUP to XIEN.
  $t_{WAKEUP\_to\_XIEN}$ < 1us
- $T_{XIEN\_to\_XIVALID}$: The maximum delay from XIEN to valid clock output.
  $t_{XIEN\_to\_XIVALID}$ is about 11ms.
- $t_{XIVALID\_to\_IPON}$: The clock cycles from XIVALID to internal Power-On.
  $t_{XIVALID\_to\_IPON}$ is about 1ms.
- $t_{XIVALID\_to\_IRSTN}$: The clock cycles from XIVALID to internal reset.
  $t_{WAKEUP\_to\_XIEN}$ is about 16ms.

4.4.9 Enter/Exit POWER-DOWN Mode

The timing sequence is very similar to the DEEP-POWER-DOWN Mode, but the major difference is the state of the core_RSTN.

In the POWER-DOWN mode, the core_RSTN always keeps 'HIGH and the processor can continue to work.

4.4.10 Enter and Exit DEEP-POWER-DOWN Mode with ITCM Booting

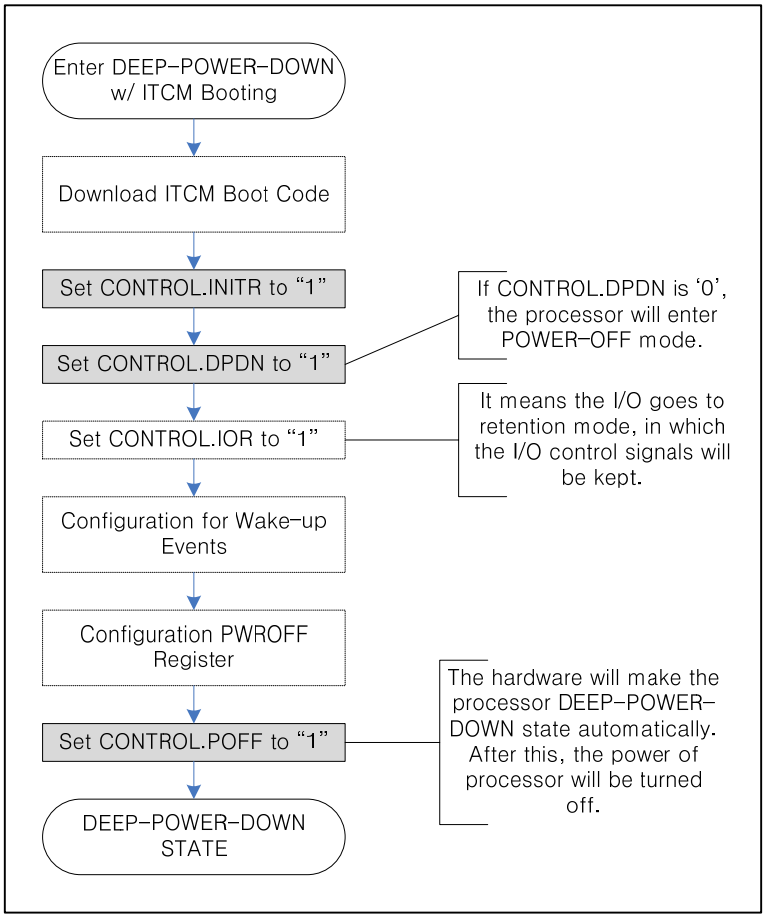


Figure 4.10 Example Flow Chart to Enter Deep-Power-Down Mode

The above figure shows a example flow chart to enter deep-power-down mode. To wake up with ITCM boot mode, the gray colored boxes are important, the "CONTROL.INITR" register field makes the ARM cpu to boot with ITCM memory which is guided in the ARM technical reference manual. And when the PMU enters power-off mode with "CONTROL.POFF", it scans the "CONTROL.DPDN" signal. If the "CONTROL.DPDN" is high, the PMU will enter the DEEP-POWER-DOWN mode, otherwise POWER-OFF mode. Just after "CONTROL.POFF" register being '1', the main oscillator will be disabled and the power of the ARM and other peripherals will be turned off consequently.

The "CONTROL.IOR" signal makes the I/O to retend their status including direction, driver strengths, pull-up/down control signals and during the "CONTROL.IOR" being '1', the I/O can't be controlled by internal signals.

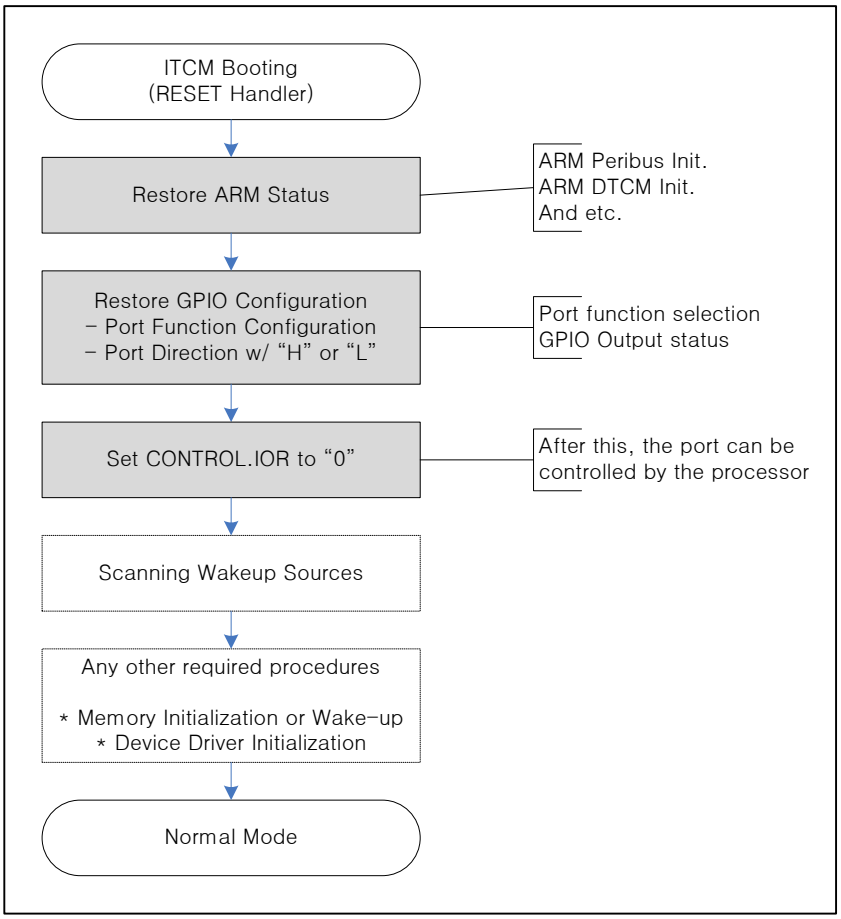


Figure 4.11 Example Flow Chart to Exit Deep-Power-Down Mode

The above figure shows a example flow chart to exit deep-power-down mode. When the wake-up events occurred, the PMU makes clock enable and de-assert the RESET to internal core and ARM cpu and then the ARM can start booting procedure. At this time, if the "CONTROL.INITR" is high, the ARM cpu will fetch the instruction with ITCM. In the case of "CONTROL.INITR" being high, the code for the above flow chart should be on the ITCM memory.

After ARM booted, the ARM status should be restored by software such as DTCM initialization and peripheral configuration and etc.

And the port status should be restored before the "CONTROL.IOR" being '0' for that the I/O status may be in the unstable(unknown) status which can make the glitch on the external port.

4.4.11 Power On/Off Sequence for the Video Bus

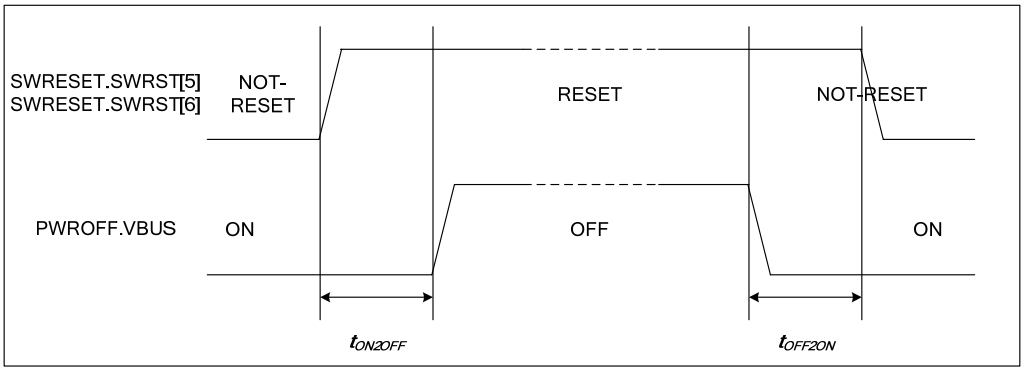


Figure 4.12 Power-On/Off Sequency for the Video Bus

The Video Bus inside the processor can be turned off to reducing the power consumption. The above figure shows the power on/off sequence controlled by the software. The SWRESET.SWRST[5] and [6] are described in the CKC.
Before entering the power-off state or exiting the power-off state of the video bus, the video bus should be in "RESET-STATE". If the video bus is in the NOT-RESET state during the power-off state, the system can be disturbed by signals from the video bus.

The "t_{ON2OFF}" time should be greater than the 100us and the "t_{OFF2ON}" time should be greater than 100ms.

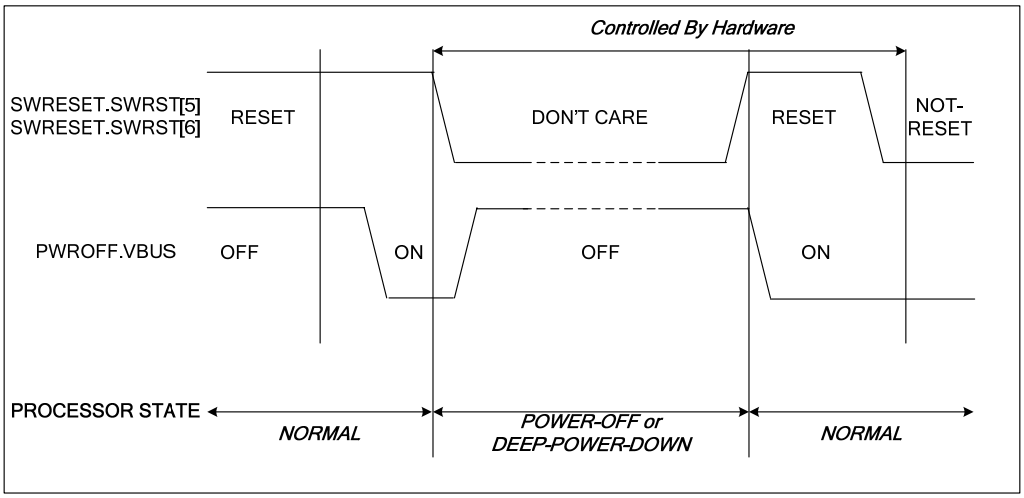


Figure 4.13 Video Bus Power Management for Entering/Exiting the POWER-OFF or DEEP-POWER-DOWN

The above figure shows the power management sequence for entering or exiting the POWER-OFF or DEEP-POWER-DOWN state.
Before entering the POWER-OFF or DEEP-POWER-DOWN state, the video bus should be in the "ON" state to prevent abnormal wake-up by the video bus. The period noted by "Controlled By Hardware" shows the internal transition sequence of the video bus.

4.4.12 Power On/Off Sequence for the Graphic Bus

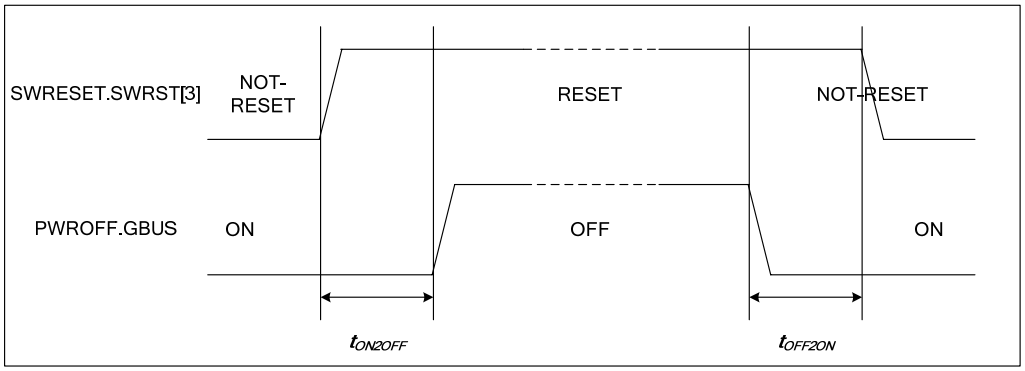


Figure 4.14 Power-On/Off Sequency for the Graphic Bus

The Graphic Bus inside the processor can be turned off to reducing the power consumption. The above figure shows the power on/off sequence controlled by the software. The SWRESET.SWRST[3] is described in the CKC.
Before entering the power-off state or exiting the power-off state of the graphic bus, the graphic bus should be in "RESET-STATE". If the video bus is in the NOT-RESET state during the power-off state, the system can be disturbed by signals from the graphic bus.

The "t_{ON2OFF}" time should be greater than the 100us and the "t_{OFF2ON}" time should be greater than 100ms.

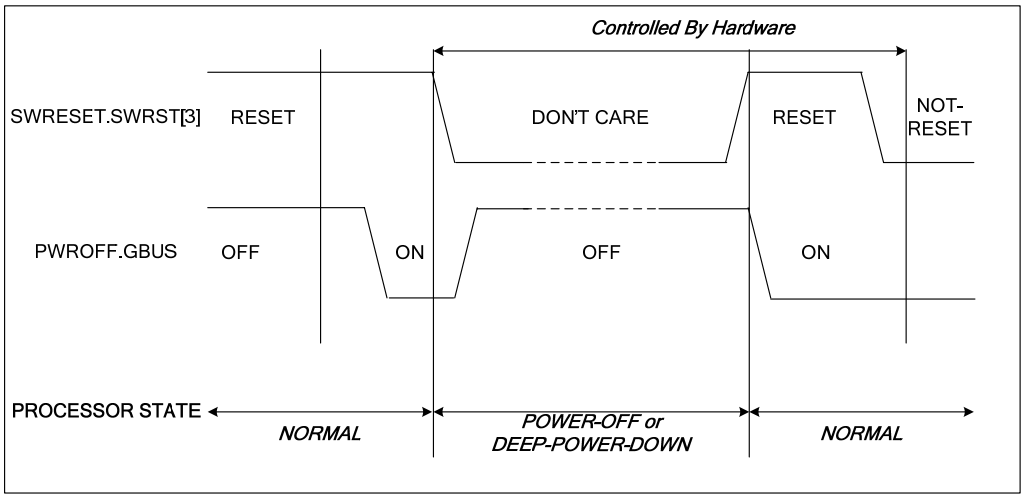


Figure 4.15 Graphic Bus Power Management for Entering/Exiting the POWER-OFF or DEEP-POWER-DOWN

The above figure shows the power management sequence for entering or exiting the POWER-OFF or DEEP-POWER-DOWN state.
Before entering the POWER-OFF or DEEP-POWER-DOWN state, the graphic bus should be in the "ON" state to prevent abnormal wake-up by the graphic bus. The period noted by "Controlled By Hardware" shows the internal transition sequence of the graphic bus.

4.4.13 Power On/Off Sequence for the DDI Bus

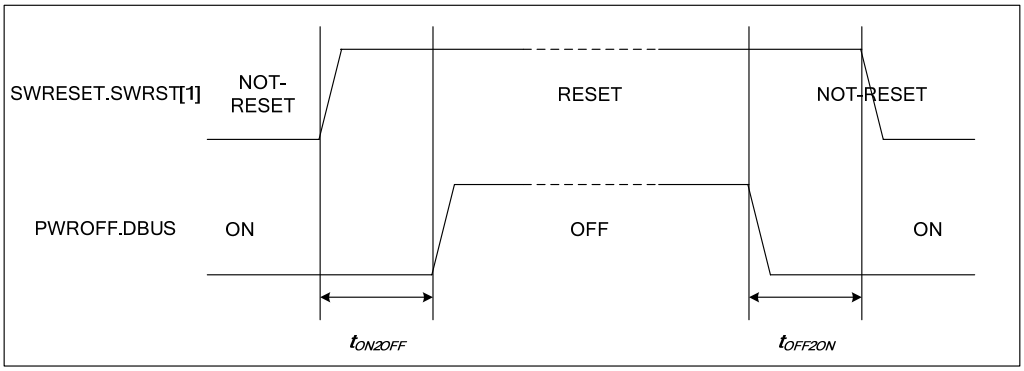


Figure 4.16 Power-On/Off Sequency for the DDI Bus

The DDI Bus inside the processor can be turned off to reducing the power consumption. The above figure shows the power on/off sequence controlled by the software. The SWRESET.SWRST[1] is described in the CKC.
Before entering the power-off state or exiting the power-off state of the DDI bus, the DDI bus should be in “RESET-STATE”. If the DDI bus is in the NOT-RESET state during the power-off state, the system can be disturbed by signals from the DDI bus.

The “t_{ON2OFF}” time should be greater than the 100us and the “t_{OFF2ON}” time should be greater than 100ms.

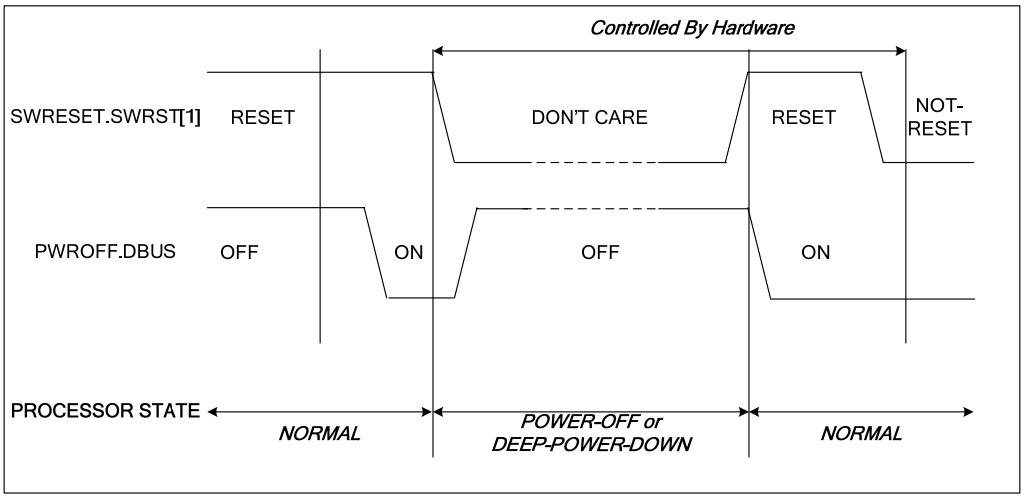


Figure 4.17 DDI Bus Power Management for Entering/Exiting the POWER-OFF or DEEP-POWER-DOWN

The above figure shows the power management sequence for entering or exiting the POWER-OFF or DEEP-POWER-DOWN state.
Before entering the POWER-OFF or DEEP-POWER-DOWN state, the DDI bus should be in the “ON” state to prevent abnormal wake-up by the DDI bus. The period noted by “Controlled By Hardware” shows the internal transition sequence of the DDI bus.

4.4.14 Power On/Off Sequence for the HDMI PHY

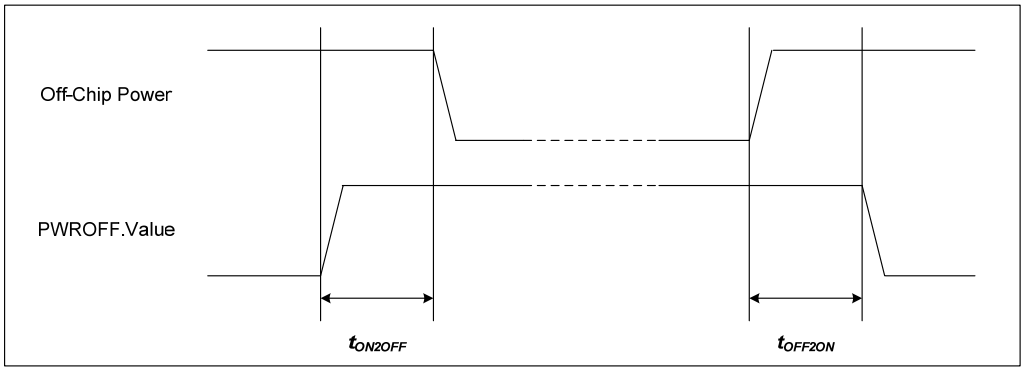


Figure 4.18 Power-On/Off Sequency for the HDMI PHY

The above figure shows the power on/off sequence for the HDMI phy. The “Off-Chip Power” means the power group for the HDMI phy (PWRHDMI, PWRHDMIOSC, PWRHDMIPLL).

Before turning off the power of the “Off-Chip Power”, the “PWROFF.HD” field should be high and should be low after turning on the power of the “Off-Chip Power”.

The “t_{ON2OFF}” and “t_{OFF2ON}” fields should be greater than 100us. And the “Off-Chip Power” can be controlled by the GPIO and by the external device.

4.4.15 Power On/Off Sequence for the SATA PHY

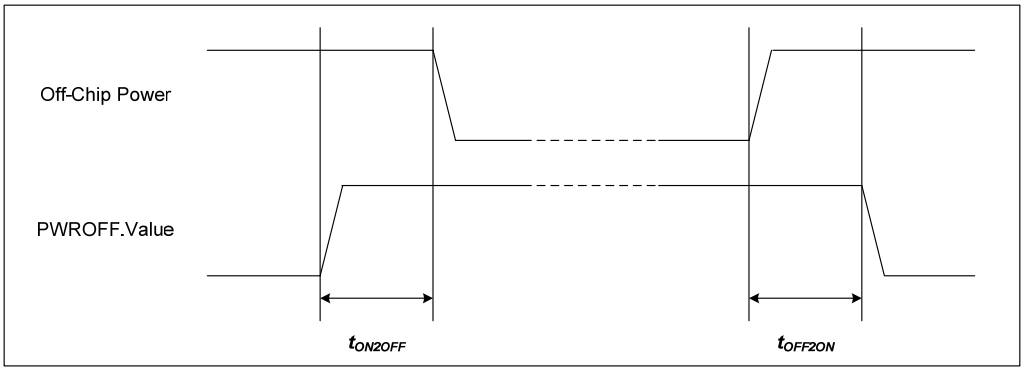


Figure 4.19 Power-On/Off Sequency for the SATA PHY

The above figure shows the power on/off sequence for the SATA phy. The “Off-Chip Power” means the power group for the SATA phy (PWRSATA, PWRSATAOSC, PWRSATAPLL).

Before turning off the power of the “Off-Chip Power”, the “PWROFF.SP” field should be high and should be low after turning on the power of the “Off-Chip Power”.

The “t_{ON2OFF}” and “t_{OFF2ON}” fields should be greater than 100us. And the “Off-Chip Power” can be controlled by the GPIO and by the external device.

4.4.16 Power On/Off Sequence for the USB PHY

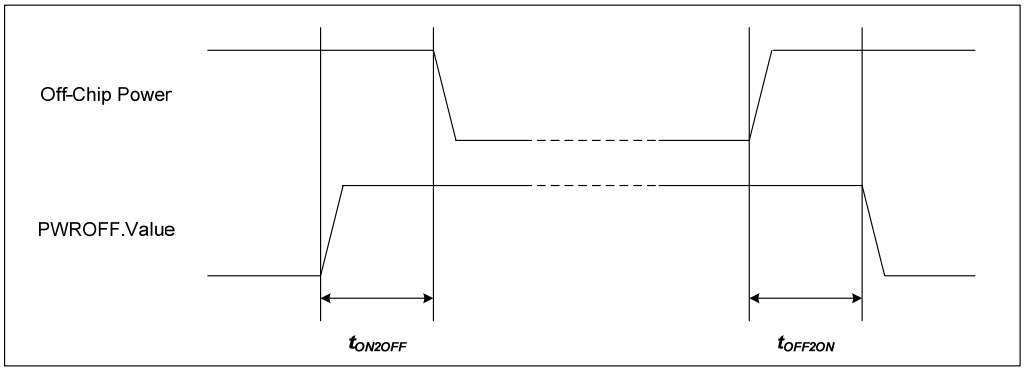


Figure 4.20 Power-On/Off Sequency for the USB PHY

The above figure shows the power on/off sequence for the USB phy. The “Off-Chip Power” means the power group for the USB phy (PWRUSB12, PWRUSB).

Before turning off the power of the “Off-Chip Power”, the “PWROFF.UP” field should be high and should be low after turning on the power of the “Off-Chip Power”.

The “t_{ON2OFF}” and “t_{OFF2ON}” fields should be greater than 100us. And the “Off-Chip Power” can be controlled by the GPIO and by the external device.

4.4.17 Power On/Off Sequence for the LVDS PHY

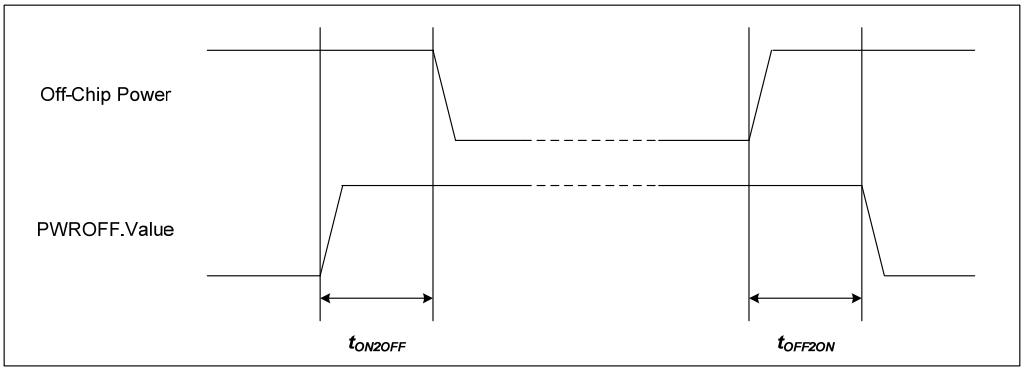


Figure 4.21 Power-On/Off Sequency for the LVDS PHY

The above figure shows the power on/off sequence for the LVDS phy. The “Off-Chip Power” means the power group for the LVDS phy (PWRLVDS).

Before turning off the power of the “Off-Chip Power”, the “PWROFF.UP” field should be high and should be low after turning on the power of the “Off-Chip Power”.

The “t_{ON2OFF}” and “t_{OFF2ON}” fields should be greater than 100us. And the “Off-Chip Power” can be controlled by the GPIO and by the external device.

5 SMU_I2C

5.1 Overview

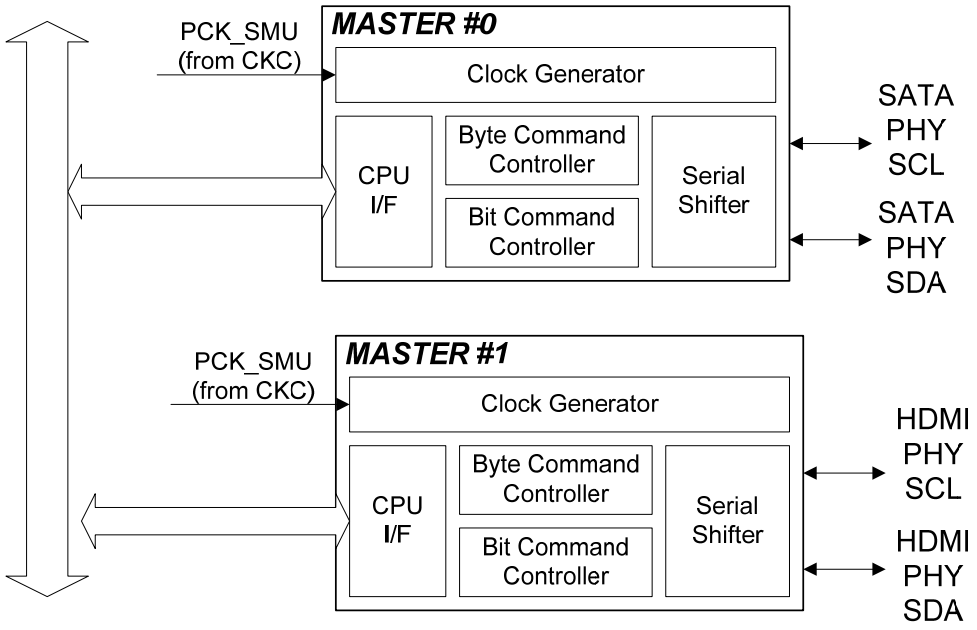


Figure 5.1 I2C Block Diagram

The SMU_I2C controller in the TCC8900 has two masters controller. The each master can control the dedicated I2C device, master 0 for SATA PHY and master 1 for HDMI PHY.

5.2 Register Descriptions

Table 5.1 SMU_I2C Register Map (Base Address = 0xF0405000)

Ch	Name	Addr. Offset	Type	Reset	Description
Master 0	PRES	0x00	R/W	0xFFFF	Clock Prescale register
	CTRL	0x04	R/W	0x0000	Control Register
	TXR	0x08	W	0x0000	Transmit Register
	CMD	0x0C	W	0x0000	Command Register
	RXR	0x10	R	0x0000	Receive Register
	SR	0x14	R	0x0000	Status Register
	TIME	0x18	R/W	0x0000	Timing Control Register
Master 1	PRES	0x40	R/W	0xFFFF	Clock Prescale register
	CTRL	0x44	R/W	0x0000	Control Register
	TXR	0x48	W	0x0000	Transmit Register
	CMD	0x4C	W	0x0000	Command Register
	RXR	0x50	R	0x0000	Receive Register
	SR	0x54	R	0x0000	Status Register
	TIME	0x58	R/W	0x0000	Timing Control Register
Common	ICLK	0x80	R/W	0x0010	I2C_SCL divider Register

Prescale Register (PRES) **0xF0405000, 0xF0405040**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Clock Prescale data															

This register is used to prescale the SCL clock line. Due to the structure of the I2C interface, the core uses a 5*SCL clock internally. The prescale register must be programmed to this 5*SCL frequency (minus 1). Change the value of the prescale register only when 'EN' bit is cleared.

Example :
CLK Input frequency = 8MHz , Desired SCL frequency = 100KHz
Prescale = (8MHz / 100KHz) – 1 = 15

Control Register (CTR) **0xF0405004, 0xF0405044**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								EN	IEN	MOD	RESERVED				

EN [7]	I2C Core enable bit
0	Disabled
1	Enabled

IEN [6]	I2C Core interrupt enable bit
0	Disabled
1	Enabled

MOD [5]	I2C Data Width
0	8bit Mode
1	16bit Mode

Transmit Register (TXR) **0xF0405008, 0xF0405048**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Transmit Data															

When CTRL[5] is set, in case of 16Bit Mode is selected, Transmit Data bit width become 16 bit. Default mode is 8bit mode.

Command Register (CMD) 0xF040500C, 0xF040504C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								STA	STO	RD	WR	ACK	RESERVE		IACK

STA [7]	Start Condition Generation
0	Disabled
1	Enabled

STO [6]	Stop Condition Generation.
0	Disabled
1	Enabled

RD [5]	Read From Slave
0	Disabled
1	Enabled

WR [4]	Write to Slave
0	Disabled
1	Enabled

ACK [3]	Sent ACK
0	Enabled
1	Disabled

IACK [0]	Interrupt Acknowledge
0	-
1	Clear a pending interrupt

Receive Register (RXR) 0xF0405010, 0xF0405050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Receive Data															

When CTRL[5] is set, in case of 16Bit Mode is selected, Transmit Data bit width become 16 bit. Default mode is 8bit mode.

Status Register (SR) 0xF0405014, 0xF0405054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								RxACK	BUSY	AL				TIP	IF

RxACK [7]	Received acknowledge from slave
0	Acknowledge received
1	No Acknowledge received

BUSY [6]	I2C Bus Busy
0	'0' after STOP signal detected
1	'1' after START signal detected

AL[5]	Arbitration lost
0	The core does not lose arbitration
1	The core loses arbitration

Arbitration is lost when a STOP signal is detected, but non requested master drives SDA high, but SDA is low

TIP [1]	Transfer in progress
0	Transfer Complete
1	Transferring Data

IF [0]	Interrupt Flag
0	-
1	Interrupt is pending

Timing Register (TR) **0xF0405018, 0xF0405058**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RC								-		CKSEL	FC				

RC [15:8]	Recovery time counter load value
value	"0" disables recovery time counter. The recovery time counter is enabled and loaded with RC[7:0] whenever a STOP condition is issued by the core. Execution of a new command written to CMD register is delayed until the counter is expired. Actual wait time = (I2CCLK period) * PRES[15:0] * 5 * RC[7:0]

CKSEL [5]	Clock Source Select
0	I2CCLK from Clock controller
1	PCLK (HCLK) divided by 2

Recommended if PCLK is not variable during system operation.

FC[4:0]	Noise filter counter load value
value	"0" disables noise filter. SCL and SDA inputs are checked for stability until the counter is expired.

2C_SCL divider Register (ICLK) **0xF0405080**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT_EN	0														
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIVIDER															

CNT_EN	I2C_SCL counter enable
	I2C clock divider counter enable to generate I2C_SCL

DIVIDER[15:0]	I2C_SCL divider
	I2C clock divider value to generate I2C_SCL

6 Boot Procedure

6.1 Power Up/Down Sequence for Core and I/O Power

The recommended power-up sequence for core and I/O power is described in the following figure.

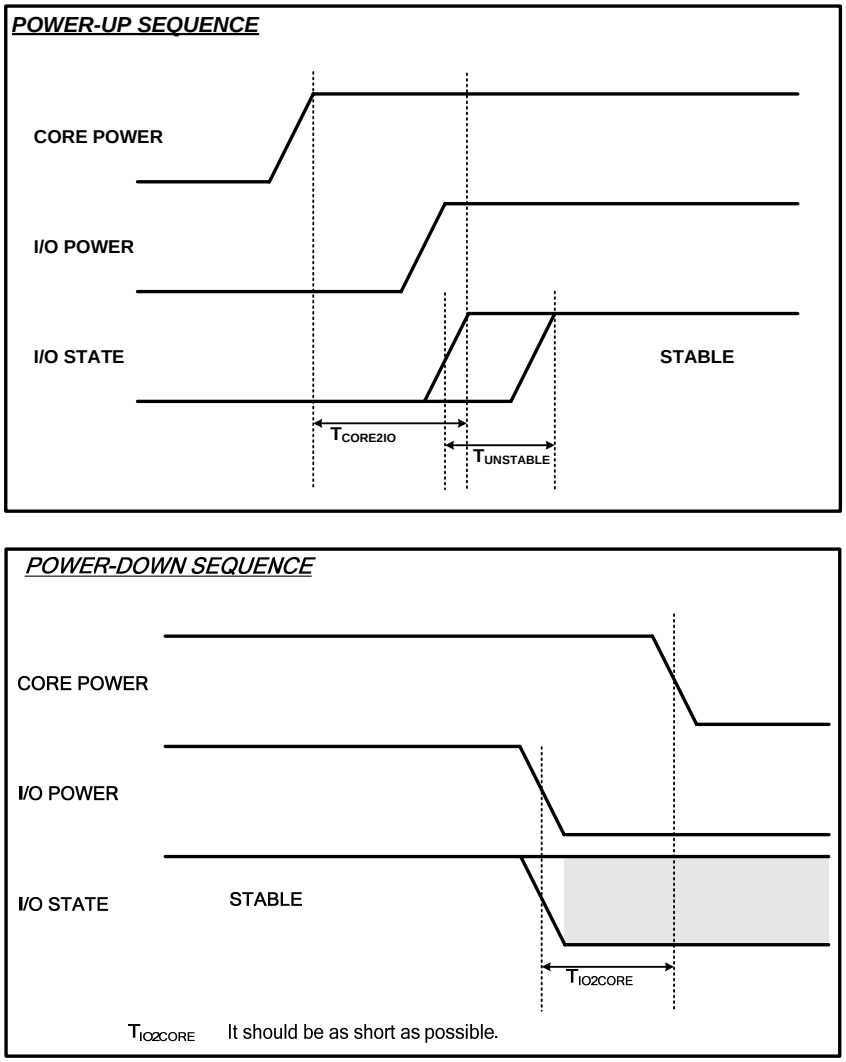


Figure 6.1 Power Up/Down Sequence for Core and I/O Power

Timing parameters in the above figure are as follows.

Name	Description	Min	Typ	Max
$T_{CORE2IO}$	It should be short as possible, namely the value should be greater or equal to 0ns.	0ns		
$T_{UNSTABLE}$	It is dependent on the application circuit. It can cause the meta-stability to CMOS I/O and GPIO connections. Sufficient power supply can make it shorter.	8ns		

* The minimum value of the TUNSTABLE is from the result of simulation, in case the supplied I/O power is ideal at 1.65 volt.

6.2 Boot Mode

The TCC8900 has the 5 pins for booting configuration with BM[2:0], GPIOE[4], GPIOE[0].

Table 6.1: Configuration Value

BM[2:0]	DESCRIPTION	COMMENTS
000b	EHI 80 Boot Mode	f_{CPU} : 240 MHz f_{IOBUS} : 120 MHz f_{EHI} : 120 MHz
001b	I2C Master Boot (EEPROM Boot) Mode {GPIOE[4]} = 0 : I2C Channel 0 Boot {GPIOE[4]} = 1 : I2C Channel 1 Boot {GPIOE[0]} = 0 : If I2C boot failed, go to UART channel 0 boot {GPIOE[0]} = 1 : If I2C boot failed, go to UART channel 1 boot	f_{CPU} : 240 MHz f_{IOBUS} : 120 MHz f_{I2C} : 3 MHz f_{SCK} : 300 KHz
010b	Serial Flash Boot (Serial EEPROM Boot) Mode {GPIOE[4]} = 0 : If boot failed, go to UART boot. {GPIOE[4]} = 1 : If boot failed, go to USB boot. {GPIOE[0]} = 0 : In UART boot, the channel is 0. {GPIOE[0]} = 1 : In UART boot, the channel is 1.	f_{CPU} : 240 MHz f_{IOBUS} : 120 MHz f_{SF} : 80 MHz
011b	USB Function Boot (USB Boot) Mode	f_{CPU} : 240 MHz f_{IOBUS} : 120 MHz
100b	SPI Slave Boot Mode {GPIOE[4]} = 0 : CMD polarity is active low {GPIOE[4]} = 1 : CMD polarity is active high {GPIOE[0]} = 0 : PCK polarity is rising edge {GPIOE[0]} = 1 : PCK polarity is falling edge	f_{CPU} : 240 MHz f_{IOBUS} : 120 MHz
101b	SDMMC Boot Mode {GPIOE[4]} = 0 : The boot port is 0. (SDMMC port 4) {GPIOE[4]} = 1 : The boot port is 1. (SDMMC port 5) {GPIOE[0]} = 0 : 4bits SDMMC boot. {GPIOE[0]} = 1 : 1bits SDMMC boot.	f_{CPU} : 240 MHz f_{IOBUS} : 120 MHz f_{SDCLK_MAX} : 24 MHz
110b	NOR Flash / UART Boot Mode {GPIOE[4],GPIOE[0]} = 00b : 8bits NOR Flash {GPIOE[4],GPIOE[0]} = 01b : 16bits NOR Flash {GPIOE[4],GPIOE[0]} = 10b : UART port 0 {GPIOE[4],GPIOE[0]} = 11b : UART port 1	f_{CPU} : 240 MHz f_{IOBUS} : 120 MHz
111b	NAND Flash Boot Mode {GPIOE[0]} = 0b : NAND_BOOT (V1) {GPIOE[0]} = 1b : NAND_BOOT (V2) {GPIOE[4]} = 0b : RDY = GPIOB[31] {GPIOE[4]} = 1b : RDY = EDIRDY0 (GPIOB[28])	f_{CPU} : 240 MHz f_{IOBUS} : 120 MHz

In the TCC8900, there is an internal boot ROM for system initialization process. It contains the fundamental routines for system initialization or boot procedure through various device or interface such as EHI, I2C master, SPI, USB device, parallel NOR flash, serial EEPROM with SPI or I2C protocol, NAND flash, SD/MMC.
The TCC8900 uses 3 clock domains in system boot operation. They are fCPU , fIOBUS , fMBUS (CPU, IOBUS, Memory bus)). The fCPU, fIOBUS and fMBUS are fixed to 240, 120 MHz and 120MHz..

Table 6.2: Frequency of f_{MBUS}

MODE	PLL VALUE	FREQUENCY
00b	PMS = (3, 240, 2)	f_{MBUS} : 120 MHz
01b	PMS = (3, 090, 1)	NOT USED
10b	PMS = (2, 100, 1)	NOT USED
11b	PMS = (2, 80, 2)	NOT USED
Default after power-on boot	-	f_{MBUS} : 120 MHz

There are 9 modes for booting procedure. The Figure 6.2 illustrates the timing of reset sequence at power-up.

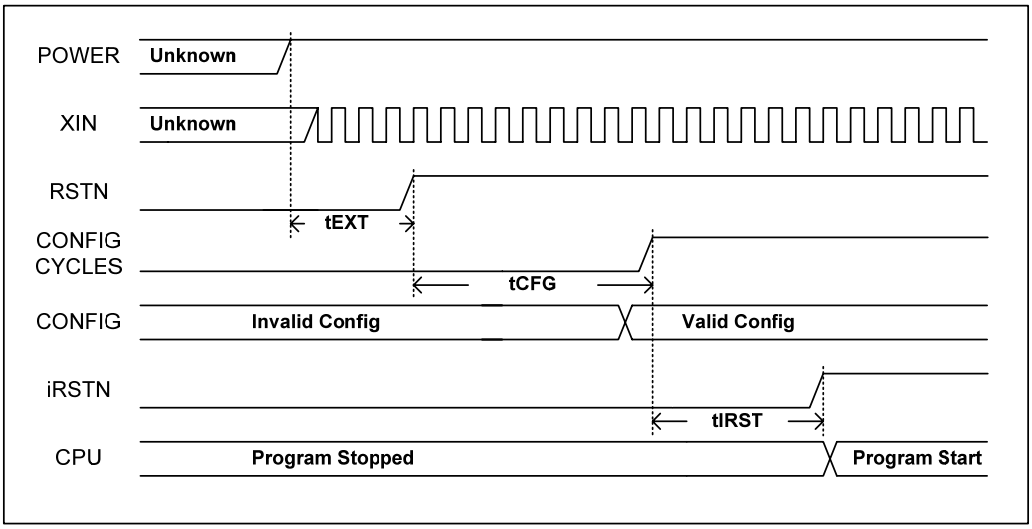


Figure 6.2 Reset Sequence

* The XIN and RSTN are the external pins and other signals are internal signals.
* The BM[2:0] are in input state during the “tEXT” and “tCFG”.

In the above figure, the ‘tEXT’ time is determined by external POR reset circuit or component and ‘tIRST’ is fixed to 64 clock cycles. But the ‘tCFG’ is about 1,048,576 clock cycles.

6.3 Overall Procedure

The following figure shows the overall flowchart of boot code.

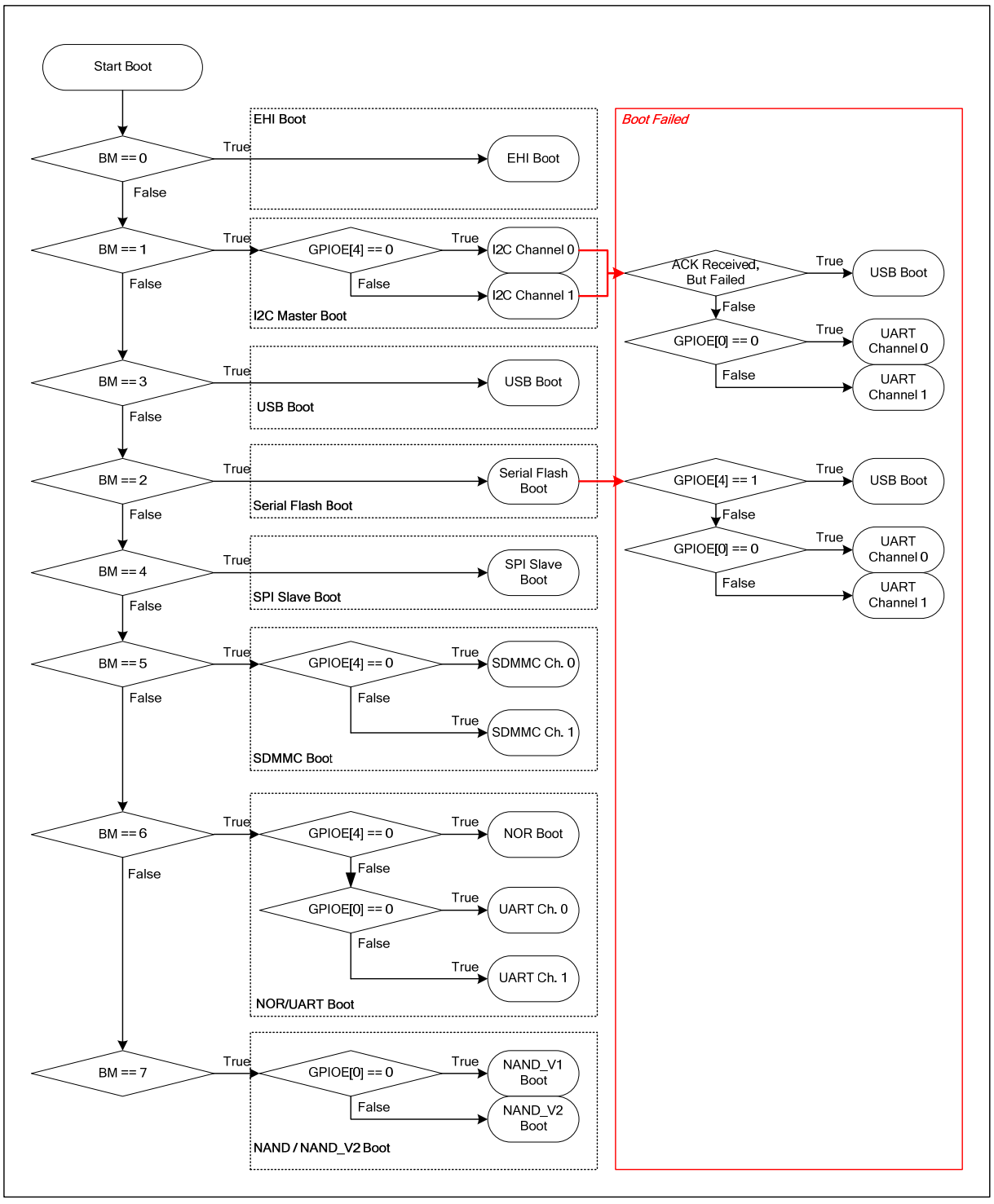


Figure 6.3 Overall Flowchart of Boot Code

6.4 EHI Boot (BM[2:0] = 000b)

In this mode, the on-chip CPU enables EHI module and sets HPINT to high. In this time, operating clock of EHI block is 120MHz. At the same time, the external host must wait until HPINT is high. After HPINT is asserted, the external host sets the ST field of EHST to INITCFG which indicates the interface bus width, the validity for this packet, and the total size of the BIP(Boot Information Packet).
After then, the on-chip CPU re-initializes the EHI with the bus width information in INITCFG.

7	6	5	4	3	2	1	0
1	0	0	0	0	BIPEN	VAL	BW

NOTE) BIIPEN = 0 (recommend)
VAL = should be 1 for valid INITCFG
BW = 0 (8bits), 1 (16bits)

Figure 6.4 INITCFG Bit Field

For handshaking between this LSI and the external host, the ST field of EHST register is used. The following is EHI booting procedure and refer to Figure 6.7.

(1) After boot codes reinitialize this LSI using INITCFG, the EHST.ST is set to 0x55 by the on-chip CPU (On-chip CPU).
(2) The external host uploads the data to this LSI. When uploading is completed, the external host set EHD register to start address of uploaded data and set EHST.ST to 0xAA. (External host).
(3) When upload is completed, EHST.ST is not equal to 0x55. The on-chip CPU determines that type of the uploaded data is the BIP or the firmware program using the BIPEN field of INITCFG.(On-chip CPU)
(4) If INITCFG. BIPEN is zero, the on-chip CPU jumps to start address in EHD register and EHI boot is completed. Otherwise, the uploaded data is the BIP and start address is BIP address. The BIP has much information about booting from EHI(Figure 6.5). After receiving the BIP from the external host and processing it, the on-chip CPU set the EHST.ST to 0x55. After then, the external host can upload program.(On-chip CPU)
(5) The external host fills buffer0 that it is indicated by BIP. If it is full, the external host sets EHST.ST to 0x2C. After then, the external host fills buffer1 when it is full. And the external host sets EHST.ST to 0x25(Figure 6.6). This procedure is repeated by the external host until all of data is uploaded.(External host)

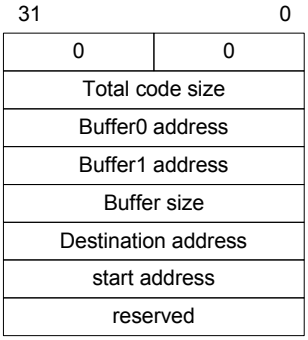


Figure 6.5 BIP Data Structure

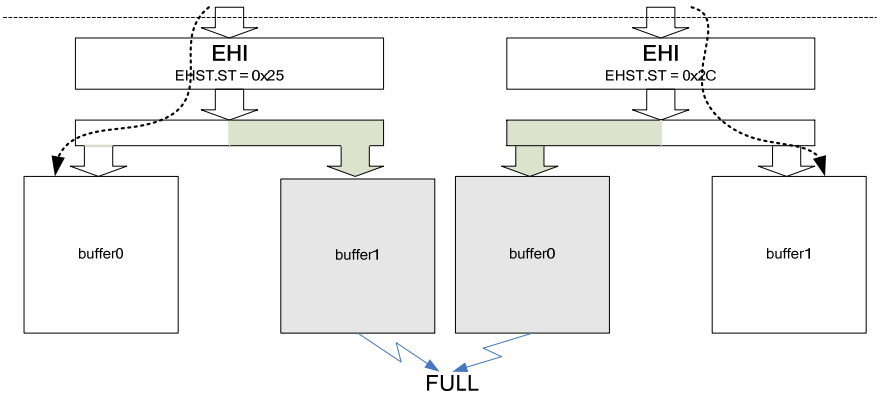


Figure 6.6 External Host Uploads Program when BIPEN = 1

The EHI can manipulate either of 80 interfacing. The bus-width of EHI is determined by the BW field of INITCFG. During the procedure for booting from EHI, the external host can access all the address space in the TCC8900. Therefore, if you want to read from or write to any register in TCC8900, just do it as you want

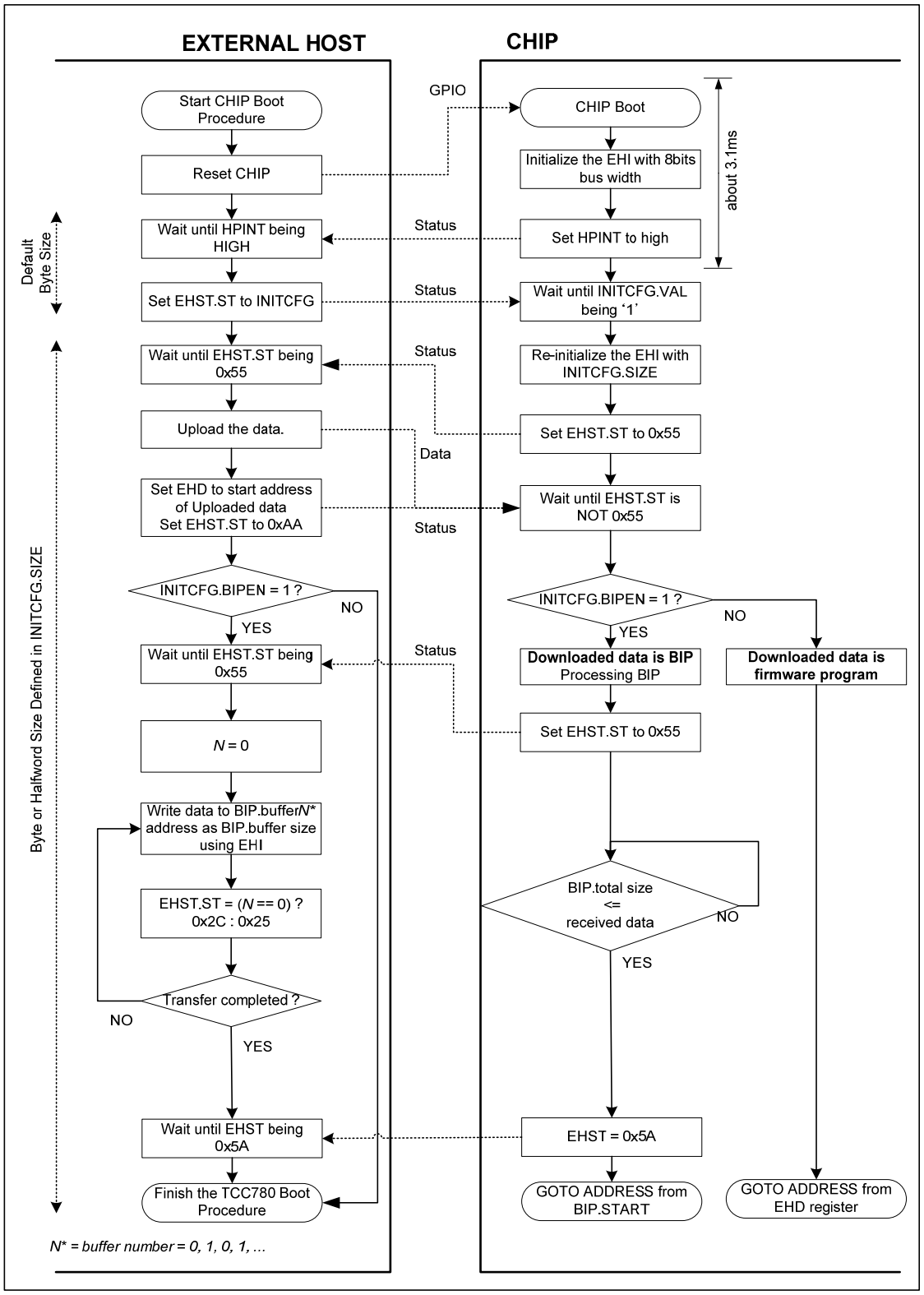


Figure 6.7 EHI Boot Procedure

6.5 USB Boot (BM == 011b)

This mode is mainly for firmware upgrade mode. In this mode, user can download a program into the user defined area When the failure occurs in some other boot modes, it may progress this boot sequence also.
The procedure of this mode is as follows.

(1) The TCC8900 makes internal SRAM area starts from zero, and copies USB interrupt service routine to internal SRAM area.
(2) It waits until USB connection is established.
(3) Once it is connected, host transfers first the parameter for USB loader routine including start address, destination address and the amount of data to be transferred (with a unit of packet).
(4) The TCC8900 starts communicating between a host PC with fixed amount of data which is called as packet. The packet size of TCC8900 is 512 bytes.
(5) At every successful reception of packet, it copies them where the destination address pointed, and after all amount of data has been copied, it starts program where the start address pointed.

Normally, the program downloaded is for writing user system firmware to non-volatile memory like NAND flash.
The following figure illustrates the sequence of USB boot mode described above.

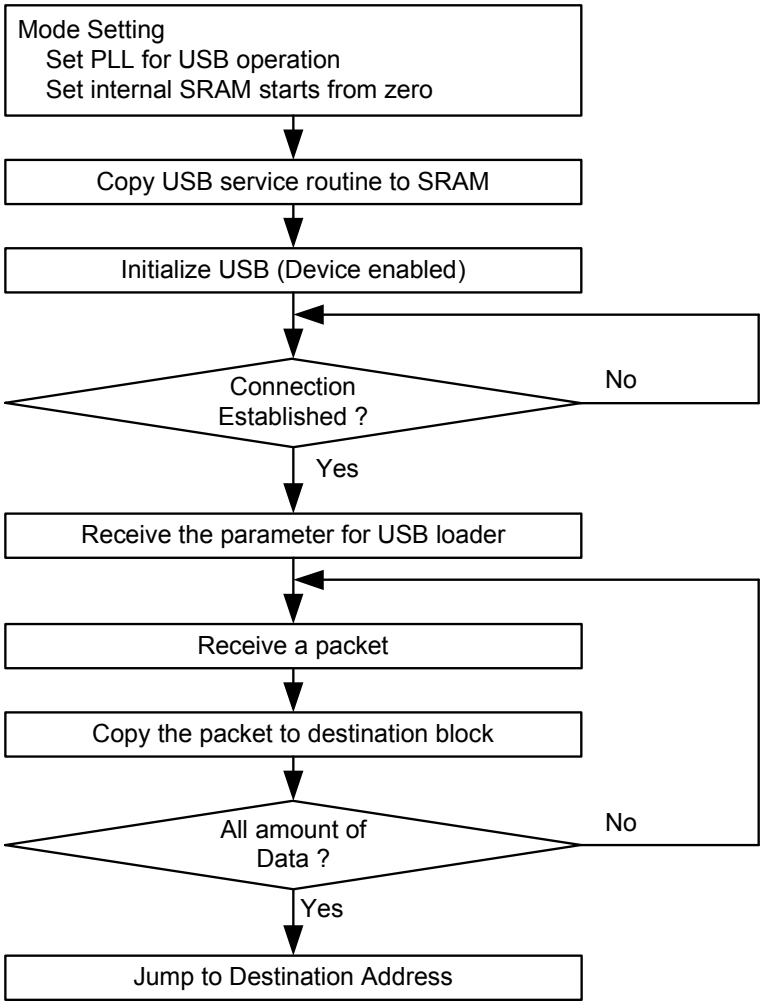


Figure 6.8 USB Boot Procedure

6.6 External NOR Boot (BM == 110b, GPIOE[4] == 0b)

In this mode, the contents of NOR flash can be examined by checking CRC. If it is OK, it boots like a normal mode. If it is not OK, the TCC8900 automatically changes to USB boot mode so user can fix NOR contents via USB interface. NOR flash must be attached to CSN_NOR(GPIOB[26]) pin. The detailed procedure is as follows.

(1) If 1st or 2nd word is 0xFFFFFFFF, TCC8900 goes to USB boot mode for F/W downloading.
(2) If the 31'th bit of 5'th word in NOR flash is zero, the CRC checking procedure will be skipped.
(3) The TCC8900 do the C2 (Dual CRC Checking) process on the image of external NOR flash. The C2 process is described at the Dual CRC Checking section.
(4) After CRC process has finished and if it returns "OK", TCC8900 finishes booting procedure by jumping to the destination address.

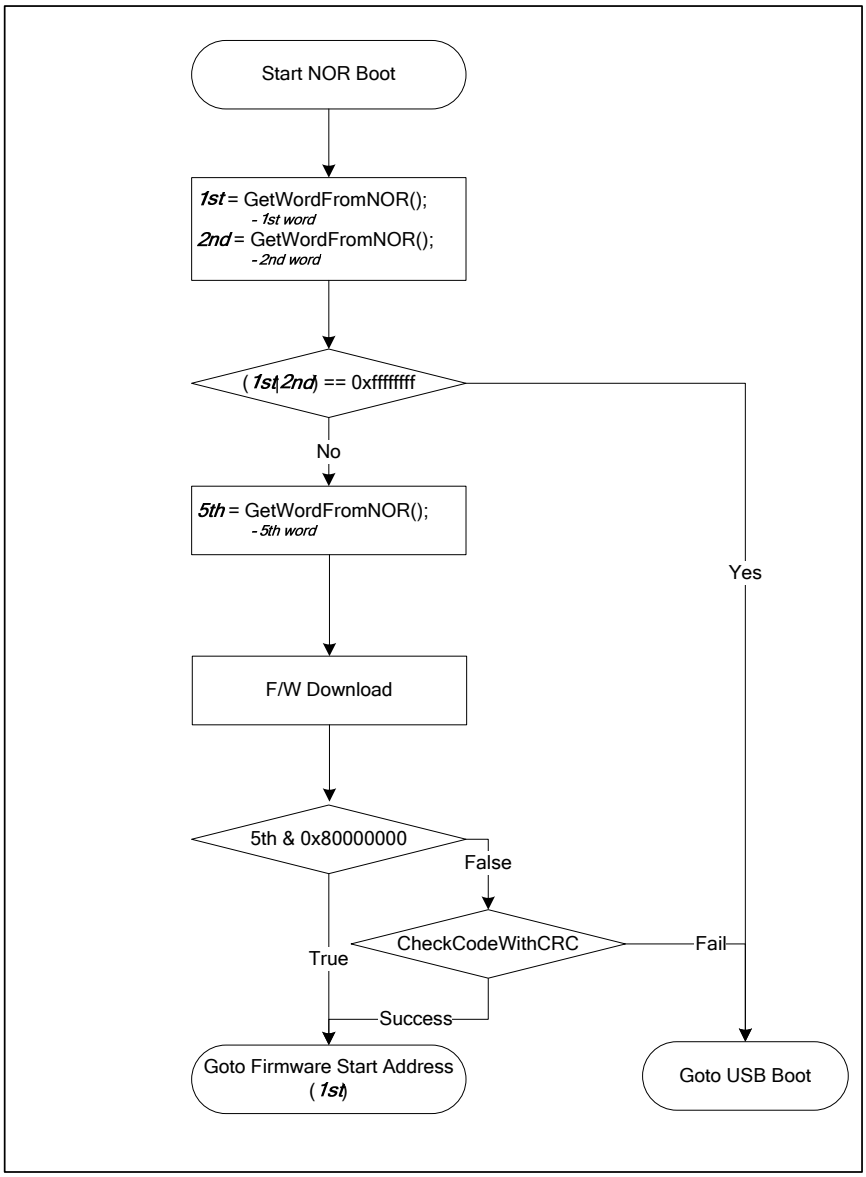


Figure 6.9 External NOR Boot Procedure

6.7 NFC NAND Boot (BM == 111b, GPIOE[0] = 0b)

In this mode, the F/W code is read from NAND flash attached. To make use of this mode, the predefined structure such as Master Block and Master Cluster should be fused in the NAND flash.
The supportable configuration of NAND flash is as follows.

Case1) one NAND flash of 8-bit bus-width.
Case2) one NAND flash of 16-bit bus-width
Case3) two NAND flashes of 8-bit bus-width with the same chip enable signals.

In case of Case3, the two NAND flashes can share the 16bit data-bus by using upper and lower 8 bit separately. The GPIOE[4] should be 1, the RDY of NFC is connected to GPIOB[31], the other is connected to EDIRDY0 (GPIOB[28]).
The boot sequence of this mode is as follows. If there exists any problem hard to recover during this sequence, it goes to USB boot mode automatically. It assumes the contents is stored as little endian format.

(1) Check if device ID exists in the device ID table (Refer to Table 6.3). Determine the configuration of NAND flashes (Case1 ~ Case3). Setup the parameter for NFC block (pages per block, number of address cycles, etc.) according to device ID.
(2) Read the last 1 word (4 bytes) in the spare area of 0 page of 0 block. It contains the block address of Master Block in upper 3 bytes, and number of Master Cluster in lower 1 byte.

31	30 8	7 0
	Block Address of Master Block	Number of Master Cluster

(3) Load and Construct the golden image of Master Cluster from the Master Block to internal SRAM. The structure of Master Block and Master Cluster is represented in Figure 6.11.
(4) The 1st word of golden Master Cluster determines the next procedure as follows. In case of 0x54C34397 ('T', C3, 'C', 97), the TCC8900 regard the Master Cluster as the Master Code, and finishes booting procedure by jumping to the address of 2nd word in golden Master Cluster. In case of 0x54C34997 ('T', C3, 'I', 97), the TCC8900 starts loading according to the contents of golden Master Cluster.
(5) After finishing download of F/W, the TCC8900 progress C^2 (Dual CRC Checking) process on the loaded image. The C^2 process is described at the "6.14 Dual CRC Checking" on page 6-108.
(6) After C^2 process has finished, and it is OK, the TCC8900 finishes booting procedure by jumping to the destination address which is contained in golden Master Cluster.

In loading from NAND flash, the TCC8900 uses BCH-ECC regardless of NAND type, so loading is accomplished by unit of 512+16 bytes or 512+20. The BCH-ECC can correct up to 16bits/512Bytes errors, so the F/W code can be stored with high robustness.

Table 6.3 Supported NAND Flash Types

Size (bytes)	Size of Page (bytes)	Pages / Block	Number of Page	Device ID x8 / x8 / x16 / x16
8M	512	16	16K	39 / E6 / 49 / 59
16M	512	16	32K	33 / 73 / 43 / 53
32M	512	32	64K	35 / 75 / 45 / 55
64M	512	32	128K	36 / 76 / 46 / 56
128M	512	32	256K	78 / 79 / 72 / 74
256M	512	32	512K	71
512M	512	32	1024K	DC
128M	2048	64	64K	A1 / F1 / B1 / C1
256M	2048	64	128K	AA / DA / BA / CA
512M	2048	64	256K	AC / DC / BC / CC
1G	2048	64	512K	A3 / D3 / B3 / C3
2G	2048	64	1024K	A5 / D5 / B5 / C5
8G	4096	128	2048K	D7/D3

Table 6.4 Each Mode as NAND Types

NAND TYPE	Data Page Size	Spare Size	ECC MODE
Small Type	512 Bytes	16 Bytes	BCH 4BIT
Large type	2048 Bytes	64 Bytes	BCH 4BIT
Extended Large Type	4096 Bytes	128 Bytes	BCH 4BIT
	4096 Bytes	218 Bytes	BCH 12BIT

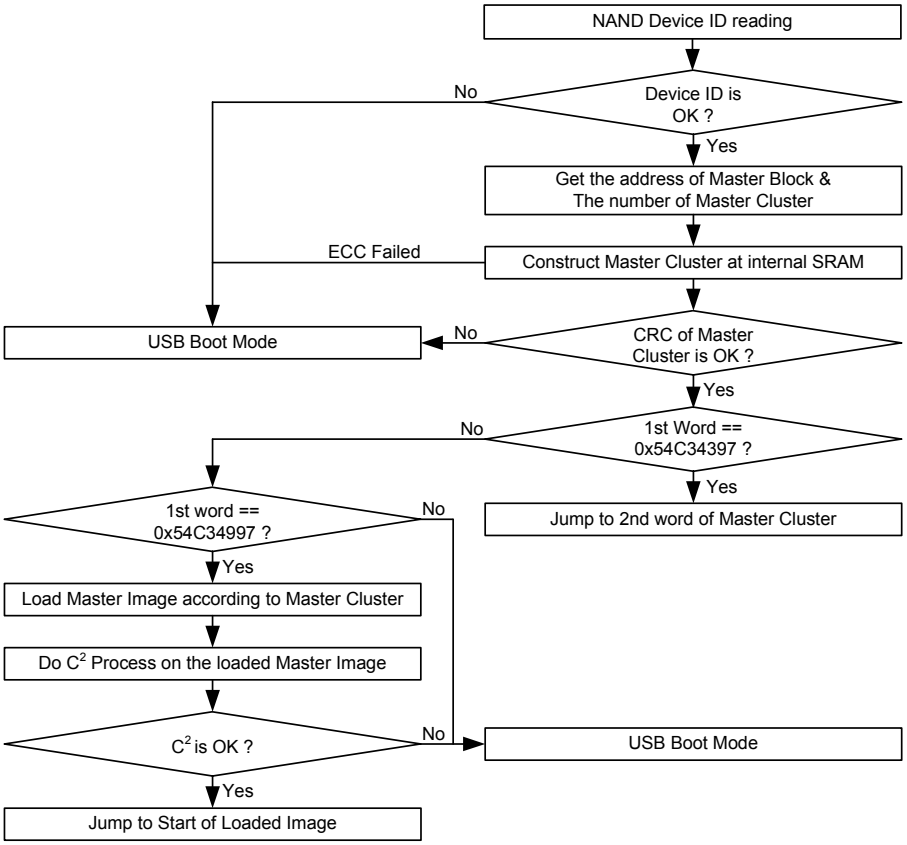


Figure 6.10 NAND Boot Procedure

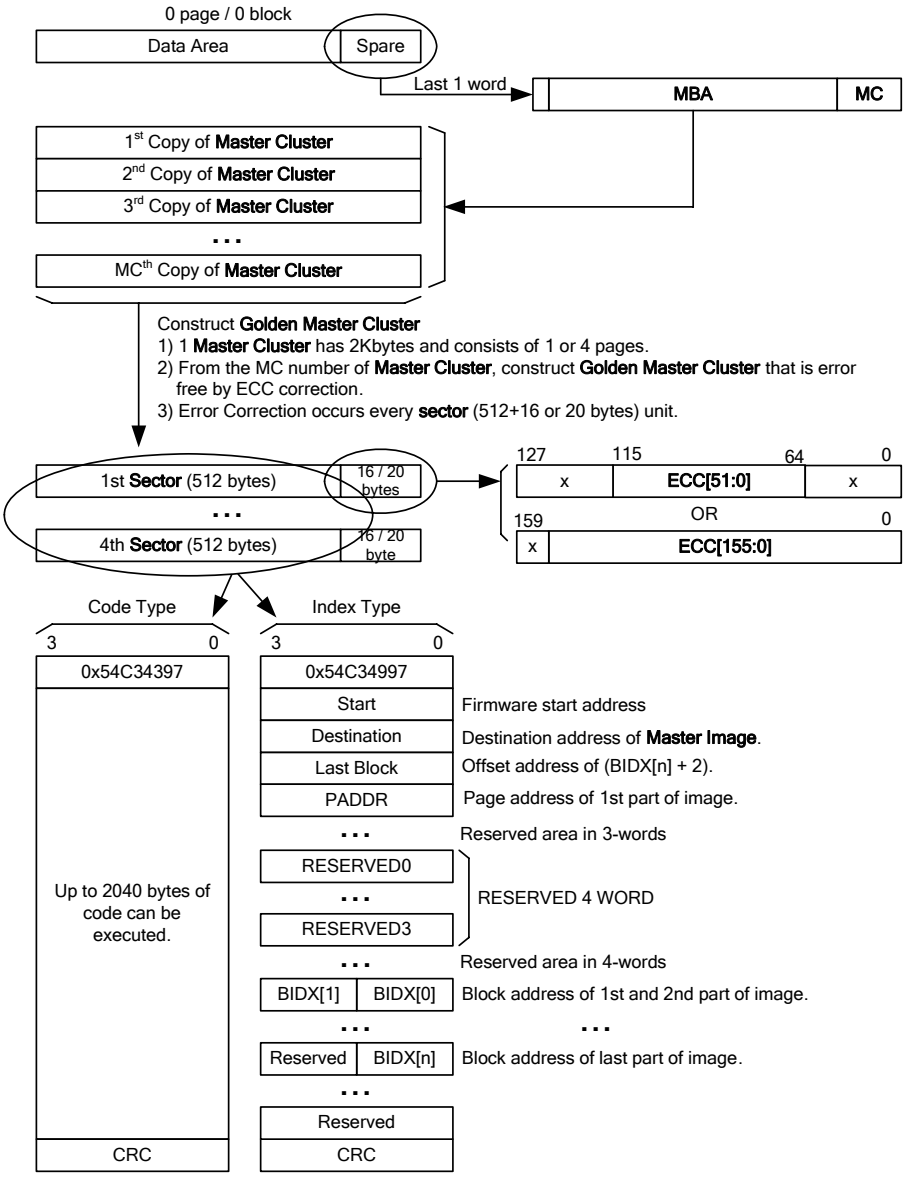


Figure 6.11 Basic Structure and Boot Flow of NAND Boot

6.8 I2C Master Boot – EEPROM Boot (BM == 001b)

In this mode, the F/W code would be read from serial EEPROM attached to the I2C ports. It interfaces with standard I2C protocol. If serial EEPROM is not attached, the TCC8900 changes to UART boot mode.
The procedure checks if there exist EEPROM first. If there exist an EEPROM, the TCC8900 do the following procedure. If certain problem which can not be solved has occurred, it goes to USB boot mode automatically.
The UTXD1 port determines the boot channel for I2C. If pulled-up, the channel 1 port would be used and otherwise the channel 0 port used.

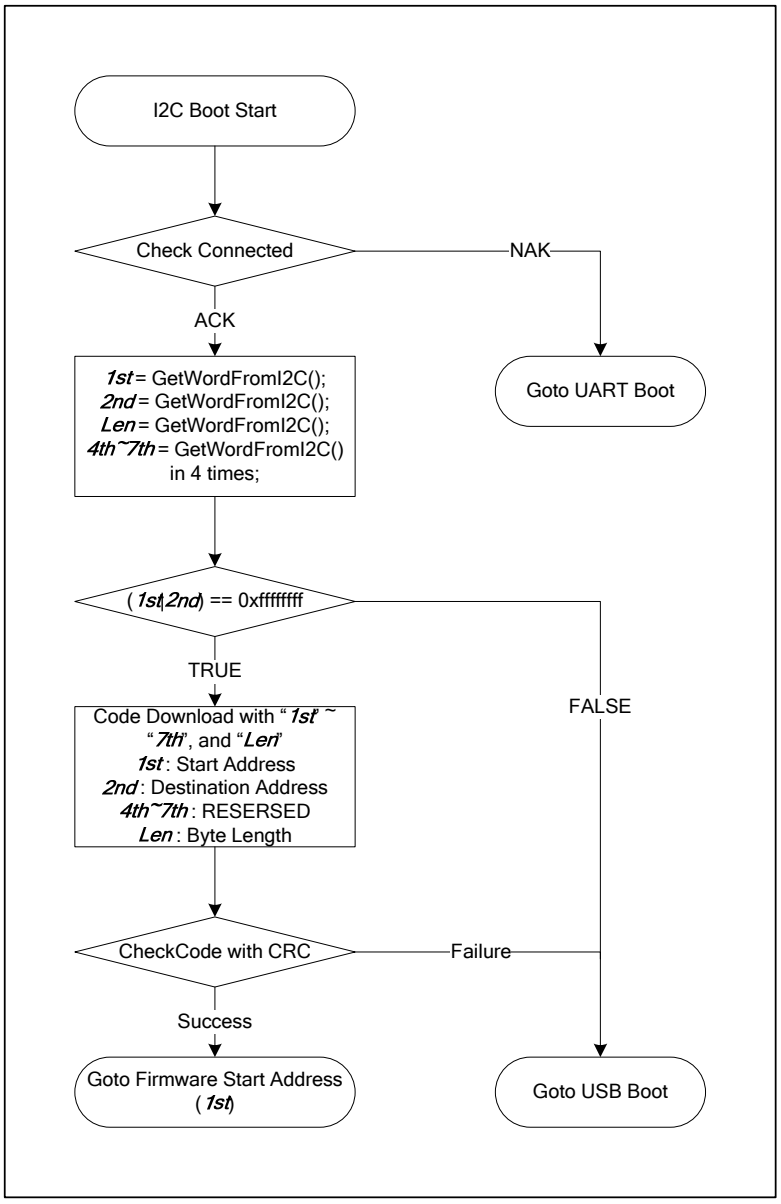


Figure 6.12 I2C Boot Procedure

6.9 Serial Flash Boot (BM==010b)

The serial flash boot mode supports the serial NOR flash with SPI protocol. The GPIOB[11:8] are used for interfacing the serial EEPROM. The GPIOE[4] pin determines which mode to branch after serial EEPROM booting failed. If GPIOE[4] is pulled-up, the boot mode would be changed to USB boot after failed and otherwise to the UART boot mode. In the case of UART boot mode, the GPIOE[0] determines that which port can be used in UART boot mode. If GPIOE[0] is pulled-up, the UART port 1 will be used and otherwise UART port 0 used.

```
typedef       struct {
  unsigned short       MID;                    // Unused for booting
  unsigned short       DID;                    // Unused for booting
  unsigned             SectorSize;             // Unused for booting
  unsigned             BlockSize;              // Unused for booting
  unsigned             BlockNum;               // Unused for booting
  unsigned             uStart;                 // Used for booting
  unsigned             uDest;                  // Used for booting
  unsigned             uLength;                // Used for booting
  unsigned             uDLDV;                  // Used for booting
  unsigned char        Name[20];               // Unused for booting
  unsigned char        Rsv[256-56];            // Unused for booting
              // Rsv[100]              // Used for booting
                  //      :               // Used for booting
                  //  Rsv[115] // MCFG0~3   // Used for booting
  unsigned             uCRC;                   // Used for booting
} sSFBootHeader;
```

Figure 6.13 Data Structure for Header Information Stored In Serial Flash

In the above figure, the data structure required for booting from serial flash is described. The variables named as uStart, uDest, uLength, and Rsv[100] to Rsv[115] are used in the boot code. If the uDest is not equal to uStart, the uDLDV is used to DLDV, which determines the clock frequency of GPSB(General Purpose Serial Bus) controller.
The uLength field is total number of bytes to stored in the serial flash and should be aligned to word(4bytes).
The overall procedure for serial flash boot is shown in the following figure.

```
Void IO_GPSBM_GetWordData (unsigned *data, unsigned nwords);
// Function to read from serial flash
// “data” is destination pointer
// “nwords” is the total number of words to read

Void StartSFlashBoot ()
{
   Unsigned char mem_header[16];

    IO_GPSB_PortConfig ();      // Port Configuration for Serial Flash Interface
    IO_GPSBM_Init ();  // Initialization for GPSB controller
    IO_GPSB_FlushBuffer ();     // Flushing Buffer for Initial Loading
    IO_GPSBM_GetWordData ((unsigned *)pHeader,0,sizeof(sSFBootHeader)/4);

    if ( IsEmptyFlash () ) return (-1); // Booting Failed

    uData = IO_UTIL_CalcCRC32(0,(unsigned *)pHeader,256-4,0);
    if ( uData != pHeader->uCRC ) return (-1);// Booting Failed

for (i=0;i<16;i++)
  mem_header[i] = pHeader->Rsv[i+100];

    IO_UTIL_SetMEM (pHeader->uDest,mem_header);
    // Initialization for Memory Controller

    if ( (pHeader->uDest&0xf0000000) != (pHeader->uStart&0xf0000000) ) {
      HwGPSB0->uMOD.bMOD.DLDV = pHeader->uDLDV;
      uExecStart = pHeader->uDest;
    }

    IO_GPSB_FlushBuffer ();
    IO_GPSBM_GetWordData
    ((unsigned *)pHeader->uDest, SFLASH_PAGE_DATA_SIZE, (pHeader->uLength>>2));
    // SFLASH_PAGE_DATA_SIZE : 256

    if (IO_UTIL_CheckCODE((unsigned *)pHeader->uDest, pHeader->uLength, 0))
      return 0;        /* to usb boot */

    goto_firmware (pHeader->uStart);
}
```

Figure 6.14 Pseudo Code for Serial Flash Booting Procedure

6.10 SPI Slave Boot (BM==100b)

The SPI slave boot uses the ports named GPIOB[11:8]. And if boot failed, the boot mode would jump to the USB boot – USB firmware download mode.
Accroding to input configuration of GPIO[4] and GPIOE[0], the polarities of CMD(GPIOB[8] and PCK(GPIOB[9]) are determined, If GPIOE[4] is pulled down to low, the CMD operates as active low signal and if pulled up to high, it operates as active high signal. Similiarly, if GPIOE[0] is pulled down to low, the rising edge of PCK is used and if GPIOE[0] is pulled up to high, the falling edge of it is used to check its data.

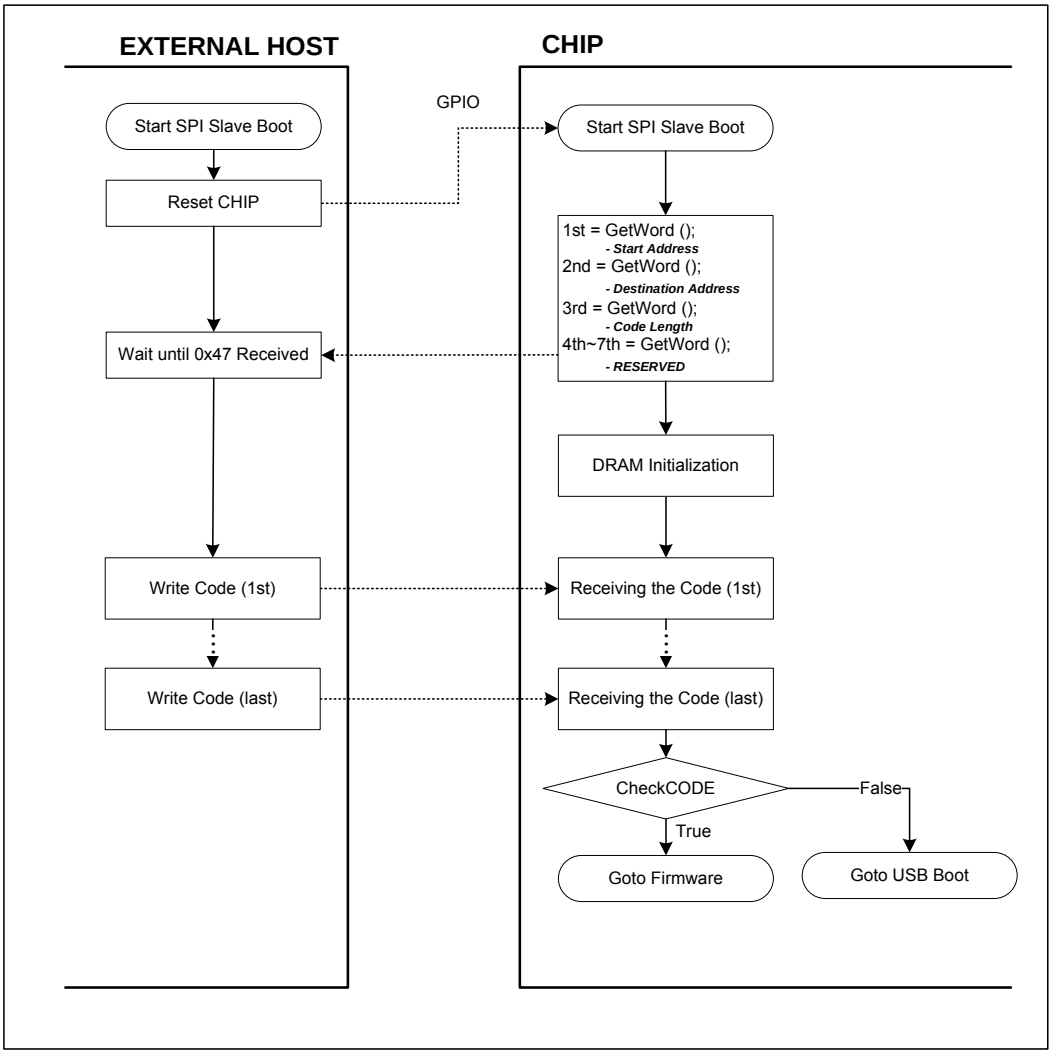


Figure 6.15 SPI Slave Boot Procedure

6.11 UART Boot (BM==110b, GPIOE[4]==1b)

The UART boot mode can be changed from I2C boot mode or serial flash boot mode. If the I2C eeprom boot mode was failed, the UART boot would be started with the configuration of the GPIOE[0] pin. If GPIOE[0] is pulled-up, the UTXD0(GPIOE[0]) and URXD0(GPIOE[1]) will be used for booting and if GPIOE[0] is pulled-down, the UTXD1(GPIOE[4]) and URXD1(GPIOE[5]) will be used.
The detailed procedure is as follows. It uses little endian for word manipulation. So, the LSB byte is received first, and MSB byte is received last.

(1) Enable UART I/F. (fUART=24MHz, 115.2kbps, Data 8bit, Non-parity, 1 Stop bit)
(2) Send ACK (0x52, ASCII code of 'R').
(3) Wait until ACK is received. ('H' or 'R')
(4) Send ACK
(5) Receive the Start Address.
(6) Send ACK
(7) Receive the Destination Address.
(8) Send ACK
(9) Receive the size of code in byte unit.
(10) Send ACK
(11) Receive the 4 words RESERVED data.
(This value is not used in rev.ax chip. But always received 4 word value.)
(12) Send ACK and go to routine (11) if memory parameter is not RESERVED3.
(13) If ACK in (3) is 'H', new divisor value (DLM, DLL) is received (fUART = 24MHz) and baud rate is reconfigured.
(14) Load the code from HOST to destination address
(15) Do the CRC checking process on the downloaded image.
(16) After CRC checking process has finished, and it is OK, TCC8900 finishes booting procedure by jumping to the destination address. Or, it restarts UART boot from the first step.

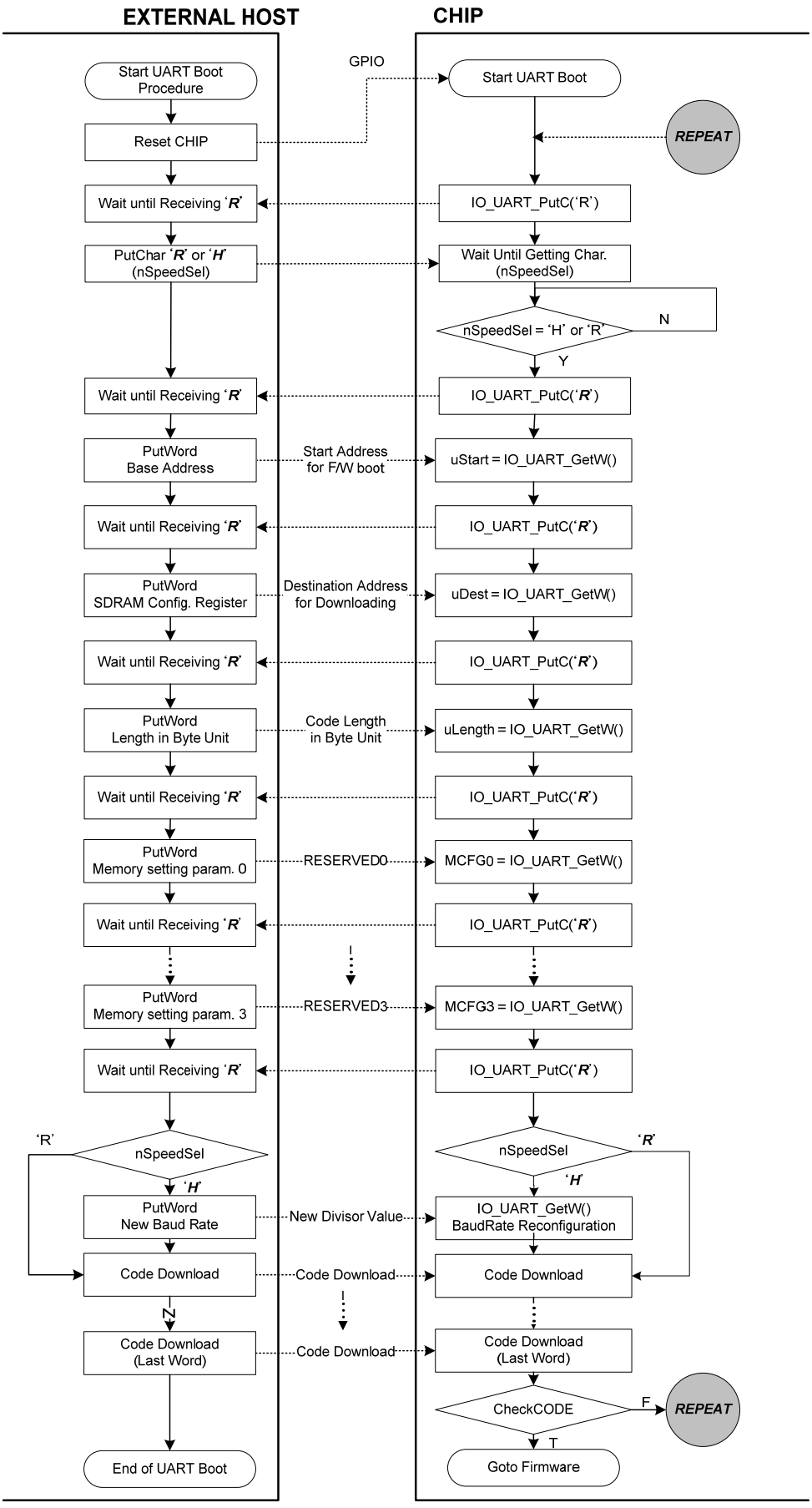


Figure 6.16 UART Boot Procedure

6.12 SD/MMC Boot (BM==101b)

The TCC8900 supports SD/MMC boot mode ,like SD, MMC, MMC+, SDHC. The GPIOE[0] indicates bit width of operation, if it is pulled down to low, the bit width will be 4 and if pulled up to high, the bit width will be 1. The GPIOE[4] is related to port mapping. If it is pulled down to low, GPIOE[11:6] will be mapped into {DATA[3:0],CMD,SDCLK} and if pulled up high, {GPIOB[3:0],GPIOB[12:13]} is mapped into {DATA[3:0],CMD,SDCLK}.
The detailed procedure is as follows. It uses little endian for word manipulation. So, the LSB byte is received first, and MSB byte is received last.

(1) Enable SD/MMC I/F and initialization. (fSDMMC=48MHz, SDCARD_CLK limit= 24MHz)
(2) Card Reset.
(3) Scan Card for checking SD/MMC/MMC+/SDHC (SD_CLK = 170 KHz)
(4) Get response for identification.(OCR,CID,RCA,CSD)
(5) Set SD_CLK for transfer mode (using SDCARD information CLK, it's limitation CLK is 24MHz)
(6) Read the first single block (512byte) for setting boot parameter
(start address, destination address, code length, memory setting parameter)
(7) Do the CRC checking process on the first single block.
(8) After CRC checking process has finished, and it it OK, setting memory controller using MCFG0~MCFG3 parameter. Or, It restarts USB boot from the first step.
(9) If user define parameter (SD port delay parameter) is enable, SD port delay parameters are change to user define parameter.
(10) If memory_initialize code size (0xc4 of first single block) is not zero, boot code is operated to the memoty initialize code. The other is this sequence is skipped.
(11) Calculating ROM code address and size refer to boot parameter.
(12) Read firmware data from card.
(13) Do the CRC checking process on the downloaded image.
(14) After CRC checking process has finished, and it is OK, TCC8900 finishes booting procedure by jumping to the start address. Or, it restarts USB boot from the first step.

The SD boot parameters are shown in following figure. These parameters are stored in the last block of hidden area in SD device. In boot mode, only 32 bytes out of 512 bytes(1 block) are used and last 4 bytes include CRC result.
The "user defined parameter(UDP)" is sdmmc port delay parameters. If. bit 31 is 'low', this parameter is skipped (default value is zero), the other, this parameter is set to user define value. This parameter is referred to SDMMC chapter in IOBUS datasheet. (Channel Control Register Map in 0xF05A0800)
The memory initialize code size is the size of memory initialize code. Unit of size is byte. The location of memory initialize code is last block munis 1 (last_block – 1) of hidden area in SD device. If size is bigger than 1 block, code is located from (last_block -2) to (last_block-1). First block is (last_block -2), next block is (last_block-1)

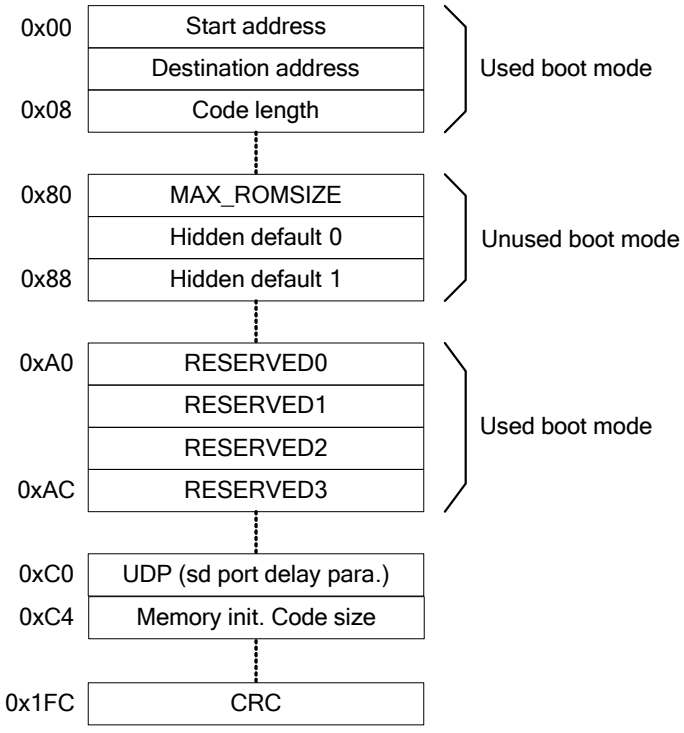


Figure 6.17 SD/MMC Boot Parameter

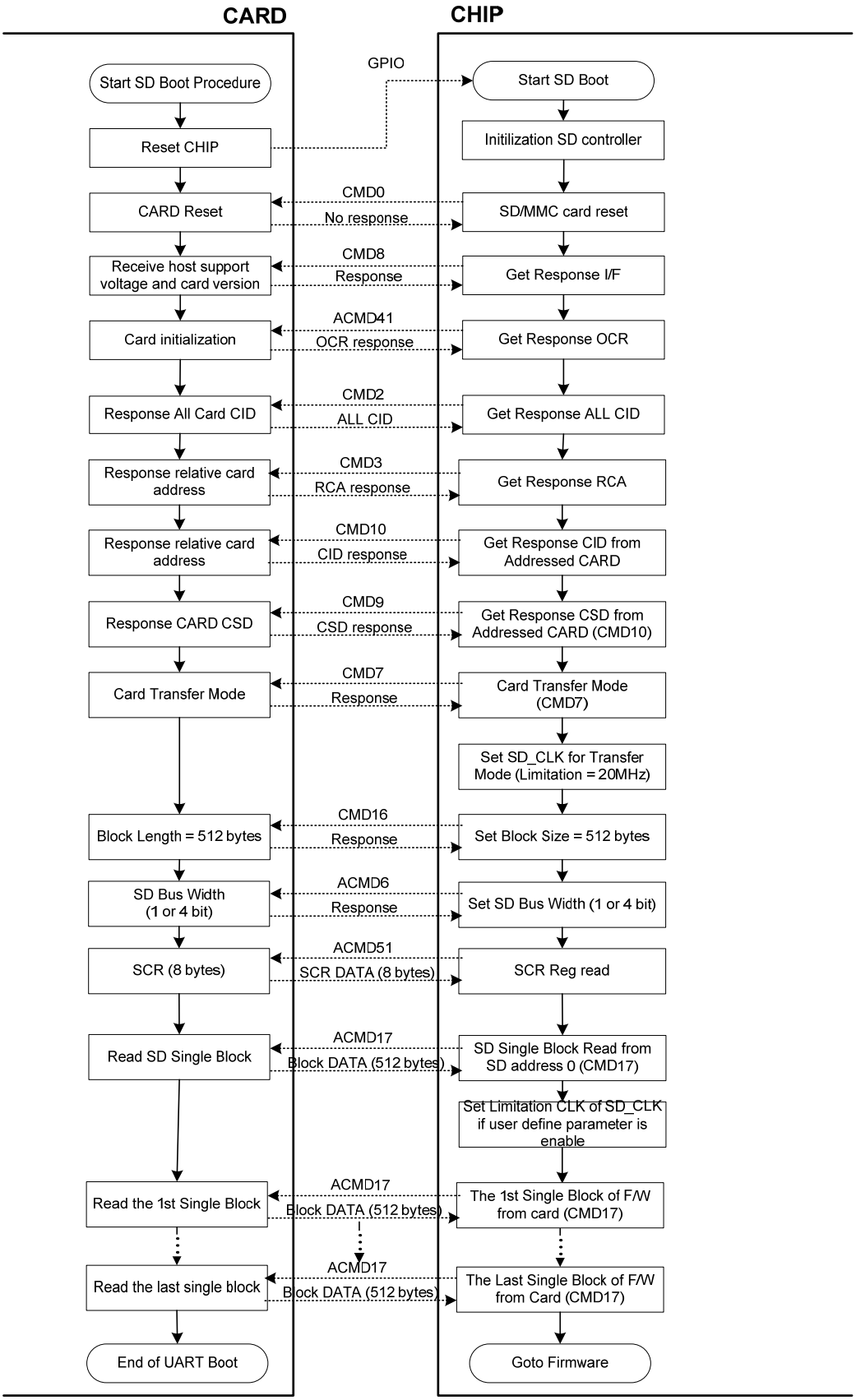


Figure 6.18 SD/MMC Boot Procedure

6.13 NAND_V2 Boot (BM==111b, GPIOE[0]= 1b)

The TCC8900 supports another NAND boot mode (NAND_V2). In this mode, the F/W code is read from NAND fresh attached, too. To make use of this mode, the predefined structure such as Master Block and Master Cluster should be fused in the NAND flash.
The supportable configuration of NAND fresh is the NAND_V1.
In this mode, the F/W code is read from NAND flash attached. To make use of this mode, the predefined structure such as Master Block and Master Cluster should be fused in the NAND flash.
The boot sequence of this mode is as follows. If there exists any problem hard to recover during this sequence, it goes to USB boot mode automatically. It assumes the contents is stored as little endian format.

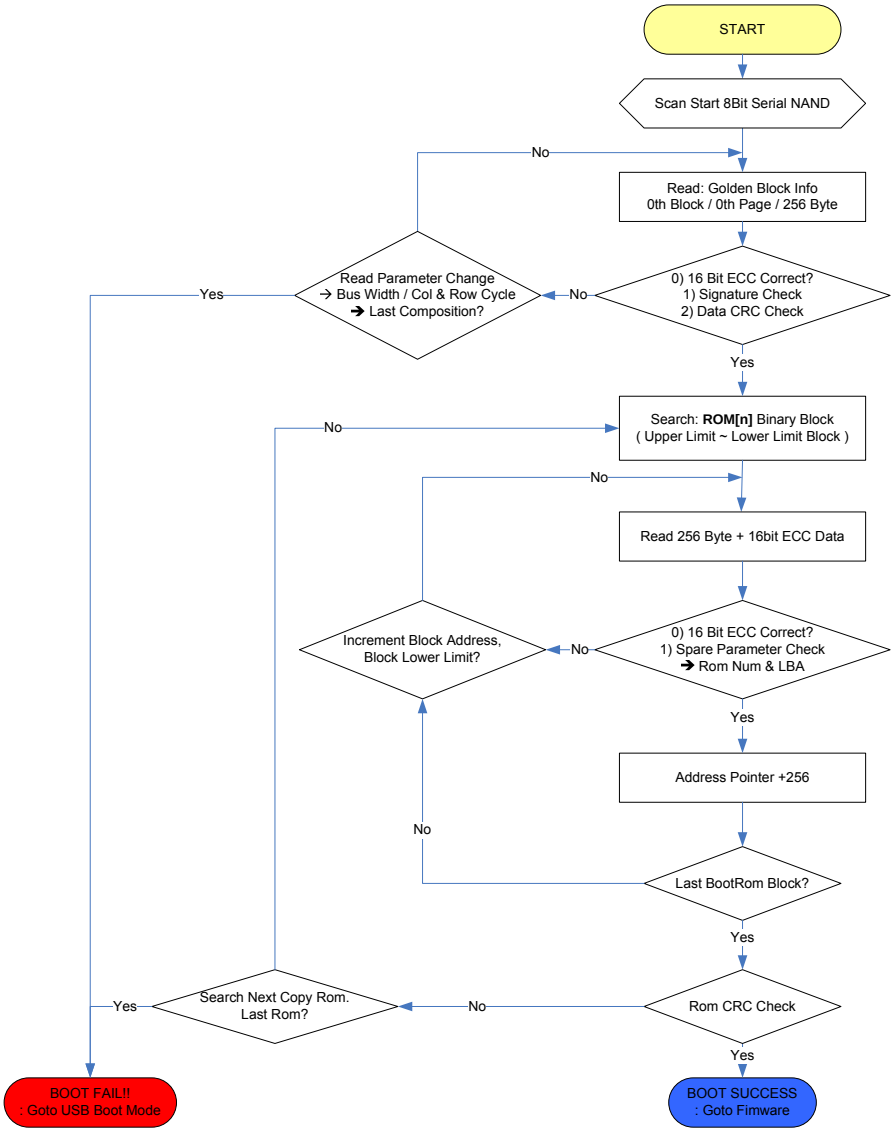


Figure 6.19 NAND_V2 Boot Procedure

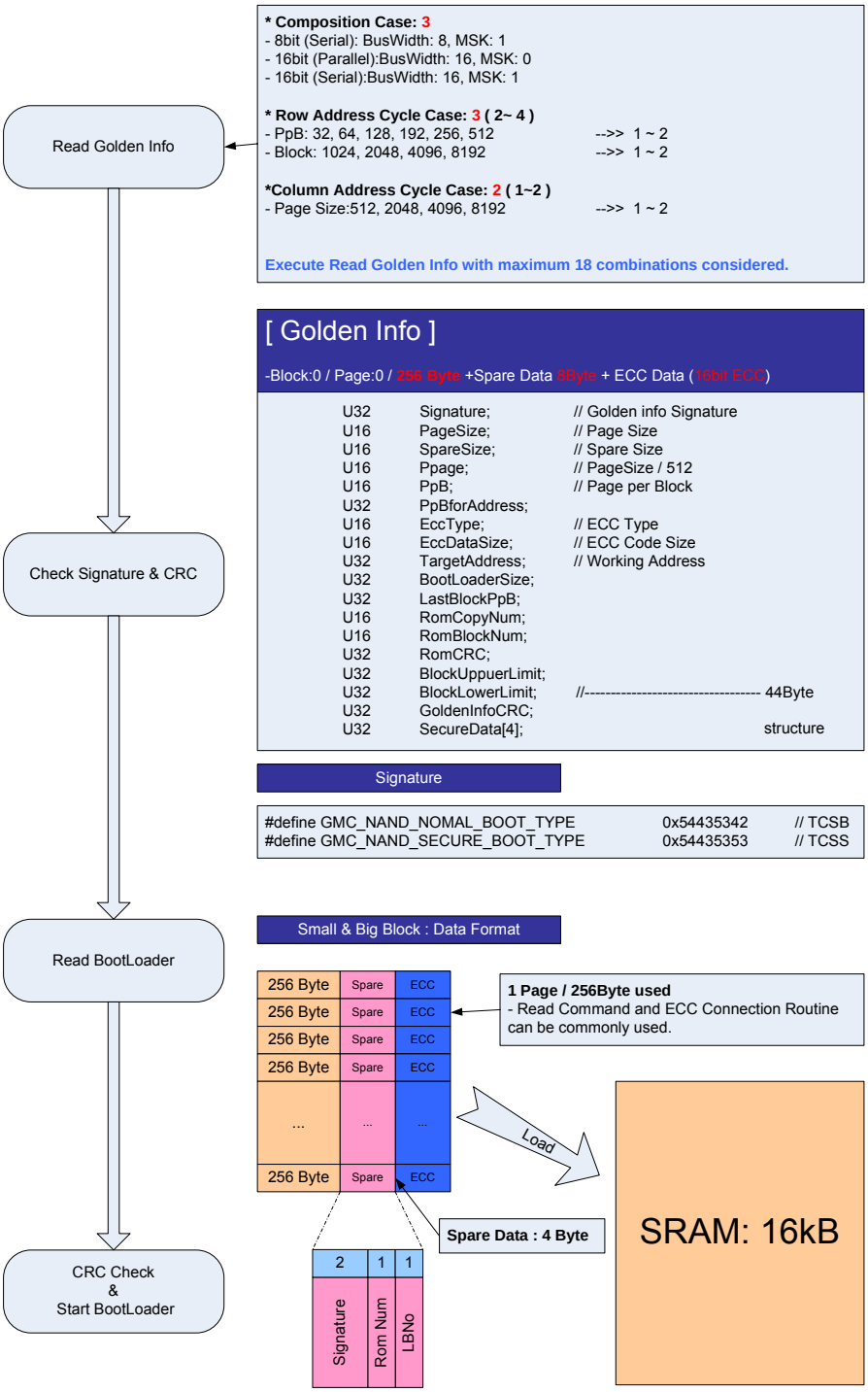


Figure 6.20 NAND_V2 Golden Info. Structure

6.14 Dual CRC Checking

The TCC8900 can check the CRC for the downloaded image according to the dedicated algorithm. This is described as follows.
Calculate CRC on the downloaded image up to 64Kbytes. In calculation, the word at the offset of 0x10 and 0x14 is skipped, because the good CRC code is contained at the offset of 0x10, and the word at the offset of 0x14 means the address of user-defined CRC routine.
After the first CRC calculation has been done, and it is same as the value at the offset address of 0x10, TCC8900 check if the address of user-defined CRC routine is not 0. Or, TCC8900 goes to pre-defined boot mode (USB or UART boot) automatically.
If there exist user-defined CRC routine, call that routine.
The user-defined CRC routine checks its own CRC code and return with Equal condition if it is OK or return Not Equal condition if it fails.
If there are failures on the user-defined CRC checking, TCC8900 goes to pre-defined boot mode (USB or UART boot) automatically.
The following code shows same CRC generation algorithm used in the first CRC routine.

```
word calc_crc (word *base, word length, word *crctable)
{   // contents of crctable is acquired by gen_crc () function
    word       crcout  = 0;
    word       cnt, i, code, tmp;

    length     >>= 2;           // convert into word unit.
    for (cnt = 0; cnt < length; cnt ++) {
      if (cnt == 0x04 or cnt == 0x05)  // skip offset of 0x10 and 0x14
      continue;
      code     = base[cnt];
      for (i = 0; i < 4; i ++) {
      tmp      = code ^ crcout;
      crcout   = (crcout >> 8) ^ crctable[tmp & 0xFF];
      code     = code >> 8;
      }
    }
    return     crcout;
}

void gen_crc (word *crctable)
{   // Polynomial = x^32 + x^31 + x^16 + x^15 + x^4 + x^3 + x + 1
    word       crc, cnt, i;

    for (cnt = 0; cnt < 0x100; cnt ++) {
      crc      = cnt;
      for (i = 0; i < 8; i ++) {
      if (crc & 1)     crc      = (crc >> 1) ^ 0xD8018001;
      else             crc      = (crc >> 1);
      }
      crctable[cnt]    = crc;
    }
}
```

PART3 - GPIO
TCC8900

High Performance and Low-Power Processor
For Digital Media Applications

Rev. 1.00

Aug 07, 2009

Telechips

Revision History

Date	Revision	Description
2008-12-11	0.00	* Initial Release
2009-02-10	0.01	* The GPIOF group doesn't have the pull-up and(or) pull-down resistors. The related documentation was removed.
2009-02-25	0.02	* The field descriptions for the Port Configuration Register 1,2 and 3 were changed. * The detailed descriptions for the corresponding port names was added in the port configuration registers.
2009-05-25	0.03	* The reset values of the GP*X*CD0/1, GP*X*PD0/1 (X=A,B,C,D,E,F) were changed. * The LXD[23:18] can not be supported – Only 18bits are supported for the CPU type LCD. * The TCO3,2,1,0 were changed to TCO0,1,2,3 correspondingly
2009-08-07	1.00	* The initial states of the GPIOs have been changed

TABLE OF CONTENTS

Contents

1 GPIO & Port Multiplexing 1
1.1 Overview 1
1.2 Register Description 2
1.3 Bypass Function 13

Figures

Figure 1.1 Bypass Mode in the TCC8900 13

Tables

Table 1.1 GPIO Register Map (Base Address = 0xF0102000) 2
Table 1.2 Port Configuration of GPIOA 7
Table 1.3 Port Configuration of GPIOB 8
Table 1.4 Port Configuration of GPIOC 9
Table 1.5 Port Configuration of GPIOD 10
Table 1.6 Port Configuration of GPIOE 11
Table 1.7 Port Configuration of GPIOF 12

1 GPIO & Port Multiplexing

1.1 Overview

The TCC8900 has a lot of general purpose I/Os which is able to be programmed by setting internal registers. All I/Os are set to input mode at reset. The block diagram of GPIO is in the following figure.

The GPIO in the TCC8900 has the following key features.

The I/Os have the multiple functions shared by various on-chip hardware controllers.

The GPSB, memory stick host controller, UART, and SD/MMC controller have the capability of interfacing to external device via the multi-channel ports by multiplexing the in/out signals of the corresponding hardware.

Most of the I/Os can be pulled-up or pulled-down by reconfigurable register.

The external interrupts passed to the interrupt controller can be selected from the various sources.

The GPIO controller has the special function. And the various I/Os are pulled-up or pulled-down at system reset to reduce the boot-up power consumption during the system reset, which prevents the reset state of the various I/Os from being floating state.

But, for this reason, if you want to design the specific I/Os with pull-up state which are pulled-down by reset configuration, you should be careful in choosing the value of resistor and the value of the pull-up/down control registers should be changed as soon as possible – for example, in the reset handler of initial boot code.

1.2 Register Description

Table 1.1 GPIO Register Map (Base Address = 0xF0102000)

Name	Offset	Type	Reset	Description
GPADAT	0x000	R/W	0x00000000	GPA Data Register
GPAEN	0x004	R/W	0x00000000	GPA Output Enable Register
GPASET	0x008	W	-	OR function on GPA Output Data
GPACLR	0x00C	W	-	BIC function on GPA Output Data
GPAXOR	0x010	W	-	XOR function on GPA Output Data
GPACD0	0x014	R/W	0x00005555	Driver strength Control 0 on GPA Output Data
GPACD1	0x018	R/W	0x00005555	Driver strength Control 1 on GPA Output Data
GPAPD0	0x01C	R/W	0x00055505	Pull-Up/Down function on GPA Output Data
GPAPD1	0x020	R/W	0x00005555	Pull-Up/Down function on GPA Output Data
GPAFN0	0x024	R/W	0x00000000	Port Configuration on GPA Output Data
GPAFN1	0x028	R/W	0x00000000	Port Configuration on GPA Output Data
GPAFN2	0x02C	R/W	0x00000000	Port Configuration on GPA Output Data
GPAFN3	0x030	R/W	0x00000000	Port Configuration on GPA Output Data
-	0x034-0x03C			Reserved
GPBDAT	0x040	R/W	0x00000000	GPB Data Register
GPBEN	0x044	R/W	0x00000000	GPB Output Enable Register
GPBSET	0x048	W	-	OR function on GPB Output Data
GPBCLR	0x04C	W	-	BIC function on GPB Output Data
GPBXOR	0x050	W	-	XOR function on GPB Output Data
GPBCD0	0x054	R/W	0x00005555	Driver strength Control 0 on GPB Output Data
GPBCD1	0x058	R/W	0x00005555	Driver strength Control 1 on GPB Output Data
GPBPD0	0x05C	R/W	0x00005555	Pull-Up/Down function on GPB Output Data
GPBPD1	0x060	R/W	0x50555000	Pull-Up/Down function on GPB Output Data
GPBFN0	0x064	R/W	0x00000000	Port Configuration on GPB Output Data
GPBFN1	0x068	R/W	0x00000000	Port Configuration on GPB Output Data
GPBFN2	0x06C	R/W	0x00000000	Port Configuration on GPB Output Data
GPBFN3	0x070	R/W	0x00000000	Port Configuration on GPB Output Data
-	0x074-0x07C			Reserved
GPCDAT	0x080	R/W	0x00000000	GPC Data Register
GPCEN	0x084	R/W	0x00000000	GPC Output Enable Register
GPCSET	0x088	W	-	OR function on GPC Output Data
GPCCLR	0x08C	W	-	BIC function on GPC Output Data
GPCXOR	0x090	W	-	XOR function on GPC Output Data
GPCCD0	0x094	R/W	0x00005555	Driver strength Control 0 on GPC Output Data
GPCCD1	0x098	R/W	0x00005555	Driver strength Control 1 on GPC Output Data
GPCPD0	0x09C	R/W	0x00005555	Pull-Up/Down function on GPC Output Data
GPCPD1	0x0A0	R/W	0x01450550	Pull-Up/Down function on GPC Output Data
GPCFN0	0x0A4	R/W	0x00000000	Port Configuration on GPC Output Data
GPCFN1	0x0A8	R/W	0x00000000	Port Configuration on GPC Output Data
GPCFN2	0x0AC	R/W	0x00000000	Port Configuration on GPC Output Data
GPCFN3	0x0B0	R/W	0x00000000	Port Configuration on GPC Output Data
-	0x0B4-0x0BC			Reserved
GPDDAT	0x0C0	R/W	0x00000000	GPD Data Register
GPDEN	0x0C4	R/W	0x00000000	GPD Output Enable Register
GPDSET	0x0C8	W	-	OR function on GPD Output Data
GPDCLR	0x0CC	W	-	BIC function on GPD Output Data
GPDXOR	0x0D0	W	-	XOR function on GPD Output Data
GPDCD0	0x0D4	R/W	0x00005555	Driver strength Control 0 on GPD Output Data
GPDCD1	0x0D8	R/W	0x00005555	Driver strength Control 1 on GPD Output Data
GPDPD0	0x0DC	R/W	0x00005555	Pull-Up/Down function on GPD Output Data
GPDPD1	0x0E0	R/W	0x00005555	Pull-Up/Down function on GPD Output Data
GPDFN0	0x0E4	R/W	0x00000000	Port Configuration on GPD Output Data
GPDFN1	0x0E8	R/W	0x00000000	Port Configuration on GPD Output Data
GPDFN2	0x0EC	R/W	0x00000000	Port Configuration on GPD Output Data
GPDFN3	0x0F0	R/W	0x00000000	Port Configuration on GPD Output Data
-	0x0F4-0x0FC			Reserved

Name	Offset	Type	Reset	Description
GPEDAT	0x100	R/W	0x00000000	GPE Data Register
GPEEN	0x104	R/W	0x00000000	GPE Output Enable Register
GPESET	0x108	W	-	OR function on GPE Output Data
GPECLR	0x10C	W	-	BIC function on GPE Output Data
GPEXOR	0x110	W	-	XOR function on GPE Output Data
GPECD0	0x114	R/W	0x00005555	Driver strength Control 0 on GPE Output Data
GPECD1	0x118	R/W	0x00005555	Driver strength Control 1 on GPE Output Data
GPEPD0	0x11C	R/W	0x00005555	Pull-Up/Down function on GPE Output Data
GPEPD1	0x120	R/W	0x00005555	Pull-Up/Down function on GPE Output Data
GPEFN0	0x124	R/W	0x00000000	Port Configuration on GPE Output Data
GPEFN1	0x128	R/W	0x00000000	Port Configuration on GPE Output Data
GPEFN2	0x12C	R/W	0x00000000	Port Configuration on GPE Output Data
GPEFN3	0x130	R/W	0x00000000	Port Configuration on GPE Output Data
-	0x134-0x13C			Reserved
GPFDAT	0x140	R/W	0x00000000	GPF Data Register
GPFEN	0x144	R/W	0x00000000	GPF Output Enable Register
GPFSET	0x148	W	-	OR function on GPF Output Data
GPFCLR	0x14C	W	-	BIC function on GPF Output Data
GPFXOR	0x150	W	-	XOR function on GPF Output Data
GPFCD0	0x154	R/W	0x00005555	Driver strength Control 0 on GPF Output Data
GPFCD1	0x158	R/W	0x00005555	Driver strength Control 1 on GPF Output Data
GPFPD0	0x15C	R/W	0x00005555	GPIOF Group has no pull-up/down resistor
GPFPD1	0x160	R/W	0x00005555	GPIOF Group has no pull-up/down resistor
GPFFN0	0x164	R/W	0x00000000	Port Configuration on GPF Output Data
GPFFN1	0x168	R/W	0x00000000	Port Configuration on GP Output Data
GPFFN2	0x16C	R/W	0x00000000	Port Configuration on GPF Output Data
GPFFN3	0x170	R/W	0x00000000	Port Configuration on GPF Output Data
-	0x174-0x17C			Reserved
EINTSEL0	0x184	R/W	0x03020100	External Interrupt Select Register 0
EINTSEL1	0x188	R/W	0x07060504	External Interrupt Select Register 1
EINTSEL2	0x18C	R/W	0x0B0A0908	External Interrupt Select Register 2
MON	0x190	R/W	0x00000000	System Monitor Enable Register
ECID0	0x194	R/W	0x00000000	CID output Register
ECID1	0x198	R	-	CID serial output data Register
ECID2	0x19C	R	-	CID parallel output data 0 Register
ECID3	0x1A0	R	-	CID parallel output data 1 Register

GPIO Data Register (GPADAT, GPBDAT, GPCDAT, GPDDAT, GPEDAT, GPFDAT)

0xF0102000, 0xF0102040, 0xF0102080, 0xF01020C0, 0xF0102100, 0xF0102140

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPDAT															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPDAT															

If writing to this register, the written data would be stored in the output buffer. And the direction of corresponding GPIO is the output, the stored data would be out to pin. In case of reading this register, if the direction of the corresponding GPIO is out, the data read is output buffer, otherwise (input mode) the data read is the status of the corresponding pin

GPIO Direction Control Register (GPAEN, GPBEN, GPCEN, GPDEN, GPEEN, GPFEN)

0xF0102004, 0xF0102044, 0xF0102084, 0xF01020C4, 0xF0102104, 0xF0102144

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPEN															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPEN															

If the GPEN[i] is ‘1’(i = 0 ~ 31), the direction of the corresponding bit of GPIO port is changed to output mode, otherwise to input mode.

GPIO Set Register (GPASET, GPBSET, GPCSET, GPDSET, GPESET, GPFSET,)

0xF0102008, 0xF0102048, 0xF0102088, 0xF01020C8, 0xF0102108, 0xF0102148

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPSET															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPSET															

The equivalent function is GPADAT = GPADAT | GPSET, when GPADAT is the value of output buffer.

GPIO Clear Register (GPACLR, GPBCLR, GPCCLR, GPDCLR, GPECLR, GPFCLR)

0xF010200C, 0xF010204C, 0xF010208C, 0xF01020CC, 0xF010210C, 0xF010214C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPCLR															

The equivalent function is GPADAT = GPADAT & ~GPCLR, when GPADAT is the value of output buffer.

GPIO XOR Register (GPAXOR, GPBXOR, GPCXOR, GPDXOR GPEXOR, GPFXOR)

0xF0102010, 0xF0102050, 0xF0102090, 0xF01020D0, 0xF0102110, 0xF0102150

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPXOR															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPXOR															

The equivalent function is GPADAT = GPADAT ^ GPXOR, when GPADAT is the value of output buffer.

Driver Strength Control Register 0(GPACD0, GPBCD0, GPCCD0, GPDCD0, GPECD0, GPFCD0)

0xF0102014, 0xF0102054, 0xF0102094, 0xF01020D4, 0xF0102114, 0xF0102154

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPCD15		GPCD14		GPCD13		GPCD12		GPCD11		GPCD10		GPCD9		GPCD8	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPCD7		GPCD6		GPCD5		GPCD4		GPCD3		GPCD2		GPCD1		GPCD0	

The TCC8900 has the function of programmable port driver strength. The control value for driver strength of the corresponding port can become 0~3 and 3 is the strongest value.

Driver Strength Control Register 1(GPACD1, GPBCD1, GPCCD1, GPDCD1, GPECD1, GPFCD1)

0xF0102018, 0xF0102058, 0xF0102098, 0xF01020D8, 0xF0102118, 0xF0102158

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPCD31		GPCD130		GPCD29		GPCD28		GPCD27		GPCD26		GPCD25		GPCD24	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPCD23		GPCD22		GPCD21		GPCD20		GPCD19		GPCD18		GPCD17		GPCD16	

The TCC8900 has the function of programmable port driver strength. The control value for driver strength of the corresponding port can become 0~3 and 3 is the strongest value.

Pull UP/DOWN Control Register 0(GPAPD0, GPBPD0, GPCPD0, GPDPD0, GPEPD0)

0xF010201C, 0xF010205C, 0xF010209C, 0xF01020DC, 0xF010211C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PD15	PU15	PD14	PU14	PD13	PU13	PD12	PU12	PD11	PU11	PD10	PU10	PD9	PU9	PD8	PU8
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PD7	PU7	PD6	PU6	PD5	PU5	PD4	PU4	PD3	PU3	PD2	PU2	PD1	PU1	PD0	PU0

Each field of pull up/down control registers is the same with corresponding port name and consists of two bits. When PUx is “1”, pull-up will be enabled, and when PDx is “1”, pull-down will be enabled. It value should be not written with both pull-up and pull-down are enabled.

Pull UP/DOWN Control Register 1(GPAPD1, GPBPD1, GPCPD1, GPDPD1, GPEPD1)

0xF0102020, 0xF0102060, 0xF01020A0, 0xF01020E0, 0xF0102120

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PD31	PU31	PD30	PU30	PD29	PU29	PD28	PU28	PD27	PU27	PD26	PU26	P25	PU25	PD24	PU24
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PD23	PU23	PD22	PU22	PD21	PU21	PD20	PU20	PD19	PU19	PD18	PU18	PD17	PU17	PD16	PU16

Each field of pull up/down control registers is the same with corresponding port name and consists of two bits. When PUx is “1”, pull-up will be enabled, and when PDx is “1”, pull-down will be enabled. It value should be not written with both pull-up and pull-down are enabled.

Port Configuration Register 0 (GPAFN0, GPBFN0, GPCFN0, GPDFN0, GPEFN0, GPFFN0)

0xF0102024, 0xF0102064, 0xF01020A4, 0xF01020E4, 0xF0102124, 0xF0102164

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPFN7				GPFN6				GPFN5				GPFN4			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPFN3				GPFN2				GPFN1				GPFN0			

Port Configuration Register 1 (GPAFN1, GPBFN1, GPCFN1, GPDFN1, GPEFN1, GPFFN1)

0xF0102028, 0xF0102068, 0xF01020A8, 0xF01020E8, 0xF0102128, 0xF0102168

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPFN15				GPFN14				GPFN13				GPFN12			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPFN11				GPFN10				GPFN9				GPFN8			

Port Configuration Register 2 (GPAFN2, GPBFN2, GPCFN2, GPDFN2, GPEFN2, GPFFN2)

0xF010202C, 0xF010206C, 0xF01020AC, 0xF01020EC, 0xF010212C, 0xF010216C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPFN23				GPFN22				GPFN21				GPFN20			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPFN19				GPFN18				GPFN17				GPFN16			

Port Configuration Register 3 (GPAFN3, GPBFN3, GPCFN3, GPDFN3, GPEFN3, GPFFN3)

0xF0102030, 0xF0102070, 0xF01020B0, 0xF01020F0, 0xF0102130, 0xF0102170

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPFN31				GPFN30				GPFN29				GPFN28			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPFN27				GPFN26				GPFN25				GPFN24			

Table 1.2 Port Configuration of GPIOA

Name	Function						
	0	1	2	3	4	5	6
GPIOA[0]	GPIOA[0]	SCL0					
GPIOA[1]	GPIOA[1]	SDA0					
GPIOA[2]	GPIOA[2]	CLK_OUT0					
GPIOA[3]	GPIOA[3]	CLK_OUT1					
GPIOA[4]	GPIOA[4]	WDTRSTO	TCO0				
GPIOA[5]	GPIOA[5]	IRDI	TCO1				
GPIOA[6]	GPIOA[6]	HDMI_CECO	TCO2				EDIXA[19]
GPIOA[7]	GPIOA[7]	HDMI_CECI	TCO3				EDIXA[20]
GPIOA[8]	GPIOA[8]	SCL1					
GPIOA[9]	GPIOA[9]	SDA1					
GPIOA[10]	GPIOA[10]	CBCLK(0)	CBCLK(1)				
GPIOA[11]	GPIOA[11]	CLRCK(0)	CLRCK(1)				
GPIOA[12]	GPIOA[12]	CDATA(0)	CDATA(1)				
GPIOA[13]	GPIOA[13]	EXTCLKI					
GPIOA[14]	GPIOA[14]	HDMI_HPD	TCO4				
GPIOA[15]	GPIOA[15]	UTM_DRVVBUS	TCO5				

FUNCTION	DESCRIPTION
GPIOA[15:0]	GPIO A Group Ports
SCL0, SDA0	I2C Channel 0
SCL1, SDA1	I2C Channel 1
CLK_OUT0, CLK_OUT1	External Clock Outputs Generated by the CKC.
WDTRSTO	Watchdog Reset Output
IRDI	Remocon Receiver Port
HDMI_CECO, HDMI_CECI, HDMI_HPD	HDMI Option Ports
CBCLK(0), CLRCK(0), CDATA(0)	CD Interface Controller 0
CBCLK(1), CLRCK(1), CDATA(1)	CD Interface Controller 1
UTM_DRVVBUS	USB OTG Interface Port
TCO0 ~ TCO5	Timer Counter Output from Timer
EDIXA[20:19]	EDI Interface Ports

Table 1.3 Port Configuration of GPIOB

Name	Function						
	0	1	2	3	4	5	6
GPIOB[0]	GPIOB[0]	EDIXD8	SD_D0(5)	MS_D0(5)			
GPIOB[1]	GPIOB[1]	EDIXD9	SD_D1(5)	MS_D1(5)			
GPIOB[2]	GPIOB[2]	EDIXD10	SD_D2(5)	MS_D2(5)			
GPIOB[3]	GPIOB[3]	EDIXD11	SD_D3(5)	MS_D3(5)			
GPIOB[4]	GPIOB[4]	EDIXD4	SD_D4(5)	MS_D4(5)			
GPIOB[5]	GPIOB[5]	EDIXD5	SD_D5(5)	MS_D5(5)			
GPIOB[6]	GPIOB[6]	EDIXD6	SD_D6(5)	MS_D6(5)			
GPIOB[7]	GPIOB[7]	EDIXD7	SD_D7(5)	MS_D7(5)			
GPIOB[8]	GPIOB[8]	EDIXD0			SFRM(1)		
GPIOB[9]	GPIOB[9]	EDIXD1			SCLK(1)		
GPIOB[10]	GPIOB[10]	EDIXD2			SDI(1)		
GPIOB[11]	GPIOB[11]	EDIXD3			SDO(1)		
GPIOB[12]	GPIOB[12]	EDIXD12	SD_CMD(5)	MS_BUS(5)			
GPIOB[13]	GPIOB[13]	EDIXD13	SD_CLK(5)	MS_CLK(5)			
GPIOB[14]	GPIOB[14]	EDIXD14					
GPIOB[15]	GPIOB[15]	EDIXD15					
GPIOB[16]	GPIOB[16]	EDIWEN0					
GPIOB[17]	GPIOB[17]	EDIWEN1[1]			SFRM(0)		
GPIOB[18]	GPIOB[18]	EDIOEN0					
GPIOB[19]	GPIOB[19]	EDIOEN1[2]			SCLK(0)		
GPIOB[20]	GPIOB[20]	EDIXA[0][3]					
GPIOB[21]	GPIOB[21]	EDIXA[1][4]	SD_D4(6)	MS_D4(6)			
GPIOB[22]	GPIOB[22]	EDIXA[2]	SD_D5(6)	MS_D5(6)			
GPIOB[23]	GPIOB[23]	EDICSN0[5]	SD_D6(6)	MS_D6(6)			
GPIOB[24]	GPIOB[24]	EDICSN1	SD_D7(6)	MS_D7(6)			
GPIOB[25]	GPIOB[25]	EDICSN2	SD_D0(6)	MS_D0(6)			
GPIOB[26]	GPIOB[26]	EDICSN3	SD_D1(6)	MS_D1(6)			
GPIOB[27]	GPIOB[27]	EDICSN4	SD_D2(6)	MS_D2(6)			EDIXA[23]
GPIOB[28]	GPIOB[28]	EDIRDY0[6]	SD_D3(6)	MS_D3(6)			
GPIOB[29]	GPIOB[29]	EDIRDY1	SD_CMD(6)	MS_BUS(6)			
GPIOB[30]	GPIOB[30]	EDICSN5	SD_CLK(6)	MS_CLK(6)	SDI(0)		EDIXA[22]
GPIOB[31]	GPIOB[31]	EDICSN6			SDO(0)		EDIXA[21]

FUNCTION	DESCRIPTION
GPIOB[31:0]	GPIO B Group Ports
SD_CMD(5), SD_CLK(5), SD_D0(5) ~ SD_D7(5)	SD/MMC Interface Port 5
MS_BUS(5), MS_CLK(5), MS_D0(5) ~ MS_D7(5)	Memory Stick Interface Port 5
SD_CMD(6), SD_CLK(6), SD_D0(6) ~ SD_D7(6)	SD/MMC Interface Port 6
MS_BUS(6), MS_CLK(6), MS_D0(6) ~ MS_D7(6)	Memory Stick Interface Port 6
SFRM(0), SCLK(0), SDI(0), SDO(0)	GPSB Interface Port 0
SFRM(1), SCLK(1), SDI(1), SDO(1)	GPSB Interface Port 1
EDIXA[23:21]	EDI Interface Ports

[1] Dedicated to NAND WEN.
[2] Dedicated to NAND OEN.
[3] Dedicated to NAND CLE.
[4] Dedicated to NAND ALE.
[5] Dedicated to NAND CSN0 for NAND Booting.
If NAND boot is not used, the EDICSN0 can be mapped to another device.
[6] Dedicated to NAND Ready signal for NAND booting,
Either EDIRDY0 or GPIOB[31] can be used for NAND ready signal as described in the BOOT PROCEDURE in datasheet

Table 1.4 Port Configuration of GPIOC

Name	Function						
	0	1	2	3	4	5	6
GPIOC[0]	GPIOC[0]	LXD[0]	LCD0_LPD[0]	TSD5(3)		LCD1_LPD[0]	GPIOF[0]
GPIOC[1]	GPIOC[1]	LXD[1]	LCD0_LPD[1]	TSD6(3)		LCD1_LPD[1]	GPIOF[1]
GPIOC[2]	GPIOC[2]	LXD[2]	LCD0_LPD[2]	TSD7(3)		LCD1_LPD[2]	GPIOF[2]
GPIOC[3]	GPIOC[3]	LXD[3]	LCD0_LPD[3]			LCD1_LPD[3]	GPIOF[3]
GPIOC[4]	GPIOC[4]	LXD[4]	LCD0_LPD[4]			LCD1_LPD[4]	GPIOF[4]
GPIOC[5]	GPIOC[5]	LXD[5]	LCD0_LPD[5]			LCD1_LPD[5]	GPIOF[5]
GPIOC[6]	GPIOC[6]	LXD[6]	LCD0_LPD[6]	SDO(3)		LCD1_LPD[6]	GPIOF[6]
GPIOC[7]	GPIOC[7]	LXD[7]	LCD0_LPD[7]	SDI(3)		LCD1_LPD[7]	GPIOF[7]
GPIOC[8]	GPIOC[8]	LXD[8]	LCD0_LPD[8]	SCLK(3)		LCD1_LPD[8]	GPIOF[8]
GPIOC[9]	GPIOC[9]	LXD[9]	LCD0_LPD[9]	SFRM(3)		LCD1_LPD[9]	GPIOF[9]
GPIOC[10]	GPIOC[10]	LXD[10]	LCD0_LPD[10]	SDO(2)		LCD1_LPD[10]	GPIOF[10]
GPIOC[11]	GPIOC[11]	LXD[11]	LCD0_LPD[11]	SDI(2)		LCD1_LPD[11]	GPIOF[11]
GPIOC[12]	GPIOC[12]	LXD[12]	LCD0_LPD[12]	SCLK(2)		LCD1_LPD[12]	GPIOF[12]
GPIOC[13]	GPIOC[13]	LXD[13]	LCD0_LPD[13]	SFRM(2)		LCD1_LPD[13]	GPIOF[13]
GPIOC[14]	GPIOC[14]	LXD[14]	LCD0_LPD[14]	SD_D7(0)	MS_D7(0)	LCD1_LPD[14]	GPIOF[14]
GPIOC[15]	GPIOC[15]	LXD[15]	LCD0_LPD[15]	SD_D6(0)	MS_D6(0)	LCD1_LPD[15]	GPIOF[15]
GPIOC[16]	GPIOC[16]	LXD[16]	LCD0_LPD[16]	SD_D5(0)	MS_D5(0)	LCD1_LPD[16]	GPIOF[16]
GPIOC[17]	GPIOC[17]	LXD[17]	LCD0_LPD[17]	SD_D4(0)	MS_D4(0)	LCD1_LPD[17]	GPIOF[17]
GPIOC[18]	GPIOC[18]		LCD0_LPD[18]	SD_D3(0)	MS_D3(0)	LCD1_LPD[18]	
GPIOC[19]	GPIOC[19]		LCD0_LPD[19]	SD_D2(0)	MS_D2(0)	LCD1_LPD[19]	
GPIOC[20]	GPIOC[20]		LCD0_LPD[20]	SD_D1(0)	MS_D1(0)	LCD1_LPD[20]	
GPIOC[21]	GPIOC[21]		LCD0_LPD[21]	SD_D0(0)	MS_D0(0)	LCD1_LPD[21]	
GPIOC[22]	GPIOC[22]		LCD0_LPD[22]	SD_CLK(0)	MS_CLK(0)	LCD1_LPD[22]	
GPIOC[23]	GPIOC[23]		LCD0_LPD[23]	SD_CMD(0)	MS_BUS(0)	LCD1_LPD[23]	
GPIOC[24]	GPIOC[24]	LWEN	LCD0_LDE	TSD4(3)		LCD1_LDE	GPIOF[19]
GPIOC[25]	GPIOC[25]	LOEN	LCD0_LCK	TSD3(3)		LCD1_LCK	GPIOF[18]
GPIOC[26]	GPIOC[26]	LXA[0]	LCD0_LHS	TSD2(3)		LCD1_LHS	GPIOF[22]
GPIOC[27]	GPIOC[27]	LCSN0	LCD0_LVS	TSD1(3)		LCD1_LVS	GPIOF[20]
GPIOC[28]	GPIOC[28]	LCSN1	SDO(10)	TSVALID(3)			GPIOF[21]
GPIOC[29]	GPIOC[29]		SDI(10)	TSCLK(3)			
GPIOC[30]	GPIOC[30]	EXTLVS0(0)	SCLK(10)	TSD0(3)			
GPIOC[31]	GPIOC[31]	EXTLVS1(0)	SFRM(10)	TSSYNC(3)			

FUNCTION	DESCRIPTION
GPIOC[31:0]	GPIO C Group Ports
GPIOF[22:0]	GPIO F Signals – In this case, the GPIO C Group is Output Mode and Functionally the same as the bypass by BPEN.
LCD0_LPD[23:0], LCD0_LDE, LCD0_LCK, LCD0_LHS, LCD0_LVS	LCD RGB Interface Ports from LCD Controller 0
LCD1_LPD[23:0], LCD1_LDE, LCD1_LCK, LCD1_LHS, LCD1_LVS	LCD RGB Interface Ports from LCD Controller 1
LXD[17:0], LWEN, LOEN, LXA[0], LCSN0, LCSN1	LCD CPU Interface Ports Which Can Be Shared By LCD Controller 0, 1 and CPU.
EXTLVS0(0)	External VSYNC Input Port 0 for LCD Controller 0
EXTLVS1(0)	External VSYNC Input Port 1 for LCD Controller 1
SFRM(10), SCLK(10), SDI(10), SDO(10)	GPSB Interface Port 10
SFRM(3), SCLK(3), SDI(3), SDO(3)	GPSB Interface Port 3
SFRM(2), SCLK(2), SDI(2), SDO(2)	GPSB Interface Port 2
SD_CMD(0), SD_CLK(0), SD_D0(0) ~ SD_D7(0)	SD/MMC Interface Port 0
MS_BUS(0), MS_CLK(0), MS_D0(0) ~ MS_D7(0)	Memory Stick Interface Port 0
TSSYNC(3), TSCLK(3), TSVALID(3), TSD0(3) ~ TSD7(3)	TS Parallel Interface Port 3

Table 1.5 Port Configuration of GPIOD

Name	Function						
	0	1	2	3	4	5	6
GPIOD[0]	GPIOD[0]	BCLK(1)	BCLK(0)				
GPIOD[1]	GPIOD[1]	LRCK(1)	LRCK(0)				
GPIOD[2]	GPIOD[2]	MCLK(1)	MCLK(0)				
GPIOD[3]	GPIOD[3]	DAO0(1)	DAO0(0)				
GPIOD[4]	GPIOD[4]	DAI0(1)	DAI0(0)				
GPIOD[5]	GPIOD[5]	DAO1(1)	SFRM(11)				
GPIOD[6]	GPIOD[6]	DAI1(1)	SCLK(11)				
GPIOD[7]	GPIOD[7]	DAO2(1)	SDI(11)				
GPIOD[8]	GPIOD[8]	DAI2(1)	SDO(11)				
GPIOD[9]	GPIOD[9]	DAO3(1)	SFRM(6)	TSD7(2)			
GPIOD[10]	GPIOD[10]	DAI3(1)	SCLK(6)	TSD6(2)			
GPIOD[11]	GPIOD[11]	SPD_TX(1)	SDI(6)	SPD_TX(0)			
GPIOD[12]	GPIOD[12]	SPD_RX(1)	SDO(6)	TSSYNC(2)			
GPIOD[13]	GPIOD[13]	UTXD(4)		TSD5(2)			
GPIOD[14]	GPIOD[14]	URXD(4)		TSD4(2)			
GPIOD[15]	GPIOD[15]	UCTS(4)	SFRM(12)	TSVALID(2)			
GPIOD[16]	GPIOD[16]	URTS(4)	SCLK(12)	TSCLK(2)			
GPIOD[17]	GPIOD[17]	UTXD(5)		TSD3(2)			
GPIOD[18]	GPIOD[18]	URXD(5)		TSD2(2)			
GPIOD[19]	GPIOD[19]	UCTS(5)	SDI(12)	TSD1(2)			
GPIOD[20]	GPIOD[20]	URTS(5)	SDO(12)	TSD0(2)			
GPIOD[21]	GPIOD[21]						
GPIOD[22]	GPIOD[22]						
GPIOD[23]	GPIOD[23]						
GPIOD[24]	GPIOD[24]						
GPIOD[25]	GPIOD[25]						

FUNCTION	DESCRIPTION
GPIOD[25:0]	GPIO D Group Ports
BCLK(1), LRCK(1), MCLK(1), DAO0(1) ~ DAO3(1), DAI0(1) ~ DAI3(1)	I2S Controller 1 – Multiple channel with DMA
BCLK(0), LRCK(0), MCLK(0), DAO0(0), DAI0(0)	I2S Controller 0 – Single channel without DMA
SFRM(11), SCLK(11), SDI(11), SDO(11)	GPSB Interface Port 11
SFRM(12), SCLK(12), SDI(12), SDO(12)	GPSB Interface Port 12
SFRM(6), SCLK(6), SDI(6), SDO(6)	GPSB Interface Port 6
SPD_TX(1), SPD_RX(1)	SPDIF Transmit/Receiver Controller 1
SPD_TX(0)	SPDIF Transmitter Controller 0
UTXD(4), URXD(4), UCTS(4), URTS(4)	UART Interface Port 4
UTXD(5), URXD(5), UCTS(5), URTS(5)	UART Interface Port 5
TSSYNC(2), TSVALID(2), TSCLK(2), TSD0(2) ~ TSD7(2)	TS Parallel Interface Port 2

Table 1.6 Port Configuration of GPIOE

Name	Function						
	0	1	2	3	4	5	6
GPIOE[0]	GPIOE[0]	UTXD(0)					
GPIOE[1]	GPIOE[1]	URXD(0)					
GPIOE[2]	GPIOE[2]	UCTS(0)	SFRM(5)				
GPIOE[3]	GPIOE[3]	URTS(0)	SCLK(5)				
GPIOE[4]	GPIOE[4]	UTXD(1)					
GPIOE[5]	GPIOE[5]	URXD(1)					
GPIOE[6]	GPIOE[6]	UCTS(1)	SDI(5)	SD_CLK(4)	MS_CLK(4)		
GPIOE[7]	GPIOE[7]	URTS(1)	SDO(5)	SD_CMD(4)	MS_BUS(4)		
GPIOE[8]	GPIOE[8]	UTXD(2)	SFRM(4)	SD_D0(4)	MS_D0(4)		
GPIOE[9]	GPIOE[9]	URXD(2)	SCLK(4)	SD_D1(4)	MS_D1(4)		
GPIOE[10]	GPIOE[10]	UTXD(3)	SDI(4)	SD_D2(4)	MS_D2(4)		
GPIOE[11]	GPIOE[11]	URXD(3)	SDO(4)	SD_D3(4)	MS_D3(4)		
GPIOE[12]	GPIOE[12]	CPD[0]	SD_D0(2)	TSD0(1)	MS_D0(2)		
GPIOE[13]	GPIOE[13]	CPD[1]	SD_D1(2)	TSD1(1)	MS_D1(2)		
GPIOE[14]	GPIOE[14]	CPD[2]	SD_D2(2)	TSD2(1)	MS_D2(2)		
GPIOE[15]	GPIOE[15]	CPD[3]	SD_D3(2)	TSD3(1)	MS_D3(2)		
GPIOE[16]	GPIOE[16]	CPD[4]	SD_D4(2)	TSD4(1)	MS_D4(2)		
GPIOE[17]	GPIOE[17]	CPD[5]	SD_D5(2)	TSD5(1)	MS_D5(2)		
GPIOE[18]	GPIOE[18]	CPD[6]	SD_D6(2)	TSD6(1)	MS_D6(2)		
GPIOE[19]	GPIOE[19]	CPD[7]	SD_D7(2)	TSD7(1)	MS_D7(2)		
GPIOE[20]	GPIOE[20]	CCKI	SD_CLK(2)	TSCLK(1)	MS_CLK(2)		
GPIOE[21]	GPIOE[21]	CVS	SD_CMD(2)	TSSYNC(1)	MS_BUS(2)		
GPIOE[22]	GPIOE[22]	CHS		TSVALID(1)			
GPIOE[23]	GPIOE[23]	CCKO	CFIELD				
AIN[0]	GPIOE[24]	AIN[0]					
AIN[1]	GPIOE[25]	AIN[1]					
AIN[2]	GPIOE[26]	AIN[2]	SD_CMD(7)	MS_BUS(7)			
AIN[3]	GPIOE[27]	AIN[3]	SD_CLK(7)	MS_CLK(7)			
AIN[4]	GPIOE[28]	TSC_YM	SD_D0(7)	MS_D0(7)			
AIN[5]	GPIOE[29]	TSC_YP	SD_D1(7)	MS_D1(7)			
AIN[6]	GPIOE[30]	TSC_XM	SD_D2(7)	MS_D2(7)			
AIN[7]	GPIOE[31]	TSC_XP	SD_D3(7)	MS_D3(7)			

FUNCTION	DESCRIPTION
GPIOE[31:0]	GPIO E Group Ports
AIN[3:0]	ADC Input Ports
TSC_YP, TSC_YM, TSC_XM, TSC_XP	Touch Screen Controller Interface These can be used as AIN[7:4] with ADC inputs.
UTXD(0), URXD(0), UCTS(0), URTS(0)	UART Interface Port 0
UTXD(1), URXD(1), UCTS(1), URTS(1)	UART Interface Port 1
UTXD(2), URXD(2), UCTS(2), URTS(2)	UART Interface Port 2
UTXD(3), URXD(3), UCTS(3), URTS(3)	UART Interface Port 3
CPD[7:0], CCKI, CVS, CHS, CCKO, CFIELD	Camera or Video Input Interface Ports
SFRM(5), SCLK(5), SDI(5), SDO(5)	GPSB Interface Port 5
SFRM(4), SCLK(4), SDI(4), SDO(4)	GPSB Interface Port 4
SD_CMD(2), SD_CLK(2), SD_D0(2) ~ SD_D7(2)	SD/MMC Interface Port 2
SD_CMD(7), SD_CLK(7), SD_D0(7) ~ SD_D3(7)	SD/MMC Interface Port 7
SD_CMD(4), SD_CLK(4), SD_D0(4) ~ SD_D3(4)	SD/MMC Interface Port 4
MS_BUS(4), MS_CLK(4), MS_D0(4) ~ MS_D3(4)	Memory Stick Interface Port 4
MS_BUS(2), MS_CLK(2), MS_D0(2) ~ MS_D7(2)	Memory Stick Interface Port 2
MS_BUS(7), MS_CLK(7), MS_D0(7) ~ MS_D3(7)	Memory Stick Interface Port 7
TSSYNC(1), TSVALID(1), TSCLK(1), TSD0(1) ~ TSD7(1)	TS Parallel Interface Port 1

Table 1.7 Port Configuration of GPIOF

Name	Function						
	0	1	2	3	4	5	6
GPIOF[0]	GPIOF[0]	HPXD[0]	SD_D0(3)	HDDXD0	TSD0(0)	MS_D0(3)	EDIXA[3]
GPIOF[1]	GPIOF[1]	HPXD[1]	SD_D1(3)	HDDXD1	TSD1(0)	MS_D1(3)	EDIXA[4]
GPIOF[2]	GPIOF[2]	HPXD[2]	SD_D2(3)	HDDXD2	TSD2(0)	MS_D2(3)	EDIXA[5]
GPIOF[3]	GPIOF[3]	HPXD[3]	SD_D3(3)	HDDXD3	TSD3(0)	MS_D3(3)	EDIXA[6]
GPIOF[4]	GPIOF[4]	HPXD[4]	SD_D4(3)	HDDXD4	TSD4(0)	MS_D4(3)	EDIXA[7]
GPIOF[5]	GPIOF[5]	HPXD[5]	SD_D5(3)	HDDXD5	TSD5(0)	MS_D5(3)	EDIXA[8]
GPIOF[6]	GPIOF[6]	HPXD[6]	SD_D6(3)	HDDXD6	TSD6(0)	MS_D6(3)	EDIXA[9]
GPIOF[7]	GPIOF[7]	HPXD[7]	SD_D7(3)	HDDXD7	TSD7(0)	MS_D7(3)	EDIXA[10]
GPIOF[8]	GPIOF[8]	HPXD[8]	SD_CMD(3)	HDDXD8	TSVALID(0)	MS_BUS(3)	EDIXA[11]
GPIOF[9]	GPIOF[9]	HPXD[9]	SD_CLK(3)	HDDXD9	TSCLK(0)	MS_CLK(3)	EDIXA[12]
GPIOF[10]	GPIOF[10]	HPXD[10]	SDO(7)	HDDXD10	TSSYNC(0)		EDIXA[13]
GPIOF[11]	GPIOF[11]	HPXD[11]	SDI(7)	HDDXD11			EDIXA[14]
GPIOF[12]	GPIOF[12]	HPXD[12]	SCLK(7)	HDDXD12			EDIXA[15]
GPIOF[13]	GPIOF[13]	HPXD[13]	SFRM(7)	HDDXD13			EDIXA[16]
GPIOF[14]	GPIOF[14]	HPXD[14]	SDO(8)	HDDXD14			EDIXA[17]
GPIOF[15]	GPIOF[15]	HPXD[15]	SDI(8)	HDDXD15			EDIXA[18]
GPIOF[16]	GPIOF[16]	HPXD[16]	SCLK(8)	HDDXA0			
GPIOF[17]	GPIOF[17]	HPXD[17]	SFRM(8)	HDDXA1			
GPIOF[18]	GPIOF[18]	HPRDN	SD_D3(1)	HDDXA2	MS_D3(1)		
GPIOF[19]	GPIOF[19]	HPWRN	SD_D2(1)	HDDCSN1	MS_D2(1)		
GPIOF[20]	GPIOF[20]	HPCSN0	SD_D1(1)	HDDRDY	MS_D1(1)		
GPIOF[21]	GPIOF[21]	HPCSN1	SD_D0(1)	HDDCSN0	MS_D0(1)		
GPIOF[22]	GPIOF[22]	HPXA	SD_CMD(1)	HDDAK	MS_BUS(1)		
GPIOF[23]	GPIOF[23]	HPINT0	SD_CLK(1)	HDDRQ	MS_CLK(1)		
GPIOF[24]	GPIOF[24]	HPINT1	SDO(9)	HDDIOW			
GPIOF[25]	GPIOF[25]		SDI(9)	HDDIOR			
GPIOF[26]	GPIOF[26]	EXTLVS1(1)	SCLK(9)	CAN_RX			
GPIOF[27]	GPIOF[27]	EXTLVS0(1)	SFRM(9)	CAN_TX			

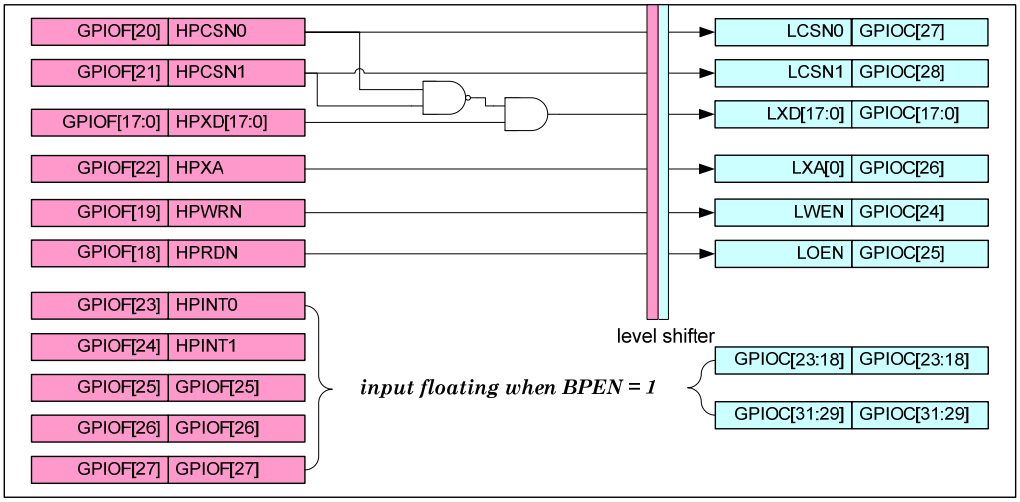
FUNCTION	DESCRIPTION
GPIOF[27:0]	GPIO F Group Ports
HPXD[17:0], HPRDN, HPWRN, HPCSN0, HPCSN1, HPINT0, HPINT1	External Host Interface Ports
EXTLVS1(1)	External VSYNC Port 1 to LCD Controller 1
EXTLVS0(1)	External VSYNC Port 1 to LCD Controller 0
SD_CMD(3), SD_CLK(3), SD_D0(3) ~ SD_D7(3)	SD/MMC Interface Port 3
SD_CMD(1), SD_CLK(1), SD_D0(1) ~ SD_D3(1)	SD/MMC Interface Port 1
SFRM(7), SCLK(7), SDI(7), SDO(7)	GPSB Interface Port 7
SFRM(8), SCLK(8), SDI(8), SDO(8)	GPSB Interface Port 8
SFRM(9), SCLK(9), SDI(9), SDO(9)	GPSB Interface Port 9
HDDXD0 ~ HDDXD15, HDDXA0 ~ HDDXA2, HDDAK, HDDRQ, HDDRDY, HDDCSN0, HDDCSN1, HDDIOW, HDDIOR	UDMA Interface Ports
CAN_TX, CAN_RX	CAN Interface Ports
MS_BUS(3), MS_CLK(3), MS_D0(3) ~ MS_D7(3)	Memory Stick Interface Port 3
MS_BUS(1), MS_CLK(1), MS_D0(1) ~ MS_D3(1)	Memory Stick Interface Port 1
TSSYNC(0), TSVALID(0), TSCLK(0), TSD0(0) ~ TSD7(0)	TS Parallel Interface Port 0
EDIXA[18:3]	EDI Interface Ports

1.3 Bypass Function

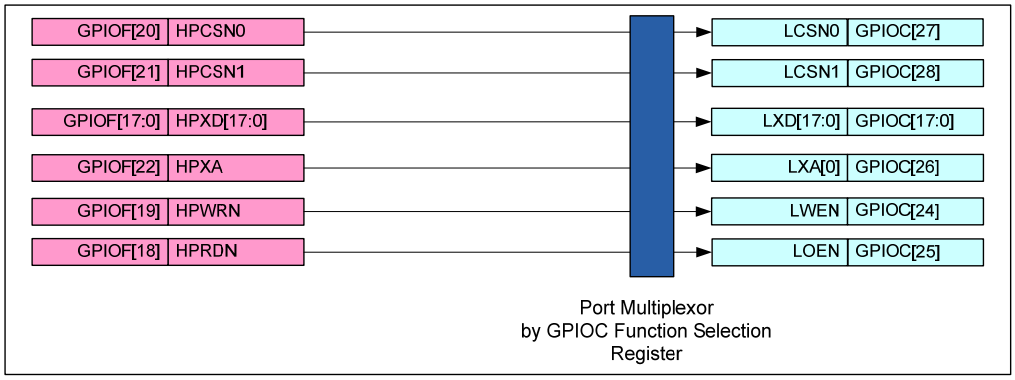
The TCC8900 supports the LCD bypass mode. In this mode, the external host device which is connected to the TCC8900 EHI ports directly controls the LCD module which is connected to the TCC8900 LCD ports. **For this mode, the TCC8900 has two ways. One is to use the external pin, which is BPEN. The other is to use the on-chip control register**.

When BPEN pin is logically high level, the TCC8900 enters the LCD bypass mode. The (a) of the following figure shows the relationship between EHI ports and LCD ports while BPEN=1. In this case, a state of the TCC8900 can be “power off” state except for I/O power of EHI and LCD ports..

When BPEN pin is logically low level, the function selection registers of the on-chip are used to enable the LCD bypass mode. The (b) of the following figure shows the relationship between EHI ports and LCD ports. Naturally, a state of the TCC8900 should be “power on” state in this case.



(a) Bypass when BPEN = 1, TCC8900 can be powered off except for I/O powers of EHI and LCD ports



(b) Bypass when BPEN = 0, TCC8900 should be on normal mode, core power should be turned-on.

Figure 1.1 Bypass Mode in the TCC8900

PART4 – CORE & MEMORY BUS

TCC8900

High Performance and Low-Power Processor

For Digital Media Applications

Rev. 1.02

Nov 06, 2009

Telechips

Revision History

Date	Revision	Description
2008-12-11	0.00	* Initial Release
2009-02-10	0.01	* The "Memory Bus Configuration Register Map" has been changed. * The description of the "A_GT_M_SYNC" and "SYNC" field in the Memory_cfg2 register has been modified * The description for the AXI ID and corresponding hardware has been added. * The "QOS_MBITS" field of the memory controller has been changed.
2009-02-25	0.02	* The “M0CFG0” and “M0CFG1” fields are duplicated. The laters were changed to “M1CFG0” and “M1CFG1” accordingly. * The description for the “TZPROT” register was added. * The address for the CKDOWN was changed from “0xf0305000” to “0xf0305004”. * The base address for the CORECFG register was changed from “0xF0050000” to “0xF0101000”. * The base address for the Virtual MMU Table was changed from “0xF7000000” to “0xF0600000”.
2009-08-07	1.00	* Change bit field description of memory_cfg2 register in DDR2 SDRAM controller
2009-10-09	1.01	* Bit field of “t_rcd Register” has been changed.
2009-11-06	1.02	* bit[14:13] of SDRAM_IOCON (0xf030302) is modified to IO_MODE and GATE_PD.

TABLE OF CONTENTS

Contents

1 Core & Memory Bus Architecture 1-1
1.1 Architecture Overview 1-1
1.2 Address Map 1-2
2 ARM1176JZ(F)-S Main Processor 2-3
2.1 Overview 2-3
2.2 Functional Description 2-3
2.2.1 Memory Formats 2-3
2.2.2 Operating Modes 2-3
2.2.3 Instruction Set Features 2-4
2.2.3.1 ARM Instruction Set 2-4
2.2.4 Addressing Modes 2-4
2.2.4.1 Load/Store Instructions 2-4
2.2.4.2 Multiple Load/Store Instructions 2-4
2.2.5 Exceptions 2-4
2.2.5.1 Actions on Entering an Exception 2-5
2.2.5.2 Actions on Leaving an Exception 2-5
2.2.5.3 Exception Entry/Exit Summary 2-5
2.2.5.4 FIQ 2-5
2.2.5.5 IRQ 2-6
2.2.5.6 Abort 2-6
2.2.5.7 Software Interrupt 2-6
2.2.5.8 Undefined Instruction 2-6
2.3 Coprocessor CP15 2-6
3 DRAM Controller 3-7
3.1 Overview 3-7
3.1.1.1 SDR/DDR SDRAM Controller 3-7
3.1.1.2 DDR2 SDRAM Controller 3-7
3.2 Functional Overview 3-8
3.2.1 Exclusive access 3-8
3.2.2 Controller management operations 3-8
3.2.3 Initialization 3-11
3.2.4 Data operations 3-14
3.2.4.1 Hazard detection 3-15
3.2.4.2 Quality of Service 3-16
3.2.4.3 Arbitration 3-16
3.2.5 Low-power operation 3-17
3.2.5.1 System low-power control 3-17
3.2.5.2 Dynamic low-power mode control 3-21
3.2.5.3 Deep power-down 3-24
3.2.5.4 AXI ID and Corresponding Hardware 3-25
3.3 Register Descriptions 3-26
3.3.1 SDR/DDR SDRAM Controller Registers 3-29
3.3.2 DDR2 SDRAM Controller Registers 3-39
3.3.3 SDRAM PHY Registers 3-50
3.3.4 Miscellaneous Configuration Registers 3-60
3.3.5 Memory Bus Configuration Registers 3-67
4 MISC. 4-69
4.1 Core Bus Configuration Registers 4-70
4.2 Virtual MMU Table 4-71

Figures

Figure 1.1 The TCC8900 Core & Memory Bus Architecture 1-1
Figure 2.1 Little-Endian Addresses of Bytes-Words 2-3
Figure 3.1 DRAM Controller Block Diagram 3-7
Figure 3.2 aclk Domain State Diagram 3-9
Figure 3.3 System State Transitions 3-19
Figure 3.4 Auto-power-down 3-22
Figure 3.5 Force Precharge with Zero Force Precharge Time 3-22
Figure 3.6 Force Precharge After pd Time 3-22
Figure 3.7 Auto Self-refresh Entry 3-23

Tables

Table 1.1 Remap & Address Map of TCC8900 1-2

Table 1.2 Base Address of Bus Components 1-2
Table 1.3 Detailed Base Address of Core Bus Components 1-2
Table 1.4 Detailed Base Address of Memory Bus Components 1-2
Table 2.1 Exception Vector and Priorities 2-4
Table 2.2 Exception Entry/Exit 2-5
Table 3.1 Address Comparison Steps Example 3-8
Table 3.2 Example One Time Initialization Sequence 3-11
Table 3.3 Example Initialization Sequence for Manual ZQ Calibration 3-11
Table 3.4 Example Initialization Sequence for Periodic Calibration 3-12
Table 3.5 Example Mobile DDR setup of SDR/DDR SDRAM Controller 3-12
Table 3.6 Example DDR2 setup of DDR2 SDRAM Controller 3-13
Table 3.7 Valid System States for FSMs 3-18
Table 3.8 Recommended power states 3-18
Table 3.9 Dynamic Low-power Mode Operation 3-21
Table 3.10 AXI ID and Corresponding Hardware 3-25
Table 3.11 SDR/DDR SDRAM Controller Register Map (Base Address = 0xF0301000) 3-26
Table 3.12 DDR2 SDRAM Controller Register Map (Base Address = 0xF0302000) 3-26
Table 3.13 SDRAM PHY Register Map (Base Address = 0xF0304000) 3-27
Table 3.14 Miscellaneous Configuration Register Map (Base Address = 0xF0303000) 3-27
Table 3.15 Memory Bus Configuration Register Map (Base Address = 0xF0305000) 3-28
Table 3.16 Memory Banks Chip Configuration 3-29
Table 3.17 DLL Phase Detection Register Field Description 3-52

1 Core & Memory Bus Architecture

1.1 Architecture Overview

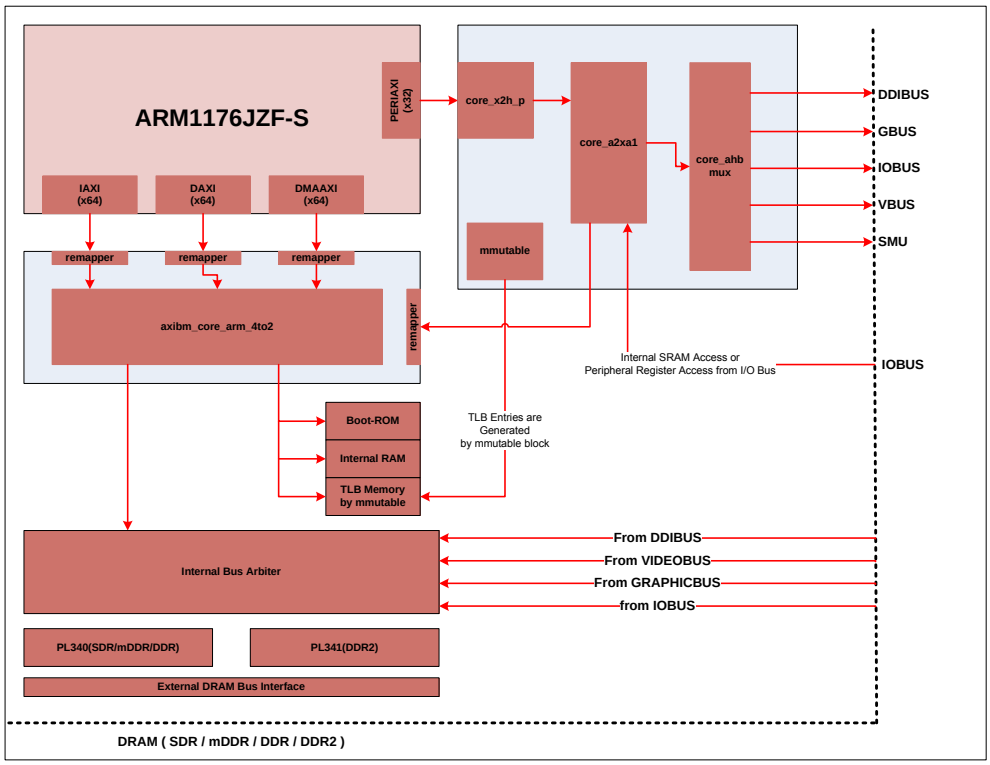


Figure 1.1 The TCC8900 Core & Memory Bus Architecture

The ARM1176JZF-S has 4 AXI bus interfaces. Only the PERIAXI has the 32bits interface and others have 64bits interface. The PERIAXI port is used to access the hardware registers.

The traffic via PERIAXI would be transferred to each buses to configure hardware registers – DDIBUS, GBUS, IOBUS, VBUS, SMU. And the traffics on the IAXI, DAXI, and DMAAXI can be transferred to the BootROM, internal RAM, TLB generated by virtual MMU and external DRAM interface.

The DRAM interface circuit has the internal bus arbiter which arbitrates the traffic to the DRAM from CPU, DDIBUS, VIDEOBUS, GRAPHICBUS, and IOBUS. Finally, PL340 and PL341 can't work simultaneously.

The required clocks are two – one is ARM1176JZF-S core clock and other is memory bus clock which are generated by CKC in SMU block.

1.2 Address Map

The TCC8900 has fixed address maps for on-chip resources and off-chip resources. The following table represents overall address space of system.

Table 1.1 Remap & Address Map of TCC8900

Address Space	Region	Device Name
0x00000000 – 0x0FFFFFFF	R.0	This region is remapped to as follows. 1) If Remap is 00b, On-chip boot-ROM for region R.E 1) If Remap is 01b, On-chip memory for region R.1 2) If Remap is 10b, Off-chip SDRAM for region R.4 3) If Remap is 11b, this region is not remapped to any region.
0x00000000 – 0x00003FFF		Special Region for Instruction TCM
0x10000000 – 0x10003FFF	R.1	Assigned to on-chip 16kB memory Region
0x20000000 – 0x20003FFF	R.2	Assigned to the TLB table generated by the virtual MMU hardware.
0x40000000 – 0x7FFFFFFF	R.3	Assigned to Off-chip SDRAM chip
0xA0000000 – 0xA0003FFF	R.4	Assigned to DTCM memory region
0xE0000000 – 0xE0003FFF	R.5	Assigned to internal boot ROM
0xF0000000 – 0xFFFFFFFF	R.6	Assigned to on-chip peripherals

The address space (0x00000000 ~ 0x0FFFFFFF) is initially allocated to internal SRAM memory for booting procedure, and a special flag exists in system controller unit for remapping this space to other type of memories. Refer to the description of system controller for detailed operation.

The TCC8900 has various peripherals for specific interface controllers or on-chip hardware components. These peripherals can be configured appropriately by its own registers that can be accessed through specially allocated address. These address maps are represented in the following table. In case of memory controller, its space is separated for preventing illegal accessing.

Refer to corresponding sections for detail information of each peripheral.

Table 1.2 Base Address of Bus Components

Base Address	Peripherals
0xF0000000	Graphic Bus Components
0xF0100000	Memory Bus Components See Table 1.3 for more details
0xF0200000	DDI Bus Components
0xF0300000	Memory Bus Components See Table 1.4 for more details
0xF0400000	System Management Unit Registers
0xF0500000	IO Bus Peripherals
0xF0600000	MMU Table
0xF0700000	Video Bus Components

Table 1.3 Detailed Base Address of Core Bus Components

Base Address	Peripherals
0xF0100000	Core Bus AXI arbiter & QoS Control Registers
0xF0101000	Core Bus Configuration Registers
0xF0102000	GPIO Registers

Table 1.4 Detailed Base Address of Memory Bus Components

Base Address	Peripherals
0xF0300000	Reserved
0xF0301000	SDR/DDR SDRAM Controller Registers
0xF0302000	DDR2 SDRAM Controller Registers
0xF0303000	Miscellaneous Configuration Registers
0xF0304000	SDRAM PHY Control Registers
0xF0305000	Memory Bus Configuration Registers

2 ARM1176JZ(F)-S Main Processor

2.1 Overview

The TCC8900 has adopted the ARM1176JZ(F)-S main processor core for controlling system and processing various kinds of digital signals. It has a Harvard architecture with separate 16Kbyte data and 16Kbytes instruction caches, each with 8-word of line length.

The ARM1176JZ(F)-S CPU core has the 3 interfaces and core features. The BUS I/F makes it possible to read from or write to the various resources. The CPU is composed of ARM9E-S core and instruction/data cache with 16KB. The cache fill or replace transfer is fulfilled through the bus interface unit. The line size of each cache is 8 words, so the burst size of the cache transfer including prefetching the instruction or data is 8 words burst.

The ARM1176JZ(F)-S main processor has the special interface different from previous ARM core. The TCM interface is very useful interface to access the external memory fast. The operating frequency of TCM interface is same as CPU core frequency, which means that if you access the TCM memory, the access speed is same as cache operation with 100% hit ratio. In this system, two memory resources are connected to TCM interface. The ITCM memory with 4KB is connected to ITCM interface of the core. The DTCM memory with 8KB is connected to DTCM interface of the core. The ITCM memory is organized by single-port SRAM and the DTCM memory by dual-port SRAM. The ITCM is very useful for exception handler or any other codes called very often such as the basis of OS kernel functions. Also, if you lay up any type of codes which require very high-performance on the ITCM, you can get the higher performance of the corresponding codes.

The DTCM memory is also very useful resource. When you access the DTCM memory, the access speed is same as data cache with 100% hit ratio. And the DTCM has the dual-port memory structure. The CPU can access the resource without any interference of another master hardware. This feature makes it possible for you to communicate with any master hardware with highest access speed.

In this chapter, we describe the key functions to make the good use of the ARM1176JZ(F)-S core and TCM such as coprocessor instruction, methods for controlling the cache and TCM. If you want to have the detailed information for the ARM1176JZ(F)-S core, refer to the ARM TRM (Technical Reference Manual) r0p5 version.

2.2 Functional Description

2.2.1 Memory Formats

The ARM1176JZ(F)-S views memory as a linear collection of bytes numbered upwards from 0. Bytes 0 to 3 hold the first word. In the same way, bytes 4 to 7 hold the second word and so on ARM1176JZ(F)-S itself can treat words in memory as being stored either in big-endian or little-endian, but in the ARM1176JZ(F)-S, there is only little-endian supported by the memory controller. The following figure illustrates the structure of little-endian type of memories.

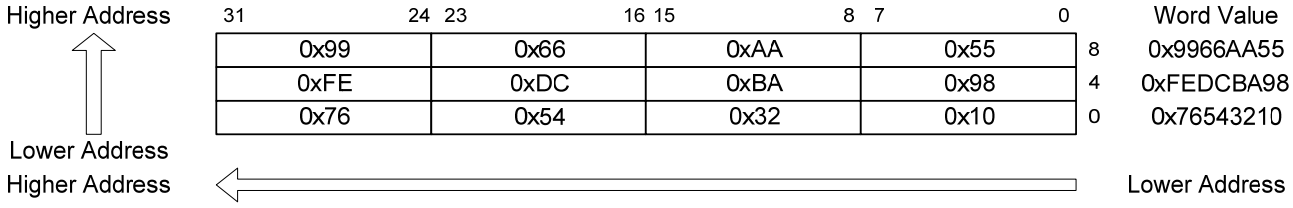


Figure 2.1 Little-Endian Addresses of Bytes-Words

2.2.2 Operating Modes

ARM1176JZ(F)-S supports seven modes of operation:

- USER (usr) The normal ARM program execution mode
- FIQ (fiq) Designed to support a data transfer or channel process
- IRQ (irq) Used for general purpose interrupt handling
- Supervisor (svc)P Protected mode for the operating system
- Abort (abt) Entered after a data or instruction prefetch abort
- System (sys) A privileged user mode for the operating system
- Undefined (und) Entered when an undefined instruction is executed

Switching between these modes may be made under software control, or may be brought about by interrupts or exception processing. Most application programs will execute in User mode. The non-user mode known as privileged modes are entered in order to service interrupts or exceptions, or to access protected resources.

2.2.3 Instruction Set Features

2.2.3.1 ARM Instruction Set

The ARM instruction set contains the following six main instruction types.

- Branch instructions
- Data processing instructions
- Status register transfer instructions
- Load/store instructions
- Cache/Protection instructions
- Exception generation instructions

Most data processing instructions update the four condition code flags (N, Z, C, V) in the current program status register (CPSR).

2.2.4 Addressing Modes

2.2.4.1 Load/Store Instructions

Load/store instructions provide three basic types of addressing modes. The addressing mode is formed from two parts; the base register and the offset.

Offset addressing mode The memory address consists of the offset added to or subtracted from the base register contents
Preindex addressing mode The memory address consists of the offset added to or subtracted from the base register contents. This new address is then overwritten to the base register.
Postindex addressing mode The memory address is the base register contents. The hardware then updates the base register by adding or subtracting the offset.

The offset takes one of three formats which are “immediate”, “register”, and “scaled register”. Immediate offset is unsigned number that can be added to or subtracted from the base register. The register offset is a general-purpose register that can be to or subtracted from the base register. The scaled register offset is a general-purpose register shifted by an immediate value.

2.2.4.2 Multiple Load/Store Instructions

Multiple load/store instructions load/store two or more general-purpose registers from/to memory. These instructions offer four addressing modes.

Increment After (IA) The base register contents are incremented after the transfer.
Increment Before (IB) The base register contents are incremented before the transfer.
Decrement After (DA) The base register contents are decremented after the transfer.
Decrement Before (DB) The base register contents are decremented before the transfer.

2.2.5 Exceptions

Exceptions arise whenever the normal flow of a program has to be halted temporarily, for example to service an interrupt from a peripheral. Before an exception can be handled, the current processor state must be preserved so that the original program can resume when the handler routine has finished.

It is possible for several exceptions to arise at the same time, and if this happens, they are dealt with in a fixed priority. This priority is illustrated in Table 2.1.

Table 2.1 Exception Vector and Priorities

Address	Exception	Mode in Entry	Priority
0x00000000	Reset	Supervisor	1
0x00000004	Undefined Instruction	Undefined	6
0x00000008	Software Interrupt	Supervisor	6
0x0000000C	Abort (Prefetch)	Abort	5
0x00000010	Abort (Data)	Abort	2
0x00000014	Reserved	Reserved	-
0x00000018	IRQ	IRQ	4
0x0000001C	FIQ	FIQ	3

• The higher priority has the lower number.

2.2.5.1 Actions on Entering an Exception

When handling an ARM exception the ARM1176JZ(F)-S core

(1) Preserves the address of the next instruction in the link register of appropriate mode. If the exception has been entered from ARM state, then the address of the next instruction is copied into the link register (current PC + 4, or +8 depending on the exception types). If the exception has been entered from THUMB state, then the value written into the link register is the current PC offset by a value such that the program resumes from the correct place on return from the exception. The exception handler does not need to determine the state when entering an exception. For example, in case of SWI exception, MOVS PC, R14_svc will always return to the next instruction regardless of whether the SWI was executed in ARM or THUMB state
(2) Copies the CPSR into the SPSR of appropriate mode
(3) Forces the CPSR mode bits to a value that represents the mode of exception
(4) Forces the PC to fetch the next instruction from the relevant exception vector

It may also set the interrupt disable flags to prevent otherwise unmanageable nesting of exceptions. If the processor is in THUMB state when an exception occurs, it will automatically switch into ARM state when the PC is loaded with the exception vector address.

2.2.5.2 Actions on Leaving an Exception

When an exception has completed, the exception handler must move the link register, minus an offset to the PC. The offset will vary depending on the type of exception.

If the S bits set rd=r15 (MOVS r15, r?), the ARM1176JZ(F)-S copies the SPSR back to the CPSR.

An explicit switch back to THUMB state is never needed, since restoring to the CPSR from the SPSR automatically sets the T bit to the value it held immediately prior to the exception.

2.2.5.3 Exception Entry/Exit Summary

The following table summarizes the PC value preserved in the relevant R14 on exception entry, and the recommended instruction for exiting the exception handler.

Table 2.2 Exception Entry/Exit

Entry	Return Instruction	Previous State		Notes
		ARM	THUMB	
SWI	MOVS PC, R14_svc	PC + 4	PC + 2	Where the PC is the address of the SWI or undefined instruction.
UNDEF	MOVS PC, R14_und	PC + 4	PC + 2	
FIQ	SUBS PC, R14_fiq, #4	PC + 4	PC + 4	Where the PC is the address of the instruction that was not executed because the FIQ or IRQ took priority.
IRQ	SUBS PC, R14_irq, #4	PC + 4	PC + 4	
PABT	SUBS PC, R14_abt, #4	PC + 4	PC + 4	Where the PC is the address of instruction that had the Prefetch Abort.
DABT	SUBS PC, R14_abt, #8	PC + 8	PC + 8	Where the PC is the address of the Load or Store instruction that generated the Data Abort.
RESET	-			The value saved in R14_svc upon reset is unpredictable.

2.2.5.4 FIQ

The FIQ (Fast Interrupt Request) exception is designed to support a data transfer or channel process, and in ARM state, it has sufficient private registers to remove the need for saving registers (thus minimizing the overhead of context switching). FIQ is generated by nFIQ signal that is controlled by internal or external interrupts defined as FIQ type. Irrespective of whether the exception was entered from ARM state or THUMB state, a FIQ handler should leave the interrupt by executing

```
SUBS    PC, R14, #4
```

FIQ may be disabled by setting the CPSR's F flag (this is not possible in USER mode). If the F flag is clear, ARM1176JZ(F)-S checks for a LOW level on the nFIQ signal at the end of every instruction.

2.2.5.5 IRQ

The IRQ (Interrupt Request) exception is a normal interrupt caused by a LOW level on the nIRQ signal that is controlled by internal or external interrupt defined as IRQ type. IRQ has lower priority than FIQ and is masked out when a FIQ sequence is entered. It may be disabled at any time by setting the I bit in the CPSR, though this can only be done in a privileged (none-User) mode.

Irrespective of whether the exception was entered from ARM state or THUMB state, an IRQ handler should return from the interrupt by executing

```
SUBS    PC, R14, #4
```

2.2.5.6 Abort

An abort indicates that the current memory access cannot be completed. ARM1176JZ(F)-S checks for the abort condition during memory access cycles.

There are two types of abort:

- Prefetch abort: occurs during an instruction prefetch
- Data abort: occurs during a data access

If a prefetch abort occurs, the prefetched instruction is marked as invalid, but the exception will not be taken until the instruction reaches the head of the pipeline. If the instruction is not executed, the abort does not take place.

If a data abort occurs, the action taken depends on the instruction type:

The swap instruction (SWP) is aborted as if it had not been executed.

Block data transfer instructions (LDM, STM) complete. If write-back is set, the base is updated. If the instruction would have overwritten the base with data, the overwriting is prevented. All register overwriting is prevented after an abort is indicated, which means in particular that R15 is preserved in an aborted LDM instruction.

After fixing the reason for the abort, the handler should execute the following irrespective of the state (ARM or THUMB):

```
SUBS    PC, R14, #4             ; for a prefetch abort
SUBS    PC, R14, #8             ; for a data abort
```

This restores both the PC and the CPSR, and retries the aborted instruction.

2.2.5.7 Software Interrupt

The software interrupt instruction (SWI) is used for entering Supervisor mode, usually to request a particular supervisor function. A SWI handler should return by executing the following irrespective of the state (ARM or THUMB):

```
MOV     PC, R14
```

This restores the PC and CPSR, and returns to the instruction following the SWI instruction.

2.2.5.8 Undefined Instruction

When ARM1176JZ(F)-S comes across an instruction that it cannot handle, it takes the undefined instruction trap. This mechanism may be used to extend either the THUMB or ARM instruction set by software emulation.

After emulating the failed instruction, the trap handler should execute the following instruction irrespective of the state (ARM or THUMB):

```
MOVS    PC, R14
```

This restores the CPSR and returns to the instruction following the undefined instruction.

2.3 Coprocessor CP15

Refer to ARM's technical reference manual.

3 DRAM Controller

3.1 Overview

The TCC8900 has a SDRAM controller for various kinds of SDRAM for digital media en-decoding system. It can manipulate SDR, DDR, DDR2 SDRAM memories. The data bus width can be configured for each chip select separately
The memory controller provides the power saving function for SDRAM (self refresh).

The following figure represents the block diagram of SDRAM control unit.

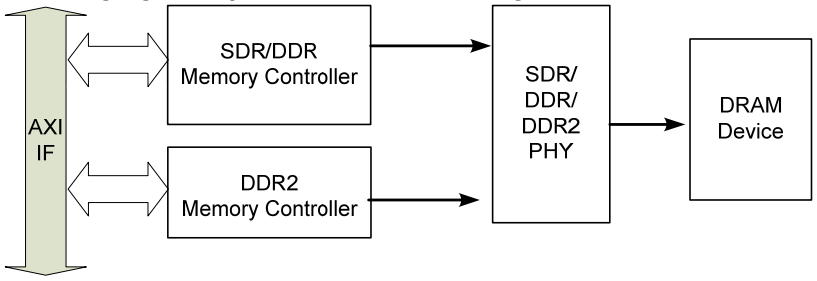


Figure 3.1 DRAM Controller Block Diagram

3.1.1.1 SDR/DDR SDRAM Controller

The SDR/DDR SDRAM controller is a high-performance, area-optimized SDRAM or Mobile SDR memory controller compatible with the AMBA AXI protocol.

The memory controller block offers the following features:

- scalable pipeline
- interface between AMBA AXI bus fabric and DDR, LPDDR (Mobile DDR), SDR, Mobile SDR and eDRAM memories
- Quality of Service (QoS) and request arbitration features for low latency transfers and optimal use of memory bandwidth
- write data interleaving supported
- multiple outstanding addresses supported
- support for ARMv6 outstanding exclusive accesses
- support synchronous and asynchronous operation between AXI bus fabric and external memory bus
- programmable support for memory power saving modes including Deep Power Down (DPD), active power-down, precharge power-down and self-refresh
- programmable through AMBA APB interface
- optimized utilization of external memory bus
- configurable for single port for all chip selects on individual cke port per external chip select.

3.1.1.2 DDR2 SDRAM Controller

The DDR2 SDRAM controller is a high-performance, area-optimized DDR2 SDRAM memory controller compatible with the AMBA AXI protocol.

The memory controller block offers the following features:

- active and precharge power-down supported in the DDR2 SDRAM
- Quality of Service (QoS) features for low latency transfers
- optimized utilization of external memory bus
- programmable selection of external memory width, see Supported memory widths
- multiple outstanding transactions
- write data interleaving supported
- hardware resource that can be rendered to optimize area versus performance
- support for a configurable number of ARMv6 outstanding exclusive accesses
- support synchronous and asynchronous operation between AXI bus fabric and external memory bus

3.2 Functional Overview

3.2.1 Exclusive access

In addition to reads and writes, exclusive reads and writes are supported in accordance with the AMBA AXI Protocol Specification.

Successful exclusive accesses have an EXOKAY response. All other accesses, including exclusive fail accesses, receive an OKAY response.

Exclusive access monitors implement the exclusive access functionality. Each monitor can track a single exclusive access. The number of monitors is a configurable option.

If an exclusive write fails, the data mask for the write is forced LOW, so that the data is not written.

When monitoring an exclusive access, the address of any write from another master is compared with the monitored address to check that the location is not being updated.

Consider the byte addresses accessed by a transaction. All the least significant bits, up to and including, the most significant bit that vary between those addresses are set to logic zero in the mask. All the stable address bits above this point are set to logic one.

Example

Exclusive Read Address = 0x100, size = WORD, length = 1, ID = 0.
Write Address = 0x104, size = WORD, length = 2, ID = 1.
Exclusive Write Address = 0x100, size = WORD, length = 1, ID = 0.

The write transaction accesses the address range 0x104-0x10B. Therefore, address bit 3 is the most significant bit that varies between byte addresses. The bit mask is therefore formed so that address bits 3 down to 0 are not compared. This has the effect that the masked write, as far as the monitoring logic has calculated, has accessed the monitored address. Therefore the exclusive write is marked as having failed.

Table 3.1 Address Comparison Steps Example

Step		Binary	Hex
1	Monitored address	b000100000000	0x100
2	Write address	b000100000100	0x104
3	Write accesses	b000100000100	0x104
		b000100000101	0x105
		b000100000110	0x105
		b000100000111	0x106
		b000100001000	0x107
		b000100001001	0x108
		b000100001010	0x10A
		b000100001011	0x10B
4	Generate a comparison mask	b111111110000	0xFF0
5	Monitored address ANDed with mask	b000100000000	0x100
6	Write Address ANDed with mask	b000100000000	0x100
7	Compare steps 5 and 6		
8	Mark exclusive write as failed		

This example shows how the logic can introduce false-negatives in exclusive access monitoring, because in reality the write has not accessed the monitored address. The implementation has been chosen to reduce design complexity, but always provides safe behavior.

When calculating the address region accessed by the write, the burst type is always taken to be INCR. Therefore, a wrapped transaction in the Example above that wraps down to 0x0 rather than cross the boundary, is treated in the same way. This is the same for a fixed burst that does not cross the boundary or wrap down to 0x0.

3.2.2 Controller management operations

The memory controller current state of operation is tracked by using an FSM.

Following figure shows the states. You can read the current status of the memory controller by reading the memc_status Register, see Memory Controller Status Register.

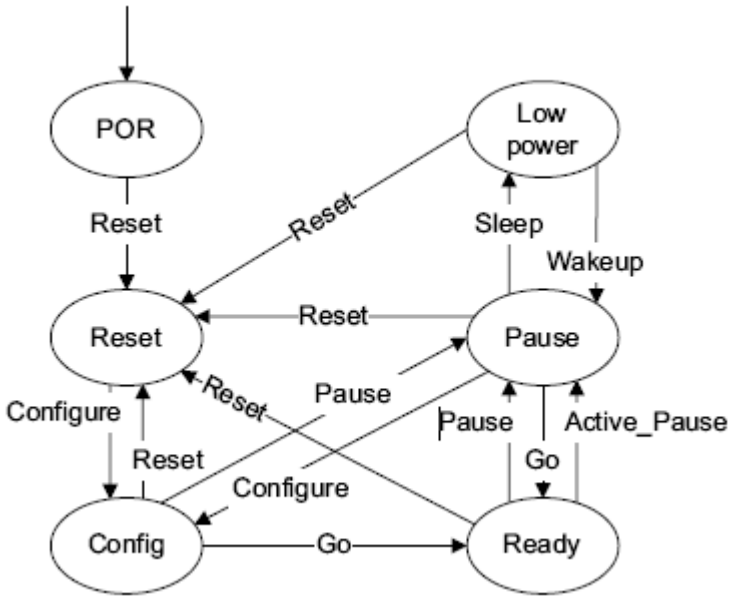


Figure 3.2 aclk Domain State Diagram

In Figure above non-state moving transitions are omitted for clarity.

Note :

- The FSM is navigated either by accepting APB direct commands or through the AXI Low-power Interface.
- If an APB command is received, that is illegal to carry out from the current state, then it is ignored and the FSM stays in the same state.
- If you move into the Pause state using Active_Pause, you are not permitted to enter the Config state.
- For the two cycles following POR, do not consider the controller to be in the Config state. For this reason, register access restrictions apply.
- You can only use the AXI low-power interface to move in and out of the Low-power state from the Ready state.
- If you enter the Low-power state using the APB interface, you must also exit the Low-power state using the APB interface.

The current status of the FSM controls the functionality of the memory controller.

- All the registers are available for writes or reads when the FSM is in the Config or Low-power state.
- When in the Config state or Low-power state, no auto-refresh commands are generated. When the Low-power state is entered, the SDRAM memories are put into self-refresh mode.
- When in the Ready state, not all registers are made available see Programmers Model section.
- You can achieve the Pause state through the APB interface using one of two commands in the memc_cmd Register, these are:
  - Pause command.

  When a Pause is requested, then the Pause state is only entered when the memory controller is idle.
  - Active_Pause command.

  When an Active_Pause is requested then the Pause state is entered when the memory interface is idle but there might still be outstanding transactions in the arbiter queue.

Note : No auto refresh commands are generated when in the Config state. If you are changing register values, it is necessary to enter the Low-power state, because this removes the risk of the memory maximum refresh time being exceeded.

The memory controller management function can issue commands to the memory interface from one of the following sources:

[Direct commands]

These are received over the APB interface as a result of a write to the direct_cmd Register. See Direct Command Register. They initialize the SDRAM.

The legal commands that the memory manager uses are:

- NOP, for SDRAM only Warning
- If you have the NVM plug-in licensed, do not use the NOP command with NVM chip selects.
- PRECHARGEALL
- AUTOREFRESH
- MODEREG
- EXTENDED MODEREG
- DEEP POWER DOWN.

[Commands from the status FSM]

You can traverse the memory manager status FSM by writing to the memc_cmd Register. You can only traverse the status FSM states when the memory controller is idle. For example, the Ready state can only be entered from the Config state when all direct commands have been completed. The exception to this is the ACTIVE_PAUSE command. You can issue this command when the memory controller is active. When you issue the command, any memory accesses that have not been arbitrated remain in the arbiter until the FSM receives a Go command.

[Refresh commands]

The refresh FSM can issue commands to the arbiter to refresh the SDRAM chips. The refresh counter is clocked by the memory clock to enable the frequency of the memory controller to be scaled without affecting the refresh rate. The refresh rate period is programmable using the Refresh Period Register. The value of this register is the count value in mclk cycles. When the refresh counter wraps around zero, an individual auto-refresh sequence is requested for each external chip in turn. You can prevent Refresh commands from being generated by using the active-chips bits in the memory_cfg Register.

Note: Refreshes are masked from the most significant chip number downwards.

These management commands are arbitrated with data commands.

3.2.3 Initialization

Before you can use the memory controller operationally to access external memory, you must:

- set up the memory controller & PHY configuration registers
- initialize the external memory devices.

You might not have to configure all the memory controller registers because some might power-up to the correct value.
Note: You might create a deadlock situation if the memory controller AXI port is accessed by a master before that master has configured the memory controller using the APB port. A master that cannot access the APB port but accesses the AXI port before the memory controller has been configured is held off until another master configures the memory controller.

Table 3.2 Example One Time Initialization Sequence

	Description
	Power up clock stabilization delay
SDRAM PHY Initialization	Configure PHY mode control register 0x400 with corresponding values
	Enable PHY DLL in 0x404
	Set PHY START_POINT & INC in 0x408
	Set PHY START in 0x404
	Wait for DLL LOCK read 0x408
	Set PHY gate control register 0x40C
	Set ZQ Calibration Ctrl register 0x428 with valid values and set CAL_START
	Wait 150 CLK clock cycles and check for END or ERROR
	Set UPDATE in 0x428
	Wait for at least 4 CLK cycles and then de-assert UPDATE in 0x428.
SDRAM Controller Initialization	Configure SDRAM Controller timing parameter registers
	Configure SDRAM Controller non-timing parameter registers [1]
DDR/DDR2 Initialization	Issue NOP
	Wait 400 ns
	Precharge All
	Issue 2 Auto Refresh commands
	Issue MRS to Reset DLL
	Issue EMRS 1(optional)
	Issue EMRS 2(optional)
	Issue EMRS 3(optional)
	Configure PL341 with memory ready command
Initialization complete	

Table 3.3 Example Initialization Sequence for Manual ZQ Calibration

	Description
	Power up clock stabilization delay
SDRAM PHY Initialization	Configure PHY mode control register 0x400 with corresponding values
	Enable PHY DLL in 0x404
	Set PHY START_POINT & INC in 0x408
	Set PHY START in 0x404
	Wait for DLL LOCK read 0x408
	Set PHY gate control register 0x40C
	Set FORCE to 1
	Set IMPP and IMPN to appropriate values
	Set UPDATE in 0x428
	Wait for atleast 4 CLK cycles and then de-assert UPDATE in 0x428.
SDRAM	Configure SDRAM Controller timing parameter registers*

[1] The following are the non timing parameters that are usually configured.
- Program Number of active chip selects
- Program memory burst length (always 4)
- Program Number of row bits
- Program Number of col bits
- Program memory width
- Program Number of leaves in each bank
- Program Initial value of CKE (mostly this will be hard coded)
- Program Initial value of DQM (mostly this will be hard coded)
- Clock relationship between AXI and memory clock

Controller Initialization	Configure SDRAM Controller non-timing parameter registers
DDR/DDR2 Initialization	Issue NOP
	Wait 400 ns
	Precharge All
	Issue 2 Auto Refresh commands
	Issue MRS to Reset DLL
	Issue EMRS 1(optional)
	Issue EMRS 2(optional)
	Issue EMRS 3(optional)
	Configure PL341 with memory ready command
Initialization complete	

Table 3.4 Example Initialization Sequence for Periodic Calibration

	Description
	Power up clock stabilization delay
SDRAM PHY Initialization	Configure PHY mode control register 0x400 with corresponding values
	Enable PHY DLL in 0x404
	Set PHY START_POINT & INC in 0x408
	Set PHY START in 0x404
	Wait for DLL LOCK read 0x408
	Set PHY gate control register 0x40C
	Set ZQ Calibration Ctrl register 0x428 with valid values and set CAL_START
	Wait 150 CLK clock cycles and check for END or ERROR
	If END is high low CAL_START
	Set UPDATE in 0x428
	Wait for at least 4 CLK cycles and then de-assert UPDATE in 0x428.
SDRAM Controller Initialization	Configure SDRAM Controller timing parameter registers*
	Configure SDRAM Controller non-timing parameter registers
DDR/DDR2 Initialization	Issue NOP
	Wait 400 ns
	Precharge All
	Issue 2 Auto Refresh commands
	Issue MRS to Reset DLL
	Issue EMRS 1(optional)
	Issue EMRS 2(optional)
	Issue EMRS 3(optional)
	Configure PL341 with memory ready command
Initialization complete	

Table 3.5 Example Mobile DDR setup of SDR/DDR SDRAM Controller

Register offset	Write data	Description
0x014	0x00000006	Set cas_latency to 3
0x018	0x00000001	Set t_dqss to 1
0x01C	0x00000002	Set t_mrd to 2
0x020	0x00000007	Set t_ras to 7
0x024	0x0000000B	Set t_rc to 11
0x028	0x00000015	Set t_rcd to 5 and schedule_rcd to 2
0x02C	0x000001F2	Set t_rfc to 18 and schedule_rfc to 15
0x030	0x00000015	Set t_rp to 5 and schedule_rp to 2
0x034	0x00000002	Set t_rrd to 2
0x038	0x00000003	Set t_wr to 3
0x03C	0x00000002	Set t_wtr to 2
0x040	0x00000001	Set t_xp to 1
0x044	0x0000000A	Set t_xsr to 10
0x048	0x00000014	Set t_esr to 20

0x00C	0x000D0020	Set memory configuration[2]
0x010	0x00000A60	Set auto refresh period to be every 2656 mclk periods
0x200	0x000000FF	Set chip select for chip 0 to be 0x00XXXXXX, RBC configuration
0x204	0x000022FF	Set chip select for chip 1 to be 0x22XXXXXX, RBC configuration
0x208	0x000055FF	Set chip select for chip 2 to be 0x55XXXXXX, RBC configuration
0x20C	0x00007FFF	Set chip select for chip 3 to be 0x7FXXXXXX, RBC configuration
0x008	0x000C0000	Carry out chip0 NOP command
0x008	0x00000000	Carry out chip0 Prechargeall command
0x008	0x00040000	Carry out chip0 Autorefresh command
0x008	0x00040000	Carry out chip0 Autorefresh command
0x008	0x00080032	Carry out chip0 Modereg command 0x32 mapped to low add bits
0x008	0x001C0000	Carry out chip1 Nop command
0x008	0x00100000	Carry out chip1 Prechargeall command
0x008	0x00140000	Carry out chip1 Autorefresh command
0x008	0x00140000	Carry out chip1 Autorefresh command
0x008	0x00180032	Carry out chip1 Modereg command 0x32 mapped to low add bits
0x008	0x002C0000	Carry out chip2 Nop command
0x008	0x00200000	Carry out chip2 Prechargeall command
0x008	0x00240000	Carry out chip2 Autorefresh command
0x008	0x00240000	Carry out chip2 Autorefresh command
0x008	0x00280032	Carry out chip2 Modereg command 0x32 mapped to low add bits
0x008	0x003C0000	Carry out chip3 Nop command
0x008	0x00300000	Carry out chip3 Prechargeall command
0x008	0x00340000	Carry out chip3 Autorefresh command
0x008	0x00340000	Carry out chip3 Autorefresh command
0x008	0x00380032	Carry out chip3 Modereg command 0x32 mapped to low add bits
0x004	0x00000000	Change memory controller state to Ready
0x014	0x00000006	Set cas_latency to 3

Table 3.6 Example DDR2 setup of DDR2 SDRAM Controller

Register offset	Write data	Description
0x014	0x00000006	Set cas_latency to 3
0x01C	0x00000002	Set t_mrd to 2
0x020	0x00000008	Set t_ras to 8
0x024	0x0000000B	Set t_rc to 11
0x028	0x00000103	Set t_rcd to 3 and schedule_rcd to 1
0x02C	0x00001315	Set t_rfc to 21 and schedule_rfc to 19
0x030	0x00001315	Set t_rfc to 21 and schedule_rfc to 19
0x034	0x00000002	Set t_rrd to 2
0x038	0x00000003	Set t_wr to 3
0x03C	0x00000002	Set t_wtr to 2
0x040	0x00000001	Set t_xp to 2
0x044	0x00000003	Set t_xsr to 3
0x048	0x000000C8	Set t_esr to 200

[2] The memory is configured as follows:
- 8 column bits, 11 row bits
- precharge all bit is shared with A10
- power-down period of 0
- auto power-down is disabled
- dynamic clock stopping is disabled
- memory burst size of 4
- ARID[5:2] bits to be used for QoS
- force precharge is disabled
- auto self-refresh is disabled.

0x00C	0x0001A411	Set memory configuration[1]
0x010	0x00000618	Set auto refresh period to be every 1560 mclk periods
0x04C	0x00000411	Set memory configuration[2]
0x200	0x000000FF	Set chip select for chip 0 to be 0x00XXXXXX, RBC configuration
0x204	0x000022FF	Set chip select for chip 1 to be 0x22XXXXXX, RBC configuration
0x208	0x000055FF	Set chip select for chip 2 to be 0x55XXXXXX, RBC configuration
0x20C	0x00007FFF	Set chip select for chip 3 to be 0x7FXXXXXX, RBC configuration
0x008	0x000C0000	Carry out chip0 NOP command
0x008	0x00000000	Carry out chip0 Prechargeall command
0x008	0x00040000	Carry out chip0 Autorefresh command
0x008	0x00040000	Carry out chip0 Autorefresh command
0x008	0x00080032	Carry out chip0 Modereg command 0x32 mapped to low add bits
0x008	0x00090000	Carry out chip0 extended Modereg command 0x0 mapped to low add bits
0x008	0x001C0000	Carry out chip1 Nop command
0x008	0x00100000	Carry out chip1 Prechargeall command
0x008	0x00140000	Carry out chip1 Autorefresh command
0x008	0x00140000	Carry out chip1 Autorefresh command
0x008	0x00180032	Carry out chip1 Modereg command 0x32 mapped to low add bits
0x008	0x00190000	Carry out chip1 extended Modereg command 0x0 mapped to low add bits
0x008	0x002C0000	Carry out chip2 Nop command
0x008	0x00200000	Carry out chip2 Prechargeall command
0x008	0x00240000	Carry out chip2 Autorefresh command
0x008	0x00240000	Carry out chip2 Autorefresh command
0x008	0x00280032	Carry out chip2 Modereg command 0x32 mapped to low add bits
0x008	0x00290000	Carry out chip2 extended Modereg command 0x0 mapped to low add bits
0x008	0x003C0000	Carry out chip3 Nop command
0x008	0x00300000	Carry out chip3 Prechargeall command
0x008	0x00340000	Carry out chip3 Autorefresh command
0x008	0x00340000	Carry out chip3 Autorefresh command
0x008	0x00380032	Carry out chip3 Modereg command 0x32 mapped to low add bits
0x008	0x00390000	Carry out chip3 extended Modereg command 0x0 mapped to low add bits
0x004	0x00000000	Change memory controller state to Ready

3.2.4 Data operations

All data operations are carried out through the AXI interface.

The number of outstanding AXI transactions that can be processed is a configurable option. Each outstanding transaction is referred to as:

[1] The memory is configured as follows:
- 9 column bits, 13 row bits
- power-down period of 0
- auto power-down is disabled
- dynamic clock stopping is disabled
- memory burst size of 4
- ARID[5:2] bits to be used for QoS – four active chips

[2] The memory is additionally configured as follows:
- aclk and mclk are synchronous
- dqm[MEMBYTES-1:0] are LOW at reset
- cke is LOW at reset
- dynamic clock stopping is disabled
- two bank bits are in use
- 16-bit data lines are in use
- DDR2 SDRAM is connected.

- arbiter queue entries, or
- entries.

An entry can be created for one of two functions:

- data entries, as the result of a AXI data transactions
- management entries, as a result of management functions.

If there is a co-incident data entry and management entry request, the management entry takes priority and delays the data entry by one clock cycle.

Entries are arbitrated with an algorithm that optimizes the efficiency of the external data bus. You can modify the algorithm to meet any programmed QoS requirement.

To achieve optimum memory bus efficiency entries might be arbitrated out of order from their arrival time. Entries that cannot be arbitrated because of hazards are removed from the algorithm until the hazard is cleared.

Each arbiter queue entry contains the transaction size remaining. This is decremented each time a memory burst of data for that entry is arbitrated until the entry is complete.

When all the data for an entry has been transferred across the memory interface the entry is deleted and the correct AXI response is sent.

An arbiter queue entry might not be arbitrated continuously. If a QoS event occurs then the highest priority entry changes. The following subsections describe:

- Hazard detection
- Quality of Service

3.2.4.1 Hazard detection

There are three types of hazard:

Read After Read (RAR)

There is a read already in the arbiter queue with the same ID as the incoming entry, that is also a read.

Write After Write (WAW)

There is a write already in the arbiter queue with the same ID as the incoming entry, that is also a write.

Read After Write (RAW)

There is a write in the arbiter queue that has received an early write response, accessing the same location as the incoming read entry.

Note: Only DDR2 SDRAM Controller supports RAW hazard detection.

The arbiter entry is flagged as having a dependency if a hazard is detected. There might be dependencies against a number of other arbiter entries. As the arbiter entries are invalidated, so the dependencies are reduced until finally, there are no outstanding dependencies, and the entry is free to start.

Note: There are no Write-After-Read (WAR) hazard checks in it. If an AXI master requires ordering between reads and writes to certain memory locations, it must wait for read data before issuing a write to a location it has read from (WAR). Similarly, the only RAW hazard checking is that performed when the write response has been issued. If an AXI master required ordering between writes and reads to certain memory locations, it must wait for the write response before issuing the read to the same location.

Write transactions are also excluded from the arbitration algorithm until either:

- a full memory burst worth of data is received
- the write data burst completes
- the write data FIFO becomes full, or
- write data with a different ID is received.

3.2.4.2 Quality of Service

QoS is defined as a method of increasing the arbitration priority of a read access that requires low-latency read data. See Arbitration for more information. The QoS for an AXI read access is determined when the arbiter receives it. No QoS exists for write accesses. There are two forms of QoS tracking:

- qos_max time-out
- qos_min time-out.

In addition, the qos_override function can dynamically force a qos_min flag for an entry.

The allocation of QoS functionality is determined by the ARID of the entry compared with a 4-bit selection mask defined by the qos_master_bits in the memory_cfg Register. The 4-bits selected can be any four contiguous bits from up to eight ARID bits, that is, [3:0], [4:1], [5:2], [6:3], or [7:4].

The resultant 4-bit QoS selection number is compared against the qos_override ports that QoS interface describes, and also enables 16 separate programmable QoS options, that the 16 id_<n>_cfg Registers. See id_<n>_cfg Registers.
If the QoS enable bit for the ARID is set in the register bank, the QoS maximum latency value is decremented every cycle until it reaches zero.

If the entry is still in the queue when the QoS maximum latency value reaches zero, then the entry becomes high priority. This is called a time-out. Also, any entry in the queue with a minimum latency QoS also produces a time-out. Minimum latency time-outs have priority over maximum latency time-outs.

When an entry times out in this way it forces a time-out onto any entries that it has dependencies against. In normal operation, these entries have already timed out because they have received the same initial QoS value, but been decrementing for longer. The highest priority arbiter entry is serviced next.

One special case exists. This is when or if the qos_override function forces a minimum latency time-out. In this instance, any accesses that the new entry has dependencies against might not have timed out and are forced to time out so that the high-priority entry can start as soon as possible.

There is also a QoS provided for the auto-refresh commands from the memory manager. The arbiter keeps track of the number of auto-refresh commands in the arbiter queue with a simple increment-decrement counter. If the number of auto-refresh commands reaches a programmable tide mark, max_out_refs, a refresh time-out is signaled to the arbiter queue. This forces all of the auto-refresh queue entries to have a time-out. This time-out is sticky, and does not disappear when the number of time-outs drops back below the threshold. Instead, it remains asserted until all of the auto-refreshes have been serviced. This provides a guaranteed refresh rate in the SDRAM.

3.2.4.3 Arbitration

The arbitration algorithm, without considering QoS issues, can operate in the following ways:

- bank preparation, that is, any memory operations to a closed row
- read or write to open rows, that is, any memory operation to an open row
- manager operations, for example, refreshes.

A particular queue item, that contains either one AXI transaction or manager operation, cannot appear in two groups. If a read transaction enters the queue, to a closed row, it is made available to the arbiter as a bank preparation operation, and not a read hit. When the row has been opened, possibly after two bank preparation operations, it is flagged as a read hit.
A read to a closed row is split into:

- a bank preparation
- an open_row read.

These are stored in the same arbiter queue slot.

Arbitration example

Assume chip 0 has recently been refreshed, all rows are closed, and nothing can be issued for tRFC. During tRFC, the following transactions enter the queue, in the following order:

1. Read, ROW1 chip0 bank 0, transaction 1
2. Read, ROW1 chip0 bank 0, transaction 2
3. Read, ROW4 chip 0 bank 1, transaction 3
4. Write, ROW1 chip 0 bank 0, transaction 4

5. Nothing is arbitrated until tRFC has expired.

The cycle numbers in this example can change depending on factors such as schedule_delays, burst lengths, and the cfifo_length. The following describes the behavior during each cycle:

- Cycle 1 No transactions are marked as either read_hit or write_hit because all rows are closed. Instead, all transactions are marked as bank preparations. The oldest transaction is selected, and transaction 1, ROW1 chip0, bank 0 is marked as open in the row cache.
- Cycle 2 Transactions 1, 2, and 4 are now open row reads or writes. However, because the activate command has only recently been issued, they are not available to the arbiter until schedule_rcd has expired. The arbiter therefore selects the next available bank preparation operation, because the refresh is the lowest priority. This is transaction 3.
- Cycle 3 Transactions 1, 2, and 4 are flagged as read_hits/write_hits because the delay has now expired, assuming that schedule_rcd was set to 1. Because a bank preparation operation was performed in the last cycle, the oldest read hit now becomes the highest priority. Transaction 1 is therefore arbitrated and removed from the queue.
- Cycle 4 Transactions 2, 3, and 4 are now available as read/write hits. The last cycle was a read operation so bank-preparation operations are the highest priority. However, because there are none to perform, transaction 2 is arbitrated. Now a new transaction arrives, read, ROW1 chip0 bank 0, transaction 5. Because ROW1 chip0 bank 0 is already open, it is flagged as a read_hit immediately.
- Cycle 5 Transactions 3, 4, and 5 are available as a read_write hits. Transaction 3 is selected because it is the oldest transaction. A new transaction now arrives, read, ROW 3 chip0 bank 1, Transaction 6. Because bank1 row 4 is still open, this transaction is to a closed row.
- Cycle 6 Transactions 4 and 5 are read/write hits. Transaction 6 is a bank preparation operation. The last transactions were reads, so the bank preparation for transaction 6 is selected, and this issues a precharge because another row in the same bank was open.
- Cycle 7 Transactions 4 is a write_hit, transaction 5 is a read_hit, and transaction 6 is unavailable while t_rp expires. The arbiter tries to maintain the data direction because this is more efficient on the memory bus. Transaction 5, a read, is therefore arbitrated.
- Cycle 8 Transactions 4 is a write hit, and transaction 6 is available as a bank preparation operation because all rows in the bank are closed. The last operation was a read, so the bank preparation operation is selected, transaction 6, and an activate command is issued.
- Cycle 9 Transactions 4 is now a write hit and transaction 6 is unavailable while t_rcd expires. The arbiter chooses the only available operation and that is transaction 4. It is then removed from the queue.
- Cycle 10 Transaction 6 is now a read_hit, and is finally arbitrated as a read_hit, and removed form the queue.

3.2.5 Low-power operation

The memory controller provides architectural support for low-power operation by supporting the SDRAM low-power modes of operation:

- automatic closing of rows
- active power-down
- precharge power-down
- automatic self-refresh entry
- self-refresh
- DPD.

3.2.5.1 System low-power control

The memory controller provides architectural support for low-power operation in the following ways:

- By using the memc_cmd and memc_status Registers at address offsets 0x4 and 0x0. The memory controller can place the SDRAM into the self-refresh state under software control.
- By using the AXI low-power interface, the memory controller can place the SDRAM into

the self-refresh state under hardware control.

Additionally, the memory controller microarchitecture provides additional power savings through extensive use of clock gating. This includes clock gating of the external memory clocks by selecting the stop_mem_clock bit in the mem_cfg Register.

You can also implement the memory controller with two power domains:

- APB and AXI, aclk
- memory, mclk.

Table 3.7 lists the valid system states of the aclk domain FSM and the mclk domain FSM. It also lists the valid power, clock, and reset states in the aclk and mclk domains.

Table 3.7 lists the valid transitions, and the text following it explains how to traverse the system states.

Table 3.7 Valid System States for FSMs

SDRAM		SDRAM Controller								System
		aclk FSM				mclk FSM				
VDD	State	VDD	Clock	Reset	State	VDD	Clock	Reset	State	States
0	Null	0	N/a	N/a	Null	0	N/a	N/a	Null	1
0	Null	>0	Running	No	POR	>0	Running	No	POR	2
0	Null	>0	Running	Yes	Reset	>0	Running	Yes	Reset	3
0	Null	>0	Running	No	Config	>0	Running	No	Powered_up	4
>0	Accessible	>0	Running	No	Config	>0	Running	No	Powered_up	5
>0	Accessible	>0	Running	No	Ready	>0	Running	No	Powered_up	6
>0	Powereddown	>0	Running	No	Ready	>0	Running	No	Powered_down	7
>0	Self_ refresh	>0	Running	No	Low_power	>0	Running	No	Self_refresh	8
>0	Self_ refresh	>0	Running	No	Low_power	>0	Stopped	No	Self_refresh	9
>0	Self_ refresh	>0	Stopped	No	Low_power	>0	Running	No	Self_refresh	10
>0	Self_ refresh	>0	Stopped	No	Low_power	>0	Stopped	No	Self_refresh	11
>0	Self_ refresh	0	N/a	N/a	Null	>0	Stopped	No	Self_refresh	12
>0	Self_ refresh	0	N/a	N/a	Null	>0	Running	No	Self_refresh	13
>0	Self_ refresh	>0	Running	No	POR	>0	Stopped	No	Self_refresh	14
>0	Self_ refresh	>0	Running	No	POR	>0	Running	No	Self_refresh	15
>0	Self_ refresh	>0	Running	Yes	Reset	>0	Stopped	No	Self_refresh	16
>0	Self_ refresh	>0	Running	Yes	Reset	>0	Running	No	Self_refresh	17
>0	Self_refresh	>0	Stopped	No	Ready	>0	Stopped	No	Self_refresh	18

The ranking of system power states, from highest power to lowest power, is as follows: 6, 7, 8, 10, 9, 11, 13, 12.
However, states 8-11 are similar and the recommendation is to use state 11 from this group if clock-stopping techniques are available. Similarly, states 12 & 13 are similar and the recommendation is to use state 12 from this pair. Table 3.8 lists a recommended set of power states.

Table 3.8 Recommended power states

System state	Power name
6	Running
7	Auto power-down
11	Shallow self-refresh or auto self-refresh
12	Deep self-refresh
18	Auto self-refresh

Note: States 1-5, 9, 14, and 16 are only used as transitional states.

Figure 3.3 highlights these states and arcs.

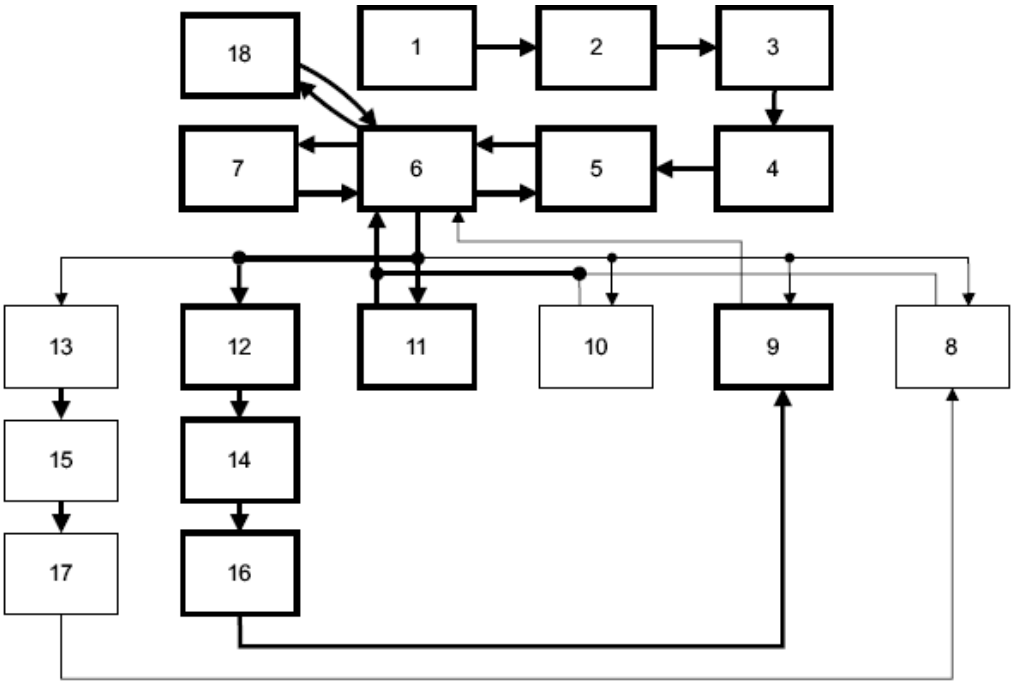


Figure 3.3 System State Transitions

State transitions are as follows:

- Arc 1 to 2
  Apply power to all memory controller power domains, and ensure that aclk and mclk are running.

- Arc 2 to 3
  Assert reset in both the aclk reset domain and the mclk reset domain.

- Arc 3 to 4
  Deassert reset in both the aclk reset domain and the mclk reset domain.

- Arc 4 to 5
  Apply power to the SDRAM power domain.

- Arc 5 to 6 You must:
  1. Write to all of the memory timing parameters, address offsets 0x14 to 0x44.
  2. Write to the memory_cfg and refresh_prd Registers, address offsets 0xC and 0x10.
  3. Initialize the memory, using the direct_cmd Register, offset 0x8, with the sequence of commands specified by the memory vendor. When you have sent these commands to the memory, you can write to the memc_cmd Register, offset 0x4 with the Go command, 0x0.
  4. Poll the memc_status Register until the value of 0x1 is returned, Ready, signifying that the memory controller is ready to accept AXI accesses to the SDRAM.

- Arc 6 to 5
  If you want to reconfigure either the memory controller or SDRAM, you must first write to the memc_cmd Register, offset 0x4, with the Pause command, 0x3, and poll the memc_status Register until the value of 0x2 is returned, Paused. Then you can write to the memc_cmd Register with the Configure command, 0x4 and poll the memc_status Register until the value of 0x0 is returned, Config.

- Arc 6 to 7
  If auto_power_down is set in the memory_cfg Register, see Memory Configuration Register, then this arc is automatically taken when the SDRAM has been idle for power_down_prd mclk cycles.

- Arc 7 to 6
  When an SDRAM access command has been received in the mclk domain, this arc is taken.

- Arc 6 to 8
  You can take this arc under either hardware or software control:
  - To take this arc under software control:
    1. Issue the Pause command, or archive the Pause command.
    2. Poll for the Paused state.
    3. Issue the Sleep command.
  - To take this arc under hardware control, use the AXI low-power interface to request a Low-power state.

- Arc 6 to 9
  The same as arc 6 to 8, but additionally stop the mclk domain clock.

- Arc 6 to 10
  The same as arc 6 to 8, but additionally stop the aclk domain clock.

- Arc 6 to 11
  The same as arc 6 to 8, but additionally stop both the mclk and the aclk domain clocks.

- Arc 6 to 12
  The same as arc 6 to 8, but additionally stop the mclk domain clock and remove power from the aclk power domain. This can only be done if the memory controller implementation has separate power domains for aclk and mclk.

- Arc 6 to 13
  The same as arc 6 to 8, but additionally remove power from the aclk power domain. This can only be done if the memory controller implementation has separate power domains for aclk and mclk.

- Arc 8 to 6
  You can take this arc under either hardware or software control:
  - To take this arc under software control:
    1. Issue the Wakeup command to the memc_cmd Register.
    2. Poll the memc_status Register for the Paused state.
    3. Issue the Go command and poll for the Ready state.
  - To take this arc under hardware control, use the AXI low-power interface to bring the memory controller out of a low-power state.

- Arc 9 to 6
  The same as arc 8 to 6, but you must first start the mclk domain clock.

- Arc 10 to 6
  The same as arc 8 to 6, but you must first start the aclk domain clock.

- Arc 11 to 6
  The same as arc 8 to 6, but you must first start both the aclk and mclk domain clocks.

- Arc 12 to 14
  Apply power to the aclk power domain.

- Arc 14 to 16
  Assert reset to the aclk reset domain.

- Arc 16 to 9
  De-assert reset to the aclk reset domain.

- Arc 13 to 15
  Apply power to the aclk power domain.

- Arc 15 to 17
  Assert reset to the aclk reset domain.

- Arc 17 to 8
  De-assert reset to the aclk reset domain.

- Arc 6 to 18
  If AUTO_PD is set in the Memory Configuration Register, then this arc is automatically taken when the SDRAM has been idle for PD_PRD mclk cycles. Also requires: FP_LINE << PD_PRD, FP_EN and SR_EN.

- Arc 18 to 6
  When an SDRAM access command has been received in the mclk domain, this arc is taken.

Note : When power is applied to the aclk domain, when leaving state 1, the memory controller status FSM moves to the Config state. When power is applied to the aclk domain, when leaving states 12 or 13, the memory controller states FSM moves to the Low-power state.

3.2.5.2 Dynamic low-power mode control

Dynamic low-power mode control operates when the memory controller is in the Ready state.

The functionality that lists is dependant on whether the memory controller is configured to have a single global cke or a cke per memory device. The functionality works for each cke pin, when using a:

- **global cke** : All memory devices must be idle.
- **local cke** : A single memory device can be entered into a low-power mode of operation.

The functionality listed in is configurable and programmable. A memory controller configuration requires the functionality to be included before it can be enabled through the APB interface.

Note: When auto self-refresh entry is configured, a 10-bit prescalar is also configured to increase the time of the auto_power_prd. Auto self-refresh entry cannot work without the auto_power_down functionality being enabled and the force precharge functionality being enabled.

Table 3.9 lists the dynamic low-power modes operation.

Table 3.9 Dynamic Low-power Mode Operation

Auto power-down	Force precharge	Auto self-refresh entry	Operation
0	0	0	No power saving
0	0	1	No power saving
0	1	0	Force precharge after fp_time
0	1	1	Force precharge after fp_time
1	0	0	Auto-power-down after power_down_prd
1	0	1	Auto-power-down after power_down_prd
1	1	0	Force precharge after fp_time plus Auto-power-down after power_down_prd
1	1	1	Force precharge after fp_time plus self-refresh entry after power_down_prd

Auto power-down logic negates cke for a memory chip putting the device into either active or precharge power-down mode if a device has been idle for power_down_prd time depending on whether the device had any open rows. The mclk clock decrements the power-down counters. The programmed power_down_prd time must always be greater than the programmed cas latency.

Force precharge logic automatically generates a precharge for an idle activated bank. If a bank has been activated and has executed a data access then subsequently, if no more data accesses are executed for fp_time, then a force precharge is

generated to close that idle bank. The aclk clock decrements the force precharge counters.

Auto self-refresh logic operates the same as the auto power-down logic but generates a self-refresh entry for a memory device command instead of negating cke.

Figure 3.4 shows the time after completion of a command to a memory chip until the memory controller puts that chip into power-down mode. Power-down affects all the banks of a chip, therefore there might be cases whereby some banks of a chip enters precharge power-down. However, it would normally be expected for at least one bank to enter active power-down.

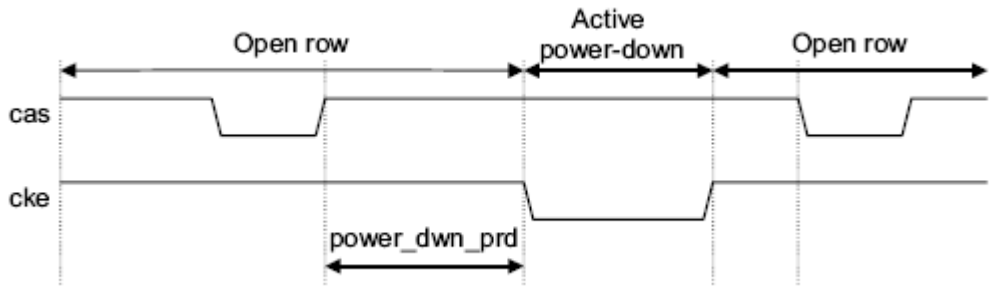


Figure 3.4 Auto-power-down

Figure 3.5 shows the time after completion of a command to a memory chip until the memory controller places that chip into power-down mode. When fp_enable is set with fp_time set to zero then the equivalent functionality of auto-precharge commands is achieved.

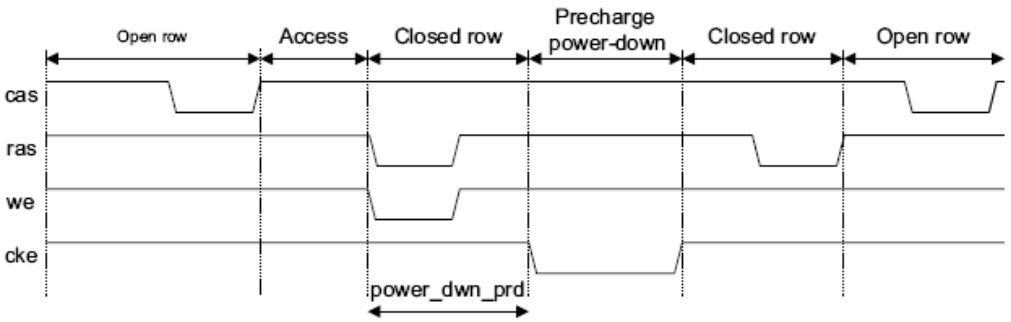


Figure 3.5 Force Precharge with Zero Force Precharge Time

Figure 3.6 shows the time after completion of a command to a memory chip until the memory controller places that chip into precharge power-down mode. To ensure precharge power-down mode for every bank, you must set fp_time to less than power_down_prd - 3 for synchronous 1:1 clocking and scaled accordingly for different clocking modes. For example if mclk is running 2 times slower than aclk then fp_time must be:

((fp_time x 2) + 3) < power_down_prd

When running asynchronously the fp_time must be scaled to ensure it must always be less than the power_down_prd - 3. This ensures that a precharge always occurs before the cke pin is negated.

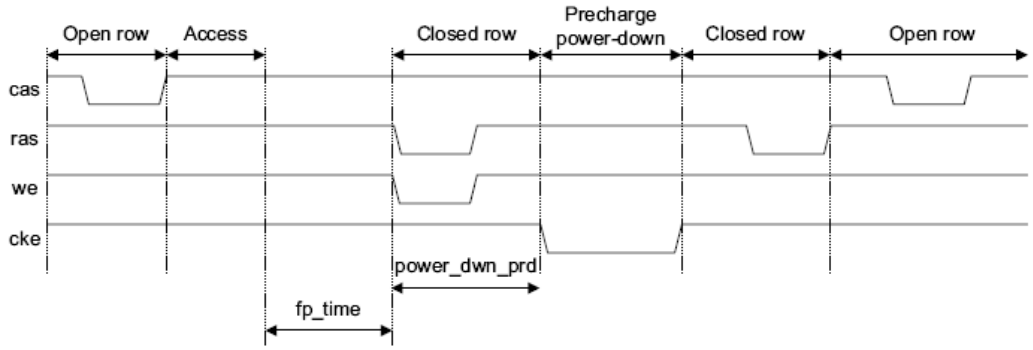


Figure 3.6 Force Precharge After pd Time

Figure 3.7 shows the time after completion of a command to a memory chip, until the memory controller places that chip into self-refresh mode. Table 3.9 shows the memory controller can only put a chip into self-refresh mode if the force precharge

logic and auto self-refresh logic is configured and enabled. This guarantees that if a self-refresh command is generated all the banks of a chip have been previously precharged. When the auto self-refresh command logic is configured a 10-bit prescalar for each power_down_prd counter is generated. If the prescale value is programmed to zero then the prescalar does not affect the power-down counter.

A prescalar is auto-generated because for some memory types there is a relatively long time for a memory chip to exit self-refresh mode and it stalls the command FIFO.

Therefore, because the penalty for exiting self-refresh mode is large, you can program a chip select to be idle for a much longer time before entering self-refresh mode when compared to the other power-down modes.

Even if auto-self-refresh is disabled, if the prescalar is programmed, then the power-down counter uses this value.

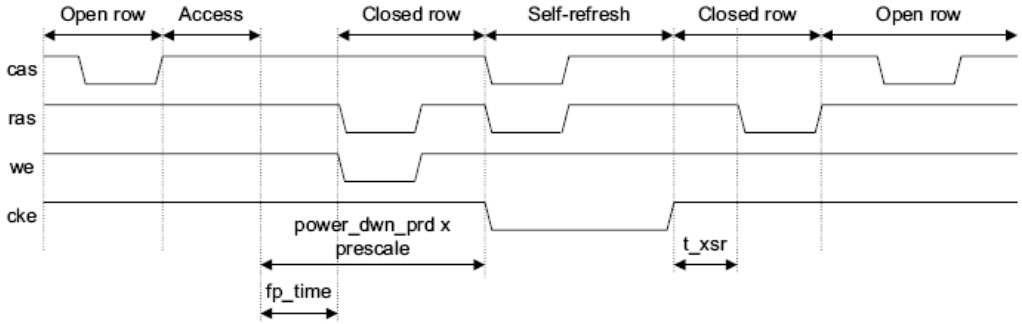


Figure 3.7 Auto Self-refresh Entry

3.2.5.3 Deep power-down

You can only achieve Deep Power Down (DPD) for individual chips if the memory controller configuration has local cke set, that is, a cke pin for each memory device.

For a memory controller configuration with a global cke it is only possible to enter all the memory devices into the DPD state simultaneously.

To ensure that no refreshes are generated for a chip select that has been put into DPD mode, the active_chips bits must be reduced in line with the DPD commands. This means that DPD mode can only be entered from the most significant chip select of a configuration downwards.

The system architect must ensure that no data transactions are sent to a chip select that has been entered into DPD mode. The flow for entering power-down mode is as follows:

1. Enter Pause state.

2. Enter Config state.

3. Write a direct command to the highest chip select number configured with a PRECHARGEALL command, so as to only enter DPD with all chips precharged.

4. Write a direct command to the highest chip select number configured with a DPD command.

5. Write a direct command to the next highest chip select number if required.

6. Write to the active_chips bits of the memory_configuration register to disable refreshes for the power-down chip selects. To remove a chip select from DPD is the reverse of the above sequence, but substituting the PRECHARGEALL command for a NOP command.

Note:

- A chip select must have the active_chips bits set before providing the NOP direct command and then carrying out the memory initialization sequence.
- If you have licensed the NVM plug-in, there is a restriction. When exiting DPD mode for SDRAM, in conjunction with the active_chips setting, the software driver must not issue NOP commands to an NVM chip select.

All the registers are available for writes or reads when the FSM is in the Config or Low-power state.

3.2.5.4 AXI ID and Corresponding Hardware

Table 3.10 AXI ID and Corresponding Hardware

AXIID[9:7]	AXIID[6:0]	Corresponding Hardware or Hardware Group
000b	110000b	I/O Bus Components
001b	0000000b	LCD Controller 0 – Channel 0 – Sub Channel 0
	0000001b	LCD Controller 0 – Channel 0 – Sub Channel 1
	0000010b	LCD Controller 0 – Channel 0 – Sub Channel 2
	0000011b	LCD Controller 0 – Channel 1 - Invalid
	0000100b	LCD Controller 0 – Channel 2 - Invalid
	0000101b	LCD Controller 1 – Channel 0 – Sub Channel 0
	0000110b	LCD Controller 1 – Channel 0 – Sub Channel 1
	0000111b	LCD Controller 1 – Channel 0 – Sub Channel 2
	0101000b	LCD Controller 1 – Channel 1
	0101001b	LCD Controller 1 – Channel 2
	0101010b	M2M Scaler 0 – Sub Channel 0
	0101011b	M2M Scaler 0 – Sub Channel 1
	0101100b	M2M Scaler 0 – Sub Channel 2
	0101101b	M2M Scaler 1 – Sub Channel 0
	0101110b	M2M Scaler 1 – Sub Channel 1
	0101111b	M2M Scaler 1 – Sub Channel 2
	1010000b	VIQE Channel 0 – Sub Channel 0
	1010001b	VIQE Channel 0 – Sub Channel 1
	1010010b	VIQE Channel 0 – Sub Channel 2
	1010011b	VIQE Channel 1 – Sub Channel 0
	1010100b	VIQE Channel 1 – Sub Channel 1
	1010101b	VIQE Channel 1 – Sub Channel 2
	1010110b	VIQE Channel 2 – Sub Channel 0
	1010111b	VIQE Channel 2 – Sub Channel 1
	1111000b	VIQE Channel 2 – Sub Channel 2
	1111001b	CIF Channel
	1111010b	DDI Cache Channel 0 – Entry Number 26
	1111011b	DDI Cache Channel 1 – Entry Number 27
	1111100b	DDI Cache Channel 2 – Entry Number 28
	1111101b	DDI Cache Channel 3 – Entry Number 29
	1111110b	DDI Cache Channel 4 – Entry Number 30
	1111111b	DDI Cache Channel 5 – Entry Number 31
010b	1100000b	ARM Instruction Bus
	1000001b	ARM Data Bus
	0100010b	ARM DMA Bus
	0000011b	Invalid
011b	0000000b	Overlay Mixer Channel 0
	0100000b	Overlay Mixer Channel 1
	1000000b	Overlay Mixer Channel 2
	11XXXXXb	3D Graphic Engine
100b	000XXXXb	Video Codec Primary Channel
	0010000b	JPEG Encoder
	0100000b	JPEG Decoder
	0111111b	Video Cache
	100XXXXb	Video Codec Secondary Channel

In the above table, the “Sub Channel” means the separated or interleaved format, such as YUV separate or YUV interleaved format.

3.3 Register Descriptions

Table 3.11 SDR/DDR SDRAM Controller Register Map (Base Address = 0xF0301000)

Name	Offset	Type	Reset	Description
M0STAT	0x000	RO	-	Status Register
M0CMD	0x004	WO	-	Command Register
M0DCMD	0x008	WO	-	Direct COmmnad Register
M0CFG	0x00C	R/W	0x00010020	Configuration Register
M0REF	0x010	R/W	0x00000A60	Refresh Period Register
M0CAS	0x014	R/W	0x00000006	CAS Latency Register
M0DQSS	0x018	R/W	0x00000001	t_dqss Register
M0MRD	0x01C	R/W	0x00000002	t_mrd Register
M0RAS	0x020	R/W	0x00000007	t_ras Register
M0RC	0x024	R/W	0x0000000B	t_rc Register
M0RCD	0x028	R/W	0x0000001D	t_rcd Register
M0RFC	0x02C	R/W	0x000000212	t_rfc Register
M0RP	0x030	R/W	0x0000001D	t_rp Register
M0RRD	0x034	R/W	0x00000002	t_rrd Register
M0WR	0x038	R/W	0x00000003	t_wr Register
M0WTR	0x03C	R/W	0x00000002	t_wtr Register
M0XP	0x040	R/W	0x00000001	t_xp Register
M0XSR	0x044	R/W	0x0000000A	t_xsr Register
M0ESR	0x048	R/W	0x00000014	t_esr Register
M0CFG2	0x04C	R/W	-	Memory_cfg2 Register
M0CFG3	0x050	R/W	0x00000007	Memory_cfg3 Register
-	0x054-0x0FC			Reserved
M0ID0	0x100	R/W	0x00000000	AXI ID0 configuration Register
M0ID1	0x104	R/W	0x00000000	AXI ID1 configuration Register
M0ID2	0x108	R/W	0x00000000	AXI ID2 configuration Register
M0ID3	0x10C	R/W	0x00000000	AXI ID3 configuration Register
M0ID4	0x110	R/W	0x00000000	AXI ID4 configuration Register
M0ID5	0x114	R/W	0x00000000	AXI ID5 configuration Register
M0ID6	0x118	R/W	0x00000000	AXI ID6 configuration Register
M0ID7	0x11C	R/W	0x00000000	AXI ID7 configuration Register
M0ID8	0x120	R/W	0x00000000	AXI ID8 configuration Register
M0ID9	0x124	R/W	0x00000000	AXI ID9 configuration Register
M0ID10	0x128	R/W	0x00000000	AXI ID10 configuration Register
M0ID11	0x12C	R/W	0x00000000	AXI ID11 configuration Register
M0ID12	0x130	R/W	0x00000000	AXI ID12 configuration Register
M0ID13	0x134	R/W	0x00000000	AXI ID13 configuration Register
M0ID14	0x138	R/W	0x00000000	AXI ID14 configuration Register
M0ID15	0x13C	R/W	0x00000000	AXI ID15 configuration Register
-	0x140-0x1FC			Reserved
M0CH0	0x200	R/W	0x0000FF00	CHIP ID0 configuration Register
M0CH1	0x204	R/W	0x0000FF00	CHIP ID1 configuration Register
M0CH2	0x208	R/W	0x0000FF00	CHIP ID2 configuration Register
M0CH3	0x20C	R/W	0x0000FF00	CHIP ID3 configuration Register

Table 3.12 DDR2 SDRAM Controller Register Map (Base Address = 0xF0302000)

Name	Offset	Type	Reset	Description
M1STAT	0x000	RO	-	Status Register
M1CMD	0x004	WO	-	Command Register
M1DCMD	0x008	WO	-	Direct Command Register
M1CFG	0x00C	R/W	0x00010020	Configuration Register
M1REF	0x010	R/W	0x00000A2C	Refresh Period Register
M1CAS	0x014	R/W	0x0000000A	CAS Latency Register
M1WRL	0x018	RO	0x00000004	Write latency Register
M1MRD	0x01C	R/W	0x00000002	t_mrd Register
M1RAS	0x020	R/W	0x0000000E	t_ras Register
M1RC	0x024	R/W	0x00000012	t_rc Register
M1RCD	0x028	R/W	0x00000305	t_rcd Register
M1RFC	0x02C	R/W	0x000002123	t_rfc Register

Name	Offset	Type	Reset	Description
M1RP	0x030	R/W	0x00000305	t_rp Register
M1RRD	0x034	R/W	0x00000004	t_rrd Register
M1WR	0x038	R/W	0x00000005	t_wr Register
M1WTR	0x03C	R/W	0x00000004	t_wtr Register
M1XP	0x040	R/W	0x00000002	t_xp Register
M1XSR	0x044	R/W	0x00000027	t_xsr Register
M1ESR	0x048	R/W	0x00000014	t_esr Register
M1CFG2	0x04C	R/W	-	Memory_cfg2 Register
M1CFG3	0x050	R/W	0x00000007	Memory_cfg3 Register
M1_FAW	0x054	R/W	0x00000011	t_faw Register
	0x058-0x0FC			Reserved
M1ID0	0x100	R/W	0x00000000	AXI ID0 configuration Register
M1ID1	0x104	R/W	0x00000000	AXI ID1 configuration Register
M1ID2	0x108	R/W	0x00000000	AXI ID2 configuration Register
M1ID3	0x10C	R/W	0x00000000	AXI ID3 configuration Register
M1ID4	0x110	R/W	0x00000000	AXI ID4 configuration Register
M1ID5	0x114	R/W	0x00000000	AXI ID5 configuration Register
M1ID6	0x118	R/W	0x00000000	AXI ID6 configuration Register
M1ID7	0x11C	R/W	0x00000000	AXI ID7 configuration Register
M1ID8	0x120	R/W	0x00000000	AXI ID8 configuration Register
M1ID9	0x124	R/W	0x00000000	AXI ID9 configuration Register
M1D10	0x128	R/W	0x00000000	AXI ID10 configuration Register
M1D11	0x12C	R/W	0x00000000	AXI ID11 configuration Register
M1ID12	0x130	R/W	0x00000000	AXI ID12 configuration Register
M1ID13	0x134	R/W	0x00000000	AXI ID13 configuration Register
M1ID14	0x138	R/W	0x00000000	AXI ID14 configuration Register
M1ID15	0x13C	R/W	0x00000000	AXI ID15 configuration Register
-	0x140-0x1FC			Reserved
M1CH0	0x200	R/W	0x0000FF00	CHIP ID0 configuration Register
M1CH1	0x204	R/W	0x0000FF00	CHIP ID1 configuration Register
M1CH2	0x208	R/W	0x0000FF00	CHIP ID2 configuration Register
M1CH3	0x20C	R/W	0x0000FF00	CHIP ID3 configuration Register

Table 3.13 SDRAM PHY Register Map (Base Address = 0xF0304000)

Name	Offset	Type	Reset	Description
REG0	0x400	R/W	0x00000000	PHY Mode Control Register
REG1	0x404	R/RW	0x00000018	DLL Control & Status Register
REG2	0x408	R/W	0x00000000	DLL Phase Detector configuration Register
REG3	0x40C	R/W	0x00000000	Gate Control Register
REG4	0x410	R/W	0x00000000	Read Data Slice 0 Control Register
REG5	0x414	R/W	0x00000000	Read Data Slice 1 Control Register
REG6	0x418	RO	0x00000000	Read Data Slice 2 Control Register
REG7	0x41C	R/W	0x00000000	Read Data Slice 3 Control Register
REG8	0x420	R/W	0x00000000	CLK Delay Register
REG9	0x424	R/W	0x00000000	DLL Force Lock Value Register
REG10	0x428	R/W	0x00000000	ZQ Calibration Control Register
REG11	0x42C	RO	0x00000000	ZQ Calibration Status Register
REG12	0x430	R/W	0x00000000	Read Delay Register

Table 3.14 Miscellaneous Configuration Register Map (Base Address = 0xF0303000)

Name	Offset	Type	Reset	Description
M0CFG0	0x00	R/W	0x80400000	SDR/DDR SDRAM Controller Configuration Register 0
M0CFG1	0x04	R/RW	0x00000018	SDR/DDR SDRAM Controller Configuration Register 1
-	0x08-0x0C			Reserved
M1CFG0	0x10	R/W	0x80000000	DDR2 SDRAM Controller Configuration Register 0
M1CFG1	0x14	R/W	0x00000000	DDR2 SDRAM Controller Configuration Register 1
-	0x18-			Reserved

Name	Offset	Type	Reset	Description
	0x1C			
COMMON	0x20	R/W	0x00010103	Common Control Register
PHYCTRL	0x24	R/W	0x00000000	SDRAM PHY Control Register
PHYTSTS	0x28	RO	0x00000000	SDRAM PHY Status Register
IOCFG	0x2C	R/W	0x00000000	SDRAM IO Control Register

Table 3.15 Memory Bus Configuration Register Map (Base Address = 0xF0305000)

Name	Offset	Type	Reset	Description
TZPROT	0x00	RW	0x00000001	Memory Area Protection Register
CKDOWN	0x04	RW	0x00000000	Clock Enable control over memory bus components

3.3.1 SDR/DDR SDRAM Controller Registers

Status Register **0xF0301000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			BANK1	XMON		BANK0	CHIPS		TYPE			WIDTH		STAT	

Field	Name	RW	Reset	Description
12	BANK1	RO	-	This returns part of the definition of the number of banks on each chip.
11-10	XMON	RO		Returns the number of exclusive access monitor resources implemented in the memory controller: 2'b00 = 0 monitors 2'b01 = 1 monitor 2'b10 = 2 monitors 2'b11 = 4 monitors.
9	BANK0	RO		This returns part of the definition of the number of banks on each chip.
8-7	CHIPS	RO		Returns the number of different chip selects that the memory controller supports: 2'b00 = 1 chip 2'b01 = 2 chips 2'b10 = 3 chips 2'b11 = 4 chips.
6-4	TYPE	RO		Returns the SDRAM that the memory controller supports: 3'b000 = SDR SDRAM 3'b001 = DDR SDRAM 3'b011 = Mobile DDR SDRAM 3'b010 = eDRAM 3'b1xx = Reserved. If Mobile DDR SDRAM or SDR SDRAM or an eDRAM is supported, the cas_half_cycle bit at address offset 0x14 is ignored.
3-2	WIDTH	RO		Returns the width of the external memory: 2'b00 = 16-bit 2'b01 = 32-bit 2'b10 = 64-bit 2'b11 = Reserved.
1-0	STAT	RO		Returns the state of the memory controller: 2'b00 = Config 2'b01 = Ready 2'b10 = Paused 2'b11 = Low_power.

Table 3.16 Memory Banks Chip Configuration

Memorybanks1 and Memorybanks0	Banks per memory chip
0	4
1	2 (Two memory banks per chip is applicable for only eDRAM configurations.)
2	Reserved
3	Reserved

Command Register **0xF0301004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													CMD		

The write-only register enables the memory controller to be traversed. The command register controls the programmer's view FSM. By writing to this register, the FSM can be traversed.

Field	Name	RW	Reset	Description
2-0	CMD	WO	-	Changes the state of the memory controller: 3'b000 = Go 3'b001 = Sleep 3'b010 = Wakeup 3'b011 = Pause 3'b100 = Configure 3'b111 = Active_Pause.

Note:

Active_Pause puts the memory controller into the Paused state without draining the arbiter queue. This enables you to enter low-power mode to change configuration settings such as memory frequency or timing register values without requiring coordination between masters in a multi-master system.

If the memory controller is put into low-power mode after using the Active_Pause command, you must not remove power from the memory controller because this results in data loss and violation of the AXI protocol. The memory controller does not issue refreshes while in the Config state. You must use low-power mode to make register updates because this ensures that the memory is put into self-refresh rather than entering the Config state when the memory contains valid data.

Direct Command Register 0xF0301008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									ECMD	CHNB		MCMD		BNKA	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		ADR													

The write-only direct_cmd Register passes commands to the external memory. The configuration of the direct_cmd Register enables you to write to any type of Mode register supported by the external memory device, and also to generate NOP, Prechargeall, and Auto-refresh commands.

The direct_cmd Register therefore enables any initialization sequence that an external memory device might require. The only timing information associated with the direct_cmd Register are the command delays defined in the timing registers. Therefore, if an initialization sequence requires additional delays between commands, they must be timed by the master driving the initialization sequence.

Field	Name	RW	Reset	Description
22	ECMD	WO	-	Extended memory command, see note after the table
21-20	CHNB	WO	-	Bits mapped to external memory chip address bits
19-18	MCMD	WO	-	Determines the command required, see note after the table
17-16	BNKA	WO	-	Bits mapped to external memory bank address bits when command is Mode_reg access
15-14	-			Reserved
13-0	ADR	WO	-	Bits mapped to external memory address bits [13:0] when command is Mode_reg access

Note:

Memory command encoding uses the ECMD bit concatenated to MCMD, therefore providing 3 bits as follows:

3’b000 = Prechargeall
3’b001 = Autorefresh
3’b010 = Modereg or Extended modereg access
3’b011 = NOP, for SDRAM only, Warning: If you have the NVM plug-in licensed, do not use the NOP command with NVM chip selects.
3’b100 = DPD

All other combinations are illegal and might cause undefined behavior.

A NOP command asserts all chip selects that are set as active_chips when the chip_nmbr is set to 0.

If chip_nmbr is set to 1 only cs_n[1] is asserted.
If chip_nmbr is set to 2 only cs_n[2] is asserted.
If chip_nmbr is set to 3 only cs_n[3] is asserted.

Configuration Register 0xF030100C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SR_EN	FP_TIME							FP_EN	ACTCH		QOS_MBITS			MBURST	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MBURST	STOP_MCLK	AUTO_PD	PD_PRD						AP_BIT	ROW_BITS			COL_BITS		

Field	Name	RW	Reset	Description
31	SR_EN[5]	R/W	0	Auto self-entry enable. Only if configured else read is undefined write as zero.
30-24	FP_TIME	R/W	0	Force precharge timeout count. Only valid if configured, else read is undefined write as zero.
23	FP_EN	F/W	0	Force precharge enable. Only valid if configured, else read undefined write as zero.
22-21	ACTCH	R/W	0	Enables the refresh command generation for the number of memory chips. It is only possible to generate commands up to and including the number of chips in the configuration that the status Register defines: 2'b00 = 1 chip 2'b01 = 2 chips 2'b10 = 3 chips 2'b11 = 4 chips.
20-18	QOS_MBITS	R/W	0	Encodes the four bits of the 8-bit AXI ARID that select one of the 16 QOS values: 3'b000 = ARID[3:0] 3'b001 = ARID[4:1] 3'b010 = ARID[5:2] 3'b011 = ARID[6:3] 3'b100 = ARID[7:4] 3'b101 = ARID[8:5] 3'b110 = ARID[9:6] 3'b111 = ARID[10:7]
17-15	MBURST	R/W	0x2	Encodes the number of data accesses that are performed to the SDRAM for each Read and Write command: 3'b000 = Burst 1 3'b001 = Burst 2 3'b010 = Burst 4 3'b011 = Burst 8 3'b100 = Burst 16. This value must also be programmed into the SDRAM mode register using the direct_cmd register at offset 0x8, and must match it.
14	STOP_MCLK[6]	R/W	0	When enabled, the memory clock is dynamically stopped when not performing an access to the SDRAM.
13	AUTO_PD	R/W	0	When this is set, the memory interface automatically places the SDRAM into power-down state by deasserting cke when the command FIFO has been empty for PD_PRD memory clock cycles.
12-7	PD_PRD	R/W	0	Number of memory clock cycles for AUTO_PD of the SDRAM. The PD_PRD programmed must be greater than the programmed cas latency
6	AP_BIT	R/W	0	Encodes the position of the auto-precharge bit in the memory address: 1'b0 = address bit 10 1'b1 = address bit 8.
5-3	ROW_BITS	R/W	0x4	Encodes the number of bits of the AXI address that comprise the row address: 3'b000 = 11 bits

[5] Do not enable the SR_EN and STOP_MCLK bits at the same time because it does not support this.
[6] To comply to the JEDEC standard, the auto_power_down and stop_mem_clock bits must not be enabled at the same time.

				3'b001 = 12 bits 3'b010 = 13 bits 3'b011 = 14 bits 3'b100 = 15 bits 3'b101 = 16 bits. The combination of row size, column size, BRC/RBC, and memory width must ensure that neither the MSB of the row address nor the MSB of the bank address exceed address range [27:0].
2-0	COL_BITS	R/W	0	Encodes the number of bits of the AXI address that comprise the column address: 3'b000 = 8 bits 3'b001 = 9 bits 3'b010 = 10 bits 3'b011 = 11 bits 3'b100 = 12 bits.

Refresh Period Register **0xF0301010**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REF_PRD														

Field	Name	RW	Reset	Description
14-0	REF_PRD	R/W	0x0A60	Memory refresh period in memory clock cycles

CAS Latency Register **0xF0301014**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												CAS_LAT			CAS_HALF

Field	Name	RW	Reset	Description
3-1	CAS_LAT	R/W	0x3	CAS latency in memory clock cycles.
0	CAS_HALF	R/W	0	Encodes whether the CAS latency is half a memory clock cycle more than the value given in bits [3:1]: 1'b0 = Zero cycles offset to value in [3:1]. b0 is forced to 0 in Mobile DDR, SDR, and eDRAM mode. 1'b1 = Half cycle offset to value in [3:1].

t_dqss Register **0xF0301018**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														t_dqss	

The read/write t_dqss Register writes to DQS in memory clock cycles.

Field	Name	RW	Reset	Description
1-0	t_dqss	R/W	0x1	Write to DQS in memory clock cycles

t_mrd Register 0xF030101C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									t_mrd						

The read/write t_mrd Register sets the mode register command time in memory clock cycles.

Field	Name	RW	Reset	Description
6-0	t_mrd	R/W	0x2	Sets mode register command time in memory clock cycles

t_ras Register 0xF0301020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												t_ras			

The read/write t_ras Register sets the RAS to precharge delay in memory clock cycles.

Field	Name	RW	Reset	Description
3-0	t_ras	R/W	0x7	Sets RAS to precharge delay in memory clock cycles

t_rc Register 0xF0301024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												t_rc			

The read/write t_rc Register sets the Active bank x to Active bank x delay in memory clock cycles.

Field	Name	RW	Reset	Description
3-0	t_rc	R/W	0xB	Sets Active bank x to Active bank x delay in memory clock cycles

t_rcd Register 0xF0301028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SCH_RCD			t_rcd		

The read/write t_rcd Register sets the RAS to CAS minimum delay in memory clock cycles.

Field	Name	RW	Reset	Description
5-3	SCH_RCD	R/W	0x3	Sets the RAS to CAS minimum delay in aclk cycles-3.
2-0	t_rcd	R/W	0x5	Sets the RAS to CAS minimum delay in memory clock cycles.

t_rfc Register **0xF030102C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						SCH_RFC					t_rfc				

The read/write t_rfc Register sets the auto-refresh command time in memory clock cycles.

Field	Name	RW	Reset	Description
9-5	SCH_RFC	R/W	0x10	Sets the autorefresh command time in aclk cycles-3.
4-0	t_rfc	R/W	0x12	Sets the auto-refresh command time in memory clock cycles.

t_rp Register **0xF0301030**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SCH_RP			t_rp		

The read/write t_rp Register sets the precharge to RAS delay in memory clock cycles.

Field	Name	RW	Reset	Description
5-3	SCH_RP	R/W	0x3	Sets the precharge to RAS delay in aclk cycles -3.
2-0	t_rp	R/W	0x5	Sets the precharge to RAS delay in memory clock cycles.

t_rrd Register **0xF0301034**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												t_rrd			

The read/write t_rrd Register sets the Active bank x to Active bank y delay in memory clock cycles.

Field	Name	RW	Reset	Description
3-0	t_rrd	R/W	0x2	Sets Active bank x to Active bank y delay in memory clock cycles

t_wr Register **0xF0301038**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														t_wr	

The read/write t_wr Register sets the write to precharge delay in memory clock cycles.

Field	Name	RW	Reset	Description
2-0	t_wr	R/W	0x3	Sets the write to precharge delay in memory clock cycles

t_wtr Register **0xF030103C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													t_wtr		

The read/write t_wtr Register sets the write to read delay in memory clock cycles.

Field	Name	RW	Reset	Description
2-0	t_wtr	R/W	0x2	Sets the write to precharge delay in memory clock cycles

t_xp Register **0xF0301040**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								t_xp							

The read/write t_wtr Register sets the write to read delay in memory clock cycles.

Field	Name	RW	Reset	Description
7-0	t_xp	R/W	0x1	Sets the exit power-down command time in memory clock cycles

t_xsr Register **0xF0301044**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								t_xsr							

The read/write t_xsr Register sets exit self-refresh command time in memory clock cycles.

Field	Name	RW	Reset	Description
7-0	t_xsr	R/W	0xA	Sets the exit self-refresh command time in memory clock cycles

t_esr Register **0xF0301048**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								t_esr							

The read/write t_xsr Register sets self-refresh command time in memory clock cycles.

Field	Name	RW	Reset	Description
7-0	t_esr	R/W	0x14	Sets the self-refresh command time in memory clock cycles

Memory_cfg2 Register **0xF030104C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					RD_DLY		TYPE			WIDTH		CKE_INIT	DQM_INIT	A_GT_M_SYNC	SYNC

The read/write memory_cfg2 Register determines the operating state of the memory controller and the memory. At reset the register value is set by the tie-off pins with the same names as the register names. After power-up you can change the register value through the APB interface.

Field	Name	RW	Reset	Description
10-9	RD_DLY	R/W	-	Sets the latency in clock cycles of the pad interface.
8-6	TYPE	R/W	-	Sets the memory type: 0 = SDR 1 = DDR 2 = eDRAM 3 = LPDDR (Mobile DDR). Note: It is only legal to program the memory type between SDR and (LP)DDR for a memory controller configuration that supports it.
5-4	WIDTH	R/W	-	Sets the width of the external memory: 2'b00 = 16-bit 2'b01 = 32-bit 2'b10 = 64-bit 2'b11 = Reserved. Note: Only a memory width that is legal for the memory controller can be programmed.
3	CKE_INIT	R/W	-	Sets the level for the cke outputs after reset.
2	DQM_INIT	R/W	-	Sets the level for the dqm outputs after reset.
1	A_GT_M_SYNC	R/W	-	Requires to be set HIGH when running the aclk and mclk synchronously but with aclk running faster than mclk. **** Should be "0" in this chip.***
0	SYNC	R/W	-	Set high when aclk and mclk are synchronous. **** Should be "1" in this chip.***

Memory_cfg3 Register **0xF0301050**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
			PRESCALE										MAX_OUT_REFS		

The read/write memory_cfg3 Register determines the operating state of the memory controller and the memory. At reset the register value is set by the tie-off pins with the same names as the register names. After power-up you can change the register value through the APB interface.

Field	Name	RW	Reset	Description
12-3	PRESCALE	R/W	0x0	Prescalar counter value.
2-0	MAX_OUT_REFS	R/W	0x7	Maximum number of outstanding refresh commands.

AXI ID n Configuration Register

0xF0301100+4*n

n=0~15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						QOS_MAX								QOS_MIN	QOS_EN

The read/write IDnCFG Registers are 16 registers that set the QoS and span address locations 0x100-0x200.

Field	Name	RW	Reset	Description
9-2	QOS_MAX	R/W	0	Sets a maximum QoS.
1	QOS_MIN	R/W	0	Sets a minimum QoS.
0	QOS_EN	R/W	0	Enables a QoS value to be applied to memory reads from address ID n.

Chip n Configuration Register

0xF0301200+4*n

n=0~3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															BRC_N_RBC
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_MATCH								ADDR_MASK							

The read/write CHIPnCFG Registers are registers that set up the external memory device configuration. The number of external chips supported, and therefore the number of these registers, depends on your configuration. They span address locations 0x200-0x300. There is one register per memory device. The registers configure the base address and address decoding method.

Field	Name	RW	Reset	Description
16	BRC_N_RBC	R/W	0	Selects the memory organization as decoded from the AXI address: 1'b0 = Row, bank, column organization 1'b1 = Bank, row, column organization.
15-8	ADDR_MATCH	R/W	0xFF	Comparison value for AXI address bits [31:24] to determine the chip that is selected.
7-0	ADDR_MASK	R/W	0	The mask for AXI address bits [31:24] to determine the chip that is selected: 1 = corresponding address bit is to be used for comparison.

3.3.2 DDR2 SDRAM Controller Registers

Status Register **0xF0302000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		BANK		XMON		BANK0	CHIPS		TYPE			WIDTH		STAT	

Field	Name	RW	Reset	Description
13-12	BANK	RO	-	Returns the maximum number of banks per memory chip. This is a rendered option, and is static for a given configuration: b00 = 4 banks b01 = 2 banks, not supported b10 = 1 banks, not supported b11 = 8 banks.
11-10	XMON	RO		Returns the number of exclusive access monitor resources implemented in the memory controller: 2'b00 = 0 monitors 2'b01 = 1 monitor 2'b10 = 2 monitors 2'b11 = 4 monitors.
9	-			Reserved
8-7	CHIPS	RO		Returns the number of different chip selects that the memory controller supports: 2'b00 = 1 chip 2'b01 = 2 chips 2'b10 = 3 chips 2'b11 = 4 chips.
6-4	TYPE	RO		Returns the type of SDRAM that the DDR2 DMC supports: b000-b100 = reserved b101 = DDR2 SDRAM b110-b111 = reserved.
3-2	WIDTH	RO		Returns the width of the external memory: 2'b00 = 16-bit 2'b01 = 32-bit 2'b10 = 64-bit 2'b11 = Reserved.
1-0	STAT	RO		Returns the state of the memory controller: 2'b00 = Config 2'b01 = Ready 2'b10 = Paused 2'b11 = Low_power.

Command Register **0xF0302004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													CMD		

Writing to the register enables the programmer's view FSM to be traversed.

Field	Name	RW	Reset	Description
2-0	CMD	WO	-	Changes the state of the memory controller: 3'b000 = Go 3'b001 = Sleep 3'b010 = Wakeup 3'b011 = Pause 3'b100 = Configure 3'b111 = Active_Pause.

Note:

Active_Pause command puts the DDR2 SDRAM Controller into the Paused state without draining the arbiter queue. This enables you to enter Low-power state to change configuration settings such as memory frequency or timing register values without requiring co-ordination between masters in a multi-master system.

If the DDR2 SDRAM Controller is put into Low-power state after using the Active_Pause command, you must not remove power from it because this results in data loss and violation of the AXI protocol.

The DDR2 DMC does not issue refreshes when in the Config state. It is recommended therefore that you use low-power mode to make register updates because this ensures that the memory is put into self-refresh rather than entering the Config state when the memory contains valid data.

If you entered the Paused state using the Active_Pause command, you must not attempt to move to the Config state by using the Configure command.

Direct Command Register 0xF0302008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										CHNB		MCMD		BNKA	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		ADR													

The write-only direct_cmd Register passes commands to the external memory. The configuration of the direct_cmd Register enables you to write to any type of Mode register supported by the external memory device, and also to generate NOP, Prechargeall, and Auto-refresh commands.

The direct_cmd Register therefore enables any initialization sequence that an external memory device might require. The only timing information associated with the direct_cmd Register are the command delays defined in the timing registers. Therefore, if an initialization sequence requires additional delays between commands, they must be timed by the master driving the initialization sequence.

Field	Name	RW	Reset	Description
21-20	CHNB	WO	-	Bits mapped to external memory chip address bits
19-18	MCMD	WO	-	Determines the command required: b00 = Prechargeall b01 = Autorefresh b10 = Modereg or Extended modereg access b11 = NOP.
17-16	BNKA	WO	-	Bits mapped to external memory bank address bits when command is Mode_reg access
15-14	-			Reserved
13-0	ADR	WO	-	Bits mapped to external memory address bits [13:0] when command is Mode_reg access

Configuration Register **0xF030200C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									ACTCH		QOS_MBITS			MBURST	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MBURST	STOP_MCLK	AUTO_PD	PD_PRD							ROW_BITS			COL_BITS		

Field	Name	RW	Reset	Description
22-21	ACTCH	R/W	0	Enables the refresh command generation for the number of memory chips. It is only possible to generate commands up to and including the number of chips in the configuration that the status Register defines: 2'b00 = 1 chip 2'b01 = 2 chips 2'b10 = 3 chips 2'b11 = 4 chips.
20-18	QOS_MBITS	R/W	0	Controls which ARID signals that the DDR2 SDRAM Controller uses when it selects the QoS value for the AXI read transfer: b000 = ARID[3:0] b001 = ARID[4:1] b010 = ARID[5:2] b011 = ARID[6:3] b100 = ARID[7:4] b101 = ARID[8:5] b110 = ARID[9:6] b111 = ARID[10:7].
17-15	MBURST	R/W	0x2	Encodes the number of data accesses that are performed to the DDR2 SDRAM for each Read and Write command: b010 = Burst 4. This value must also be programmed into the DDR2 SDRAM mode register using the Direct Command Register.
14	STOP_MCLK	R/W	0	When set to 1, the clk_out[MEMORIES-1:0] signals are dynamically stopped after the memories enter self-refresh mode.
13	AUTO_PD	R/W	0	When this is set, the memory interface automatically places the DDR2 SDRAM into power-down state by deasserting cke when the command FIFO has been empty for PD_PRD memory clock cycles.
12-7	PD_PRD	R/W	0	Number of memory clock cycles for auto power-down of the DDR2 SDRAM.
6	-			Reserved
5-3	ROW_BITS	R/W	0x4	Encodes the number of bits of the AXI address that comprise the row address: 3'b000 = 11 bits 3'b001 = 12 bits 3'b010 = 13 bits 3'b011 = 14 bits 3'b100 = 15 bits 3'b101 = 16 bits. b110-b111 = reserved.
2-0	COL_BITS	R/W	0	Encodes the number of bits of the AXI address that comprise the column address: b000 = Reserved. b001 = 9 bits. b010 = 10 bits. b011 = 11 bits. This means that A0-A9, and A11 are used for column address because A10 is a dedicated AP bit. b100-b111 = Reserved.

Refresh Period Register 0xF0302010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REF_PRD														

Field	Name	RW	Reset	Description
14-0	REF_PRD	R/W	0x0A2C	Memory refresh period in memory clock cycles

CAS Latency Register 0xF0302014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												CAS			

Field	Name	RW	Reset	Description
3-1	CAS	R/W	0xA	CAS latency in memory clock cycles.
-				Reserved

Write Latency Register 0xF0302018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													WRL		

The write_latency Register indicates the write latency in memory clock cycles.

Field	Name	RW	Reset	Description
2-0	t_dqss	RO	0x4	Write latency in memory clock cycles

t_mrd Register 0xF030201C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									t_mrd						

The read/write t_mrd Register sets the mode register command time in memory clock cycles.

Field	Name	RW	Reset	Description
6-0	t_mrd	R/W	0x2	Sets mode register command time in memory clock cycles

t_ras Register 0xF0302020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											t_ras				

The read/write t_ras Register sets the RAS to precharge delay in memory clock cycles.

Field	Name	RW	Reset	Description
4-0	t_ras	R/W	0xE	Sets RAS to precharge delay in memory clock cycles

t_rc Register 0xF0302024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											t_rc				

The read/write t_rc Register sets the Active bank x to Active bank x delay in memory clock cycles.

Field	Name	RW	Reset	Description
4-0	t_rc	R/W	0x12	Sets Active bank x to Active bank x delay in memory clock cycles

t_rcd Register **0xF0302028**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					SCH_RCD								t_rcd		

The read/write t_rcd Register sets the RAS to CAS minimum delay in memory clock cycles.

Field	Name	RW	Reset	Description
10-8	SCH_RCD	R/W	0x3	Sets the RAS to CAS minimum delay in clock cycles minus 3. It is used as a scheduler delay and values in the range 0-4 are supported.
7-3	-			Reserved
2-0	t_rcd	R/W	0x5	Sets the RAS to CAS minimum delay in memory clock cycles.

t_rfc Register **0xF030202C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCH_RFC								t_rfc						

The read/write t_rfc Register sets the auto-refresh command time in memory clock cycles.

Field	Name	RW	Reset	Description
14-8	SCH_RFC	R/W	0x21	Sets the Autorefresh command time in clock cycles minus 3. It is used as a scheduler delay.
7	-			Reserved
6-0	t_rfc	R/W	0x23	Sets the auto-refresh command time in memory clock cycles.

t_rp Register **0xF0302030**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					SCH_RP								t_rp		

The read/write t_rp Register sets the precharge to RAS delay in memory clock cycles.

Field	Name	RW	Reset	Description
10-8	SCH_RP	R/W	0x3	Sets the precharge to RAS delay in clock cycles, minus 3. It is used as a scheduler delay and values in the range 0-4 are supported.
3-0	t_rp	R/W	0x5	Sets the precharge to RAS delay in memory clock cycles.

t_rrd Register **0xF0302034**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												t_rrd			

The read/write t_rrd Register sets the Active bank x to Active bank y delay in memory clock cycles.

Field	Name	RW	Reset	Description
3-0	t_rrd	R/W	0x4	Sets Active bank x to Active bank y delay in memory clock cycles

t_wr Register **0xF0302038**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													t_wr		

The read/write t_wr Register sets the write to precharge delay in memory clock cycles.

Field	Name	RW	Reset	Description
2-0	t_wr	R/W	0x5	Sets the write to precharge delay in memory clock cycles

t_wtr Register **0xF030203C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													t_wtr		

The read/write t_wtr Register sets the write to read delay in memory clock cycles.

Field	Name	RW	Reset	Description
2-0	t_wtr	R/W	0x4	Sets the write to precharge delay in memory clock cycles

t_xp Register **0xF0302040**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								t_xp							

The read/write t_wtr Register sets the write to read delay in memory clock cycles.

Field	Name	RW	Reset	Description
7-0	t_xp	R/W	0x2	Sets the exit power-down command time in memory clock cycles

t_xsr Register **0xF0302044**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								t_xsr							

The read/write t_xsr Register sets exit self-refresh command time in memory clock cycles.

Field	Name	RW	Reset	Description
7-0	t_xsr	R/W	0x27	Sets the exit self-refresh command time in memory clock cycles

t_esr Register **0xF0302048**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								t_esr							

The read/write t_xsr Register sets self-refresh command time in memory clock cycles.

Field	Name	RW	Reset	Description
7-0	t_esr	R/W	0x14	Sets the self-refresh command time in memory clock cycles

Memory_cfg2 Register **0xF030204C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								WIDTH		BNK_BITS		CKE_INIT	DQM_INIT	CLK_CFG	

The read/write memory_cfg2 Register determines the operating state of the memory controller and the memory. At reset the register value is set by the tie-off pins with the same names as the register names. After power-up you can change the register value through the APB interface.

Field	Name	RW	Reset	Description
7-6	WIDTH	R/W	-	Encodes the physical memory width of the attached memory device. This is half the memory_width in the Memory Controller Status Register. The value is dependent on the rendered configuration of the implementation: b00 = 16-bit b01 = 32-bit b10 = 64-bit b11 = reserved.
5-4	BNK_BITS	R/W	-	Encodes the number of bits of the AXI address that comprise the bank address. The value is dependent on the rendered configuration of the implementation: b10 = 0 bits, not supported by the DDR2 SDRAM Controller b01 = 1 bits, not supported by the DDR2 SDRAM Controller b00 = 2 bits b11 = 3 bits.
3	CKE_INIT	R/W	-	State of cke when mresetn is de-asserted..
2	DQM_INIT	R/W	-	State of dqm[MEMBYTES-1:0] when mresetn is de-asserted.
1-0	CLK_CFG	R/W	-	Encodes the clocking scheme: b00 = aclk and mclk are asynchronous b01 = aclk and mclk are synchronous, and aclk is the same frequency or slower than mclk b10 = reserved b11 = aclk and mclk are synchronous, and aclk is greater than mclk.

Memory_cfg3 Register 0xF0302050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													REF_TIMEOUT		

The memory_cfg3 Register determines the number of acceptable outstanding refreshes on a chip before a refresh timeout occurs and refreshes are raised to the highest priority in the queue.

Field	Name	RW	Reset	Description
2-0	REF_TIMEOUT	R/W	0x7	Sets the number of acceptable outstanding refreshes on a chip before a refresh timeout occurs and refreshes are raised to the highest priority in the queue.

t_faw Register 0xF0302054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			SCH_FAW								t_faw				

The t_faw Register sets four bank activate time in memory clock cycles

Field	Name	RW	Reset	Description
12-8	SCH_FAW	R/W	0x1	t_faw in clock cycles, minus 3. Used as a scheduler delay.
7-5	-			Reserved
4-0	t_faw	R/W	0x1	Four-bank activate period in clock cycles.

AXI ID n Configuration Register 0xF0302100+4*n

n=0~15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						QOS_MAX								QOS_MIN	QOS_EN

The read/write IDnCFG Registers are 16 registers that set the QoS and span address locations 0x100-0x200.

Field	Name	RW	Reset	Description
9-2	QOS_MAX	R/W	0	Sets a maximum QoS.
1	QOS_MIN	R/W	0	Sets a minimum QoS.
0	QOS_EN	R/W	0	Enables a QoS value to be applied to memory reads from address ID n.

Chip n Configuration Register 0xF0302200+4*n

n=0~3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															BRC_N_RBC
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDR_MATCH								ADDR_MASK							

The read/write CHIPnCFG Registers are registers that set up the external memory device configuration. The number of external chips supported, and therefore the number of these registers, depends on your configuration. They span address locations 0x200-0x300. There is one register per memory device. The registers configure the base address and address decoding method.

Field	Name	RW	Reset	Description
16	BRC_N_RBC	R/W	0	Selects the memory organization as decoded from the AXI address: 1'b0 = Row, bank, column organization 1'b1 = Bank, row, column organization.
15-8	ADDR_MATCH	R/W	0xFF	Comparison value for AXI address bits [31:24] to determine the chip that is selected.
7-0	ADDR_MASK	R/W	0	The mask for AXI address bits [31:24] to determine the chip that is selected: 1 = corresponding address bit is to be used for comparison.

3.3.3 SDRAM PHY Registers

PHY Mode Control Register **0xF0304400**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													PD	SE_MODE	GDDR_MODE

Field	Name	RW	Reset	Description
2	PD	R/W	0	INPUT GATE POWER DOWN 0 – Normal 1 – I/O pad input gating mode
1	SE_MODE	R/W	0	DIFFERENTIAL STOBE MODE Differential strobe mode selection. 1'b1 : for Single Ended DQS mode. 1'b0 : for Differential DQS mode. DDR mode : 1'b1 DDR2 mode : 1'b0(recommended) / 1'b1 GDDR3 mode : 1'b1 Single Ended DQS mode.
0	GDDR_MODE	R/W	0	PHY mode selection DDR mode : 1'b0 DDR2 mode : 1'b0 GDDR3 mode : 1'b1

DLL Control Register **0xF0304404**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	LOCK										FLOCK	CLOCK	HALF	START	DLL_ON

Field	Name	RW	Reset	Description
14-5	LOCK	RO	-	Locked delay line encoding value. LOCK [9:2] : number of delay cells for coarse lock. LOCK [1:0] : control value for fine lock.
4	FLOCK	RO	-	Fine lock information. Specifies that DLL has fine locked the clock.
3	CLOCK	RO	-	Coarse lock information. Specifies that DLL has been locked
2	HALF	R/W	0	HIGH active start signal to turn on the low speed mode for DLL. If this bit is set, DLL can run at low speed(80MHz ~ 100MHz).
1	START	R/W	0	HIGH active start signal to make the DLL run and lock. This signal should be kept HIGH during normal operation. If this signal becomes LOW, DLL stops running. To re-run DLL, make this signal HIGH again. In the case of re-running, DLL loses previous lock information. Before ctrl_start is set, be sure that ctrl_dll_on is HIGH.
0	DLL_ON	R/W	0	HIGH active start signal to turn on the DLL. This signal should be kept HIGH for for normal operation. If this signal becomes LOW, DLL is turned off and ctrl_clock and ctrl_flock become HIGH. This bit should be kept set before ctrl_start is set to turn on the DLL.

Notes:

{CLOCK, FLOCK = 2'b00} : DLL is not locked.
{CLOCK, FLOCK = 2'b01} : Impossible value.
{CLOCK, FLOCK = 2'b10} : Locked and "phase offset error" is less than 160ps.
{CLOCK, FLOCK = 2'b11} : Locked and "phase offset error" is less than 80ps.

The reset value of READ ONLY Bits is provided as “-“ as the status of these registers immediately after reset is depends on PHY and does not have any significance. The significance of these is only after DLL lock operation is started.

DLL Phase Detector Configuration Register **0xF0304408**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INC								START_POINT							

Field	Name	RW	Reset	Description
15-8	INC	R/W	0	Increase amount of start point
7-0	START_POINT	R/W	-	Initial DLL lock start point. This is the number of delay cells and is the start point where "DLL" start tracing to be locked. Initial delay time is calculated by multiplying the unit delay of delay cell and this value.

Table 3.17 DLL Phase Detection Register Field Description

Frequency	START_POINT	INC
100MHz	0x10≤ ≤0x4F	0x10≤ ≤0x4F
133MHz	0x10≤ ≤0x3B	0x10≤ ≤0x3B
166MHz	0x10≤ ≤0x2F	0x10≤ ≤0x2F
200MHz	0x10≤ ≤0x28	0x10≤ ≤0x28
266MHz	0x10≤ ≤0x1E	0x10≤ ≤0x1E
333MHz	0x10≤ ≤0x18	0x10≤ ≤0x18
400MHz	0x10≤ ≤0x14	0x10≤ ≤0x14

Gate Control Register **0xF030440C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					ADB_EN	OFFSETC							SHIFTC		

Field	Name	RW	Reset	Description
10	ADB_EN	R/W	0	ADB ENABLE If this bit is set, PHY starts 32-bit CRC generation according to control and address signals.(ADCT[19:0], RAS, CAS, WE and CKE). If this bit is cleared, it stops CRC generation.
9-3	OFFSETC	R/W	0	GDOFFSET DDR/DDR2 : Required for DQS cleaning External Gate Feedback : 7'b0000000 Package Gate Feedback : refer to Application Note Gate offset amount for DDR/DDR2. If this field is fixed, this should not be changed during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. OFFSETC [6] = 1 :GATEout delay amount − OFFSETC[5:0] OFFSETC[6] = 0 :GATEout delay amount + OFFSETC [5:0]
2-0	SHIFTC	R/W	0	GDLY DDR/DDR2 : Required for DQS cleaning External Gate Feedback : 3'b110 Package Gate Feedback : refer to Application Note GATEout signal delay amount for DDR/DDR2. If this field is fixed, this should not be changed during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. 3’b000 : T/128(2.8125°shift) 3’b001 : T/64(5.625°shift) 3’b010 : T/32(11.25°shift) 3’b011 : T/16(22.5°shift) 3’b100 : T/8(45°shift) 3’b101 : T/4(90°shift) 3’b110 : T/2(180°shift) 3’b111 : T(360°shift) 180°shift (3'b110) can secure best timing margin for DQS cleaning for external gate feedback(refer to application note)

Read Data Slice 0 Control Register **0xF0304410**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									OFFSET0						

Field	Name	RW	Reset	Description
6-0	OFFSET0	R/W	0	RDOFFSET0 This field is used for debug purpose. 7'b0000000 for function mode. If this field is fixed, this should not be changed during operation. This value is valid only after RESYNC becomes HIGH and LOW. rd_slice_0 offset amount : OFFSET0[6] = 1 : 90° delay amount − OFFSET0 [5:0] OFFSET0 [6] = 0 : 90° delay amount + OFFSET0 [5:0]

Read Data Slice 1 Control Register 0xF0304414

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									OFFSET1						

Field	Name	RW	Reset	Description
6-0	OFFSET1	R/W	0	RDOFFSET1 This field is used for debug purpose. 7'b0000000 for function mode. If this field is fixed, this should not be changed during operation. This value is valid only after RESYNC becomes HIGH and LOW. rd_slice_1 offset amount : OFFSET1[6] = 1 : 90° delay amount − OFFSET1 [5:0] OFFSET1[6] = 0 : 90° delay amount + OFFSET1 [5:0]

Read Data Slice 2 Control Register 0xF0304418

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									OFFSET2						

Field	Name	RW	Reset	Description
6-0	OFFSET2	R/W	0	RDOFFSET2 This field is used for debug purpose. 7'b0000000 for function mode. If this field is fixed, this should not be changed during operation. This value is valid only after RESYNC becomes HIGH and LOW. rd_slice_2 offset amount : OFFSET2[6] = 1 : 90° delay amount − OFFSET2 [5:0] OFFSET2[6] = 0 : 90° delay amount + OFFSET2 [5:0]

Read Data Slice 3 Control Register 0xF030441C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									OFFSET3						

Field	Name	RW	Reset	Description
6-0	OFFSET3	R/W	0	RDOFFSET3 This field is used for debug purpose. 7'b0000000 for function mode. If this field is fixed, this should not be changed during operation. This value is valid only after RESYNC becomes HIGH and LOW. rd_slice_3 offset amount : OFFSET3[6] = 1 : 90° delay amount − OFFSET3 [5:0] OFFSET3[6] = 0 : 90° delay amount + OFFSET3 [5:0]

CLK Delay Register 0xF0304420

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									OFFSET2						

Field	Name	RW	Reset	Description
6-0	OFFSETD	R/W	0	CLK DLY This field is for debug purpose. If this field is fixed, this should not be changed during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. offset amount for 270°clock generation: OFSETD[6] = 1 : 270° delay amount − OFFSETD[5:0] OFFSETD[6] = 0 : 270° delay amount + OFFSETD[5:0]

DLL Force Lock Value Register 0xF0304424

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FORCE							

Field	Name	RW	Reset	Description
7-0	FORCE	R/W	0	DLL Forced Value This field is used instead of LOCK[9:2] to be found by the DLL only when DLL_ON is LOW ,i.e. If the DLL is off, this field is used to generate 270° clock and shift DQS by 90°.

ZQ Calibration Control Register 0xF0304428

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											PRD_CAL				PRD_CEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DRV_STR			TERM_DIS	TERM_VAL			PULL_DOWN			PULL_UP			ZQ	UPDATE	CAL_START

Field	Name	RW	Reset	Description
20-17	PRD_CAL	R/W	0x3	Periodic calibration Update counter load value
16	PRD_CEN	R/W	0	Periodic calibration Enable
15-13	DRV_STR	R/W	0	Driver strength selection.
12	TERM_DIS	R/W	0	Termination disable selection. 1 : termination disable. 0 : termination enable.
11-9	TERM_VAL	R/W	0	On-die-termination resistor value selection.
8-6	PULL_DOWN	R/W	0	Immediate control code for pull-down
5-3	PULL_UP	R/W	0	Immediate control code for pull-up
2	ZQ	R/W	0	Override IMPP[2:0]/IMPN[2:0] instead of calibration control code found after auto calibration.
1	UPDATE	R/W	0	Update calibration control code found by auto calibration
0	CAL_START	R/W	0	Auto calibration start signal.

Note: Periodic ZQ calibration bit should be enabled only in case when periodic ZQ calibration method is intended for ZQ calibration. Periodic ZQ calibration bit should be enabled simultaneously along with the CAL_START signal once the intial software calibration is finished.

ZQ Calibration Status Register 0xF030442C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								NMON			PMON			ERROR	END

Field	Name	RW	Reset	Description
7-5	NMON	RO	-	Control code found by auto calibration for pull-down.
4-2	PMON	RO	-	Control code found by auto calibration for pull-up.
1	ERROR	RO	-	Status indicating calibration completion with error.
0	END	RO	-	Status indicating calibration completion without error.

Read Delay Register 0xF0304430

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								NMON					ZQ	UPDATE	CAL_START

Field	Name	RW	Reset	Description
2-0	READ_DLY	R/W	0	Programmable read delay between controller and memory device 000 – Zero Delay 001 – 1 clock cycle delay 010 – 2 clock cycles delay 011 – 3 clock cycles delay

3.3.4 Miscellaneous Configuration Registers

SDR/DDR SDRAM Controller Configuration Register 0 (M0CFG0) **0xF0303000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYNC	RDW		TYPE			WIDTH		SYNC_OPT	SLOW						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
QOS_OVR															

Field	Name	RW	Reset	Description
31	SYNC	R/W	1	When HIGH, indicates aclk is synchronous to mclk. Otherwise, they are asynchronous. * Should be “1”
30-29	RDW	R/W	0	This value varies the delay permitted before the capture of read data into the memory clock domain from the memory device.
28-26	TYPE	R/W	0	Selects the memory type of the current configuration. Not all values are valid for a single configuration. You can override this value using the APB interface.
25-24	WIDTH	R/W	0	These configure the external memory width. Note: You can use each memory configuration at half of its memory width, by setting the memory_width tie-off, provided that: • the new memory width is not less than 16 bits • the effective memory width is not less than half the AXI interface width.
23	SYNC_OPT	R/W	0	Synchronizing option 0: two step synchronizer 1: one step synchronizer
22	SLOW	R/W	1	Additional Register Slice Insertion 0 : Slice inserted makes additional wait. 1 : Slice inserted will be bypassed
21-16	-			Reserved
15-0	QOS_OVR	R/W	0	When one or more bits are HIGH, coincident with arvalid and arready, and when the arid match bits are equivalent to the qos_override bit(s), then the QoS for the read access is forced to minimum latency.

SDR/DDR SDRAM Controller Configuration Register 1 (M0CFG1) 0xF0303004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									RST_BYPASS	EBIOFF	EBIGNT	EBIUSE	DQM_INIT	CKE_INIT	A_GT_M_SYNC

Field	Name	RW	Reset	Description
6	RST_BYPASS	R/W	0	This signal is used for ATPG testing only.
5	EBIOFF	R/W	0	External memory bus access backoff. The EBI backoff signal goes active HIGH when the EBI wants to remove the memory controller from the memory bus so that another memory controller can be granted the memory bus.
4	EBIGNT	R/W	0	External memory bus grant. The EBI grant signal goes HIGH when the EBI grants the external memory bus.
3	EBIUSE	R/W	0	Use EBI
2	DQM_INIT	R/W	0	The dqm output ports to the external memory reset to this value.
1	CKE_INIT	R/W	0	The cke output port to the external memory resets to this value.
0	A_GT_M_SYNC	R/W	0	When HIGH, indicates aclk is greater than mclk but is still synchronous.

DDR2 SDRAM Controller Configuration Register 0 (M1CFG0) 0xF0303010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYNC	BANK					WIDTH		SYNC_OPT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
QOS_OVR															

Field	Name	RW	Reset	Description
31	SYNC	R/W	1	When HIGH, indicates aclk is synchronous to mclk. Otherwise, they are asynchronous. * Should be “1”
30-29	BANK	R/W	0	Encodes number of bits of AXI address that comprise the bank address.
28-26	-			Reserved
25-24	WIDTH	R/W	0	These configure the external memory width. Note: You can use each memory configuration at half of its memory width, by setting the memory_width tie-off, provided that: • the new memory width is not less than 16 bits • the effective memory width is not less than half the AXI interface width.
23	SYNC_OPT	R/W	0	Synchronizing option 0: two step synchronizer 1: one step synchronizer
22-16	-			Reserved
15-0	QOS_OVR	R/W	0	When one or more bits are HIGH, coincident with arvalid and arready, and when the arid match bits are equivalent to the qos_override bit(s), then the QoS for the read access is forced to minimum latency.

DDR2 SDRAM Controller Configuration Register 1 (M1CFG1) 0xF0303014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									RST_BYPASS	EBIOFF	EBIGNT	EBIUSE	DQM_INIT	CKE_INIT	A_GT_M_SYNC

Field	Name	RW	Reset	Description
6	RST_BYPASS	R/W	0	This signal is used for ATPG testing only.
5-3	-			Reserved
2	DQM_INIT	R/W	0	The dqm output ports to the external memory reset to this value.
1	CKE_INIT	R/W	0	The cke output port to the external memory resets to this value.
0	-			Reserved

Common Configuration Register (COMMON) 0xF0303020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							OPT_AMAP								CSYSREQ1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
							CSYSREQ0						MODE_1CS	IO_SEL	AXI_SEL

Field	Name	RW	Reset	Description
24	OPT_AMAP	R/W	0	Optional Address mapping If set to high, araddr[30] and awaddr[30] is forced to 0
23-17	-			Reserved
16	CSYSREQ1	R/W	1	DDR2 SDRAM Controller system low-power request. This signal is a request to enter a Low-power state
15-9	-			Reserved
8	CSYSREQ0	R/W	1	SDR/DDR SDRAM Controller system low-power request. This signal is a request to enter a Low-power state
7-3	-			Reserved
2	MODE_1CS	R/W	0	In the DDR2 case, the 1 CS mode enabled. In this case, the CSN1 and ODT1 can be used to extra addresses.
1	IO_SEL	R/W	1	IO Interface mux selection signal between SDRAM PHY and SDRAM Controller 0: SDR/DDR SDRAM Controller 1: DDR2 SDRAM Controller
0	AXI_SEL	R/W	1	AXI Interface mux selection signal between Memory Bus and SDRAM Controller 0: SDR/DDR SDRAM Controller 1: DDR2 SDRAM Controller

SDRAM PHY Control Register (PHYCTRL) 0xF0303024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
							PHY_SEL							FNC_FB	

Field	Name	RW	Reset	Description
8	PHY_SEL	R/W	0	0 : DDR2 1 : SDR/mDDR/DDR
7-2	-			Reserved
1-0	FNC_FB	R/W	0	Function Feedback Test Enable

SDRAM PHY Status Register (PHYSTAT) 0xF0303028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADB_CHK															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADB_CHK															

Field	Name	RW	Reset	Description
31-0	ADB_CHK	RO	-	

SDRAM IO Control Register (IOCON) 0xF030302C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	IO_MODE	GATE_PD	CKB_PD	CK_PD	CKE_PD	WEN_PD	RAS_PD	CAS_PD	ODT_PD	CSN_PD	ADD_PD	BA_PD	DM_PD	DQ_PD	DQS_PD

Field	Name	RW	Reset	Description
14	IO_MODE	R/W	0	DRAM controller PAD type 0 : CMOS (SDR/DDR/mDDR) 1 : SSTL (DDR2)
13	GATE_PD	R/W	0	Gate signal Power down
12	CKB_PD	R/W	0	Ckb signal Power down
11	CK_PD	R/W	0	Ck signal Power down
10	CKE_PD	R/W	0	Cke signal Power down
9	WEN_PD	R/W	0	Wen signal Power down
8	RAS_PD	R/W	0	Ras signal Power down
7	CAS_PD	R/W	0	Cas signal Power down
6	ODT_PD	R/W	0	Odt signal Power down
5	CSN_PD	R/W	0	Csn signal Power down
4	ADD_PD	R/W	0	Address signal Power down
3	BA_PD	R/W	0	Ba signal Power down
2	DM_PD	R/W	0	Dqm signal Power down
1	DQ_PD	R/W	0	Dq signal Power down
0	DQS_PD	R/W	0	Dqs signal Power down

3.3.5 Memory Bus Configuration Registers

Memory Protection Register (TZPROT) **0xF0305000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
															ACC

Field	Name	RW	Reset	Description
0	ACC	R/W	1	External Memory Access Enable 1 for Enabled, 0 for Disabled

Clock Control Register 0 (CKDOWN) **0xF0305004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
													CD_DDR2	CD_SDDR	CD_MBUS

Field	Name	RW	Reset	Description
2	CD_DDR2	R/W	0	Clock down over DDR2 SDRAM Controller, mclk_m1
1	CD_SDDR	R/W	0	Clock down over SDR/DDR SDRAM Controller, mclk_m0
0	CD_MBUS	R/W	0	Clock down over Memory Bus Clock, aclk

4 MISC.

Table 4.1 Core Bus Configuration Register Map (Base Address = 0xF0101000)

Name	Address	Type	Reset	Description
CORECFG	0xF0101000	R/W	0x90000000	Core Bus Configuration Register

Table 4.2 Virtual MMU Table Register Map (Base Address = 0xF0600000)

Name	Address	Type	Reset	Description
REGION0	0x00	R/W	-	Configuration Register for Region 0
REGION1	0x04	R/W	-	Configuration Register for Region 1
REGION2	0x08	R/W	-	Configuration Register for Region 2
REGION3	0x0C	R/W	-	Configuration Register for Region 3
REGION4	0x10	R/W	-	Configuration Register for Region 4
REGION5	0x14	R/W	-	Configuration Register for Region 5
REGION6	0x18	R/W	-	Configuration Register for Region 6
REGION7	0x1C	R/W	-	Configuration Register for Region 7
TABBASE	0x20000000	R	-	MMU Table Base Address

4.1 Core Bus Configuration Registers

Core Bus Configuration Register (CORECFG) **0xF0101000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ROM		RAM		TLB				SLBUS	VBHPRT	IBHPRT	SBHPRT	MBHPRT	DBHPRT	GBHPRT	CBHPRT
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								IRQAVSYNCEN	INITSYNCEN	CP15DIS			CP15DIS	REMAP	

Field	Name	RW	Reset	Description
1-0	REMAP	RW	0x0	Remap configuration register Refer to 1.2 Address Map for more information about address remapping.
4-2	-	-	-	Reserved
5	CP15DIS	R/W	0x0	Disables write access to some system control processor registers
6	INITSYNCEN	R/W	0x0	Synchronize VIC interface
7	IRQAVSYNCEN	R/W	0x0	Synchronize IRQADDRV signal
15-8	-	-	-	Reserved
16	CBHPRT	R/W	0x0	Core Bus HPROT2
17	GBHPRT	R/W	0x0	Graphic Bus HPROT2
18	DBHPRT	R/W	0x0	DDI Bus HPROT2
19	MBHPRT	R/W	0x0	Memory Bus HPROT2
20	SBHPRT	R/W	0x0	SMU HPROT2
21	IBHPRT	R/W	0x0	IO Bus HPROT2
22	VBHPRT	R/W	0x0	Video Bus HPROT2
23	SLBUS	R/W	0x0	Slow Bus Indicator ***In the case of memory bus clock being under 180MHz @ 1.2V, if SLBUS is '1', the memory access latency can be reduced.***
25-24	-			Reserved
27-26	TLB	R/W	0x0	Add TLB Read Wait State
29-28	RAM	R/W	0x1	Add RAM Read Wait State
31-30	ROM	R/W	0x2	Add ROM Read Wait State

4.2 Virtual MMU Table

Region Configuration Register (REGIONx) **0xF0600000~0xF060001C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SA												-	0	-	SIZE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SIZE				AP		EN	DO				-	C	B	10b	

Field	Name	RW	Reset	Description
2	B	R/W	0x0	Bufferable Register * Defined same as ARM1176JZF-S
3	C	R/W	0x0	Cacheable Register * Defined same as ARM1176JZF-S
8 ~ 5	DO	R/W	0x0	Domain Register * Defined same as ARM1176JZF-S
9	EN	R/W	0x0	Region Enable Register
11 ~ 10	AP	R/W	0x0	Access Permission Register * Defined same as ARM1176JZF-S
16 ~ 12	SIZE	R/W	0x0	Size Register 0x13 : 1MB 0x14 : 2MB … 0x1E : 2GB 0x1F : 4GB
31 ~ 20	SA	R/W	0x0	Section Base Address * Defined same as ARM1176JZF-S

* All the bit field except for SZ and EN are same as defined in ARM1176JZF-S technical reference manual.

The REGION7 has the highest priority and the REGION0 has the lowest priority.

After setting according register, the address of TABBASE should be written into TLB base address for ARM1176JZF-S.

PART5 – IO BUS
TCC8900

High Performance and Low-Power Processor
For Digital Media Applications

Rev. 1.03

Nov 06, 2009

Telechips

Revision History

Date	Revision	Description
2008-12-11	0.00	Initial release
2009-01-21	0.01	- Replace “Audio DMA” to “ADMA” in SD/SDIO/MMC/CE-ATA Host controller - Change “ATAP” to “ATAPI” in IDE controller - Exchange EDI_XD[3:0] with EDI_XD[11:8] in EDI Configuration - Change bit width of EDI_XD&EDI_OEN in SMC - Add “synchronize txf empty signal” of UPCR3 in UTMI control register - Change “TSDI Input Data” to “TSIF Enable” in TSIF register descriptions - Add register description section to Test Block - Add a figure of main waveform to TSADC interface - Add warnings about TxFIFO number in USBOTG Device Endpoint Control registers. - Change the register description table of memory stick and insert a picture related to byte order mode - Revice SD/MMC base address in overall address map table - Revice OTGMAP description in OTGCR
2009-02-25	0.02	- Change the base address of IOBUS Config. Registers, OTGID, OTGCR and UPCR0/1/2/3 from 0xF05F0000 to 0xF05F5000 - Add TSIF CH1 base address description - Change register descriptions for PORTDLY, SDPORTCTRL and SDPORTDLY0/1/2/3
2009-06-08	0.03	- Add some descriptions about EDI configuration according to boot mode - Change the base address of GPSB block to 0xF0536000 - Change address values of TSADC block registers- - Modity Table 26.3 Analog Input Selection Table for 4-wire touch screen panels and Table 26.3 Analog Input Selection Table for Pressure Measurement - Modifiy the description of IOBUS AHB2AXI Control Register (IO_A2X). - Add OTG Device OUT Endpoint 0 Transfer Size Register (DOEPTSIZ0) - Add SATA BIST registers(RXBISTPD, RXBISTD1, RXBISTD2, TXBISTPD, TXBISTD1, TXBISTD2, BISTCR, BISTFCTR, BISTSR, BISTDECR) - Modify the block diagram & remove note1,2 in TSADC interface - Modify UPCR3 register in USBOTG - Modify “SPDIF Transmit Channel Status Control Register” in Audio DMA and SPDIF - Add INPOL register in remote control interface - Change initial value of RTCIM register in RTC block
2009-08-07	1.00	- minor changes - Change default value of HFIR in USBOTG - Change PMWKUP generation condition in RTC - Change descriptions about CLK DIVIDER register in remote control interface - Change descriptions about SATA_PHY_I2C_REG7 in SATA Host interface
2009-09-04	1.01	- Update description of SATA_PHY_I2C_REGA in SATA host interface - Update register description about "IOBUS AHB Clock Enable Register 0(HCLKEN0)" in "IOBUS Configuration Registers". - Update description about device connections related to EDI
2009-10-09	1.02	- Update description of IR Read Data register in remote control interface - Update description of USB 1.1 Host Configuration Register (USB11H) in USB 1.1 Host Controller & Transceiver
2009-11-06	1.03	- Remove all descriptions about CAN controller - Update descriptions about "External Chip Select Configuration Register" in SMC - Update the block diagram of USB 1.1 Host Controller & Transceiver - Update the feature of USB OTG Controller

TABLE OF CONTENTS

Contents

1 Introduction 1-1
2 Bus Architecture 2-3
3 Address and Register Map 3-5
4 Memory Stick Host Controller 4-7
4.1 Overview 4-7
4.2 Register Descriptions 4-10
4.3 Operation 4-30
4.3.1 Operation during Reset 4-30
4.3.2 Operation Flow 4-30
4.3.3 Communication Method to Start up smshc through Host CPU 4-30
4.3.4 Communication Method to Start up smshc through I-CON 4-33
4.3.5 Interface Mode Switching Sequence 4-34
4.4 Timing Diagram 4-37
4.4.1 Serial Interface Mode 4-37
4.4.2 4-bit Parallel Interface Mode 4-39
4.4.3 8-bit Parallel Interface Mode 4-41
4.5 I-CON Block 4-44
4.5.1 Control of smshc 4-44
4.5.2 Self-run 4-44
4.5.3 Instruction 4-44
4.5.4 Data Transfer 4-49
5 SD/SDIO/MMC/CE-ATA Host Controller 5-53
5.1 Overview 5-53
5.2 Block Diagram of Each Host Controller 5-53
5.3 Register Description 5-53
5.4 Timing Diagram 5-84
5.4.1 Command Format 5-84
5.4.2 Timing of the Transaction in SD Mode 5-84
5.4.3 Timing of the Transaction in CE-ATA Mode 5-86
6 NFC 6-89
6.1 Function Description 6-89
6.2 Register Description 6-90
7 Static Memory Controller(SMC) 7-105
7.1 Overview 7-105
7.2 Register Description 7-107
7.3 Address Mode of SMC 7-112
7.3.1 Normal Address mode 7-112
7.3.2 Indirect Address Mode 7-113
7.3.3 Wrapping Address Mode 7-114
8 EDI (External Device Interface) 8-115
8.1 Overview 8-115
8.2 Registers Description 8-117
9 IDE 9-123
9.1 Overview 9-123
9.2 Register Description 9-123
10 SATA Host Interface 10-137
10.1 Overview 10-137
10.2 SATA Host Bus Interface 10-137
10.2.1 Bus Interface Overview 10-138
10.2.2 Bus Interface FIS Decomposition 10-139
10.2.3 Bus Interface FIS Construction 10-139
10.2.4 Programmed I/O Operation 10-140
10.2.4.1 Read Transfer (Device to Host) 10-140
10.2.4.2 Write Transfer (Host to Device) 10-140
10.2.4.3 Slow Device Access 10-140
10.2.5 Direct Memory Access Operation 10-140
10.2.5.1 DMA Initialization 10-141
10.2.5.2 DMA Read Transfer (Device to Host) 10-141
10.2.5.3 DMA Write Transfer (Host to Device) 10-141
10.2.5.4 DMA Requirements 10-142
10.2.5.5 DMA Termination 10-142
10.2.5.6 First-Party DMA 10-143
10.2.6 Interrupt Control 10-144
10.2.7 Register Access 10-144
10.2.8 AHB Error Conditions 10-144

10.2.9 Bus Interface Power Management 10-145
10.2.10 Hot-Plug 10-145
10.2.11 PHY and Link Control 10-145
10.2.12 Reset Conditions 10-145
10.2.13 BIST Operation 10-145
10.2.13.1 Loopback Device 10-146
10.2.13.2 Loopback Initiator 10-146
10.3 SATA Host Transport Layer 10-148
10.3.1 Transport Layer FIS Reception 10-148
10.3.2 Transport Layer FIS Transmission 10-149
10.3.3 Error Handling 10-150
10.3.4 Transfer Termination 10-150
10.3.5 Module Description 10-151
10.3.5.1 Receive/Transmit FIFO (Rx/TxFIFO) 10-151
10.3.5.2 Transport Check (TCHK) 10-151
10.3.5.3 Transport State Machine (TSM) 10-152
10.4 SATA Host Link Layer 10-152
10.4.1 Link Layer Overview 10-152
10.4.1.1 Features 10-152
10.4.1.2 Operational Overview 10-153
10.4.2 PHY Interface 10-153
10.4.2.1 Rx and Tx Data 10-153
10.4.2.2 8b/10b Encoding and Decoding 10-154
10.4.3 PHY Initialization Details 10-154
10.4.3.1 Link Layer Tx OOB Initialization Sequence Details 10-154
10.4.3.2 Link Layer Tx OOB Sequence Generation 10-155
10.4.3.3 Link Layer Rx OOB Sequence Detection 10-155
10.4.4 Link Layer Power Management Details 10-156
10.5 Register Descriptions 10-157
10.5.1 SATA Host Controller Registers 10-158
10.5.2 SATA PHY Registers 10-178
11 AUDIO DMA Controller 11-183
11.1 Overview 11-183
11.1.1 AUDIO DMA 11-183
11.1.2 DAI & CDIF 11-183
11.1.3 SPDIF 11-185
11.2 Register Description 11-187
11.2.1 Audio DMA Register 11-189
11.2.2 DAI Register 11-198
11.2.3 CDIF Registers 11-207
11.2.4 SPDIF Registers 11-209
11.2.5 Audio Data Format 11-222
12 DAI & CD Interface 12-225
12.1 DAI (Digital Audio Interface) 12-225
12.2 CDIF (CD Media Interface) 12-232
13 SPDIF Transmitter 13-237
13.1 Overview 13-237
13.2 Register Descriptions 13-238
13.3 Operation & Timing Diagram 13-243
13.3.1.1 Resetting 13-243
13.3.1.2 Selecting Transmit Data Rate 13-243
13.3.1.3 Selecting Data Format 13-243
13.3.1.4 Setting up Channel Status Bits 13-243
13.3.1.5 Setting up User Data Bits 13-243
13.3.1.6 Preparing Sample Buffer 13-243
13.3.1.7 Start Transmission 13-244
14 USB 1.1 Host Controller & Transceiver 14-245
14.1 Overview 14-245
14.2 Register Description for USB 1.1 Host Controller & Transceiver 14-247
15 USB 2.0 OTG Controller 15-251
15.1 Overview 15-251
15.2 Register Description for USB 2.0 OTG Controller 15-253
15.3 Register Description for UTMI (USB PHY) 15-325
15.4 Programming Model 15-330
15.4.1 Overview 15-330
15.4.2 Core Initialization 15-330
15.4.3 Host Initialization 15-331
15.4.4 Device Initialization 15-331
15.5 Modes of Operation 15-332

15.5.1 DMA Mode 15-332
15.5.1.1 Transfer-Level Operation 15-332
15.5.1.2 Transaction-Level Operation 15-332
15.5.2 Slave Mode 15-332
15.5.2.1 Transaction-Level Operation 15-332
15.5.2.2 Pipelined Transaction-Level Operation 15-334
15.5.3 Thresholding in DMA Mode 15-335
15.6 Host Programming Model 15-336
15.6.1 Channel Initialization 15-336
15.6.2 Halting a Channel 15-336
15.6.3 Ping Protocol 15-337
15.6.4 Sending a Zero-Length Packet 15-337
15.6.5 Operational Model 15-338
15.6.5.1 Writing the Transmit FIFO in Slave Mode 15-338
15.6.5.2 Reading the Receive FIFO in Slave Mode 15-338
15.6.5.3 Bulk and Control OUT/SETUP Transactions in Slave Mode 15-339
15.6.5.4 Bulk and Control IN Transactions in Slave Mode 15-341
15.6.5.5 Bulk and Control OUT/SETUP Transactions IN DMA Mode 15-342
15.6.5.6 Bulk and Control IN Transactions in DMA Mode 15-344
15.6.5.7 Control Transactions in Slave Mode 15-345
15.6.5.8 Control Transactions in DMA Mode 15-345
15.6.5.9 Interrupt OUT Transactions in Slave Mode 15-345
15.6.5.10 Interrupt IN Transactions in Slave Mode 15-348
15.6.5.11 Interrupt OUT Transactions in DMA Mode 15-349
15.6.5.12 Interrupt IN Transactions in DMA Mode 15-351
15.6.5.13 Isochronous OUT Transactions in Slave Mode 15-352
15.6.5.14 Isochronous IN Transactions in DMA Mode 15-354
15.6.5.15 Bulk and Control OUT/SETUP Split Transactions in Slave Mode 15-354
15.6.5.16 Bulk and Control IN Split Transactions in Slave Mode 15-356
15.6.5.17 Bulk and Control OUT/SETUP Split Transactions in DMA Mode 15-357
15.6.5.18 Bulk/Control IN Split Transactions in DMA Mode 15-359
15.6.5.19 Interrupt OUT Split Transactions in Slave Mode 15-360
15.6.5.20 Interrupt IN Split Transactions in Slave Mode 15-362
15.6.5.21 Interrupt OUT Split Transactions in DMA Mode 15-363
15.6.5.22 Interrupt IN Split Transactions in DMA Mode 15-365
15.6.5.23 Isochronous OUT Split Transactions in Slave Mode 15-366
15.6.5.24 Isochronous IN Split Transactions in Slave Mode 15-368
15.6.5.25 Isochronous OUT Split Transactions in DMA Mode 15-369
15.6.5.26 Isochronous IN Split Transactions in DMA Mode 15-371
15.6.6 Selecting the Queue Depth 15-373
15.6.7 Handling Babble Conditions 15-373
15.7 Device Programming Model 15-374
15.7.1 Endpoint Initialization 15-374
15.7.1.1 Initialization on USB Reset 15-374
15.7.1.2 Initialization on Enumeration Completion 15-374
15.7.1.3 Initialization on SetAddress Command 15-374
15.7.1.4 Initialization on SetConfiguration/SetInterface Command 15-375
15.7.1.5 Endpoint Activation 15-375
15.7.1.6 Endpoint Deactivation 15-375
15.7.1.7 Device DMA/Slave Mode Initialization 15-375
15.7.2 Operational Model 15-376
15.7.2.1 SETUP and OUT Data Transfers 15-376
15.7.2.2 IN Data Transfers 15-387
15.7.2.3 Control Transfers 15-402
15.7.3 Handling Babble Conditions 15-406
15.7.4 Worst Case Response Time 15-407
15.7.5 Choosing the Value of GUSBCFG.USBTrdTim 15-408
15.8 OTG Programming Model 15-408
15.8.1 A-Device Session Request Protocol 15-408
15.8.2 B-Device Session Request Protocol 15-410
15.8.3 A-Device Host Negotiation Protocol 15-411
15.8.4 B-Device Host Negotiation Protocol 15-412
15.8.5 Clock Gating 15-413
15.8.5.1 Host Mode Suspend and Resume With Clock Gating 15-413
15.8.5.2 Host Mode Suspend and Remote Wakeup With Clock Gating 15-413
15.8.5.3 Host Mode Session End and Start With Clock Gating 15-414
15.8.5.4 Host Mode Session End and SRP With Clock Gating 15-414
15.8.5.5 Device Mode Suspend and Resume With Clock Gating 15-415
15.8.5.6 Device Mode Suspend and Remote Wakeup With Clock Gating 15-415

15.8.5.7 Device Mode Session End and Start With Clock Gating 15-415
15.8.5.8 Device Mode Session End and SRP With Clock Gating 15-416
15.9 Miscellaneous Topics 15-416
15.9.1 Data FIFO RAM Allocation 15-416
15.9.1.1 Device Mode 15-416
15.9.1.2 Host Mode 15-418
15.9.1.3 Calculating the Total FIFO Size for OTG 15-418
15.9.2 Dynamic FIFO Allocation 15-423
15.9.3 Core Interrupt Handler 15-424
16 EHI 16-425
16.1 Overview 16-425
16.2 Register Description 16-425
16.3 Operation 16-433
16.3.1 Access to EHI registers 16-433
16.3.2 Access to the On-chip System Bus 16-435
16.3.3 Entrance to the Critical Section Using the Semaphore Register 16-437
16.3.4 Interrupt Request 16-437
17 GPSB (General Purpose Serial Bus) 17-439
17.1 Functional Description 17-439
17.2 Feature 17-439
17.3 Register Description 17-439
17.4 GPSB Timing Diagram 17-454
17.5 MPEG2-TS Interface 17-455
18 TSIF(The transport Stream Interface) 18-457
18.1 Overview 18-457
18.2 Register Description 18-459
19 Remote Control Interface 19-463
19.1 Functional Description 19-463
19.2 Register Description 19-465
20 I2C Controller 20-469
20.1 Functional Description 20-469
20.2 I2C Slave Function 20-469
20.3 Related Blocks 20-470
20.4 Register Description 20-470
21 UART 21-481
21.1 Overview 21-481
21.2 Operation Modes 21-483
21.2.1 AFC (Auto Flow Control) in RX Operation 21-483
21.2.2 AFC (Auto Flow Control) in TX Operation 21-484
21.2.3 Operation with General DMA 21-484
21.2.4 RX Interrupt in DMA Transfer 21-485
21.2.4.1 TX Interrupt in DMA Transfer 21-486
21.2.5 RX Operation with DMA and AFC 21-487
21.2.6 TX Operation with DMA and AFC 21-488
21.3 Register Description 21-489
21.3.1 UART Controller Register 21-490
21.3.2 Port Mux Register 21-504
22 DMA CONTROLLER 22-505
22.1 Overview 22-505
22.2 Register Description 22-507
23 RTC (Real-Time Clock) 23-519
23.1 Overview 23-519
23.2 Function Description 23-520
23.2.1 System Clock Frequency Control 23-520
23.2.2 System Power Operation 23-520
23.2.3 Backup Battery Operation 23-520
23.2.4 Alarm Function 23-520
23.3 RTC Operation 23-521
23.3.1 Boot-Up Sequence 23-521
23.3.2 RTC Time Setting 23-521
23.3.3 RTC Alarm Time Setting 23-521
23.3.4 RTCPEND Clear 23-522
23.3.5 RTC Operation 23-522
23.3.6 Crystal Oscillator Circuit 23-523
23.4 Programmer's model 23-525
23.4.1 Register memory map 23-525
23.4.2 Register Description 23-525
24 TSADC Interface 24-533
24.1 Overview 24-533

24.1.1 A/D Conversion Time 24-533
24.1.2 Touch Screen Interface Mode 24-534
24.1.2.1 Normal Conversion Mode (AUTO = 0, XY_PST = 0) 24-534
24.1.2.2 Separate X/Y position Conversion Mode (AUTO = 0, XY_PST = control) 24-534
24.1.2.3 Auto(Sequential) X/Y position Conversion Mode (AUTO = 1, XY_PST = 0) 24-534
24.1.2.4 Waiting Interrupt Mode (ADCTSC = 0x00D3) 24-535
24.1.2.5 Stand-By Mode 24-535
24.2 TSADC Controller Register Description 24-535
24.3 Programing Note 24-540
25 ECC 25-543
25.1 Functional Description 25-543
25.2 Register Description 25-544
25.3 ECC Coding Sequence 25-558
25.3.1 SLC ECC Encoding Sequence 25-558
25.3.2 SLC ECC Decoding Sequence 25-559
25.3.3 MLC ECC4/8/12/14/16 Encoding Sequence 25-560
25.3.4 MLC ECC4/8/12/14/16 Decoding Sequence 25-561
26 MPEFEC 26-563
26.1 Overview 26-563
26.2 Block Diagram 26-565
26.3 Register Description 26-565
27 IOBUS Configuration Registers 27-569

Figures

Figure 2.1 The I/O Hardware Bus Architecture 2-3
Figure 4.1 The Components in SMSHC_I 4-7
Figure 4.2 Memory Stick Host Controller Block Diagram 4-9
Figure 4.3 Memory Stick Byte Order Mode 4-29
Figure 4.4 Communication Example for Starting up smshc through CPU 4-32
Figure 4.5 Communication Example for starting up smshc through I-CON 4-33
Figure 4.6 Sequence to Switch Interface Mode 4-34
Figure 4.7 Serial Protocol TPC Transfer State (BS1) 4-37
Figure 4.8 Serial Protocol Data Transfer State (BS3) 4-37
Figure 4.9 Serial Protocol Data Transfer State (BS2) 4-38
Figure 4.10 Serial Protocol Handshake State 4-38
Figure 4.11 Serial Protocol INT Transfer State 4-38
Figure 4.12 4-bit Parallel Protocol TPC Transfer State 4-39
Figure 4.13 4-bit Parallel Protocol Data Transfer State (BS3) 4-39
Figure 4.14 4-bit Parallel Protocol Data Transfer State (BS2) 4-40
Figure 4.15 4-bit Parallel Protocol Handshake State 4-40
Figure 4.16 4-bit Parallel Protocol INT Transfer State 4-41
Figure 4.17 8-bit Parallel Protocol TPC Transfer State 4-41
Figure 4.18 8-bit Parallel Protocol Data Transfer State (BS3) 4-42
Figure 4.19 8-bit Parallel Protocol Data Transfer State (BS2) 4-42
Figure 4.20 8-bit Parallel Protocol Handshake State 4-43
Figure 4.21 8-bit Parallel Protocol INT Transfer State 4-43
Figure 5.1 SD/SDIO/MMC/CE-ATA Block Diagram 5-53
Figure 5.2 An Example of ADMA2 Data Transfer 5-83
Figure 5.3 32-bit Address Descriptor Table 5-83
Figure 5.4 SDIO/SD – Write Interrupt Cycle Timing 5-85
Figure 5.5 SDIO/SD – Read Interrupt Cycle Timing 5-85
Figure 5.6 SDIO/SD – Multiple Block 4-Bit Write Interrupt Cycle Timing 5-85
Figure 5.7 SDIO/SD – Multiple Block 4-Bit Read Interrupt Cycle Timing 5-86
Figure 5.8 Multiple Register Read (CMD60)Transaction 5-86
Figure 5.9 Multiple Register Write (CMD60)Transaction 5-86
Figure 5.10 Single Block Read (CMD61) Transaction 5-87
Figure 5.11 Multiple Block Read (CMD61) Transaction 5-87
Figure 5.12 Single Block Write (CMD61) Transaction 5-87
Figure 5.13 Multiple Block Write (CMD61) Transaction 5-88
Figure 6.1 NAND Flash Controller Block Diagram 6-89
Figure 6.2 An Example of Interface for Two NAND Flash Memory 6-90
Figure 6.3 Example of Address/Command Writing Operation 6-93
Figure 6.4 Timing Diagram of Read/Write Enanle Signal 6-97
Figure 6.5 Timing Diagram of NAND Page Program 6-101
Figure 6.6 A Sequence of NAND Page Program 6-101
Figure 6.7 Timing Diagram of NAND Page Read 6-102
Figure 6.8 A Sequence of NAND Page Read 6-102

Figure 6.9 Example of Multiple Bytes/Half Word Write Operation 6-103
Figure 6.10 Example of Multiple Bytes/Half Word Read Operation 6-104
Figure 7.1 SMC Block Diagram 7-105
Figure 7.2 The Interface from SMC to External Static Memory in Default Configuration 7-106
Figure 7.3 Basic Timing Diagram For External Memories 7-110
Figure 7.4 Example of Normal Address Mode 7-112
Figure 7.5 Example of Indirect Address Mode 7-113
Figure 7.6 Example of Wrapping Address Mode 7-114
Figure 8.1 Block Diagram and Interface of EDI 8-115
Figure 8.2 Block Diagram for EDI_CSN[5:0] 8-119
Figure 8.3 Block Diagram for EDI_CSN[6] 8-120
Figure 8.4 Block Diagram for EDI_RDY[1:0] 8-121
Figure 9.1 IDE Controller Block Diagram 9-123
Figure 9.2 PIO Interface Timing Diagrams 9-126
Figure 9.3 HSTROBE and Max3 Counter Timing Diagrams 9-128
Figure 10.1 SATA Host Adaptor Block Diagram 10-137
Figure 10.2 SATA Host Link Block Diagram 10-137
Figure 10.3 SATA Host Bus Interface Block Diagram 10-138
Figure 10.4 Transport Layer Block Diagram 10-148
Figure 10.5 Link Layer Generated Tx OOB Signaling 10-155
Figure 10.6 Link Layer Rx OOB Detection 10-156
Figure 10.7 Power Mode Example: Rx & Tx OOB In Link 10-156
Figure 11.1 AUDIO DMA Top Block Diagram 11-183
Figure 11.2 DAI Block Diagram 11-184
Figure 11.3 CDIF Block Diagram 11-185
Figure 11.4 SPDIF Tx Block Diagram 11-185
Figure 11.5 SPDIF Rx Block Diagram 11-186
Figure 11.6 Relation between Hop and Burst Transfers 11-192
Figure 11.7 DAI Bus Timing Diagram 11-203
Figure 11.8 DAI TDM Mode Timing Diagram 11-206
Figure 11.9 CDIF Bus Timing Diagram 11-208
Figure 11.10 Data Format between Memory and Buffer 11-223
Figure 12.1 DAI Block Diagram 12-225
Figure 12.2 DAI Bus Timing Diagram 12-231
Figure 12.3 CDIF Block Diagram 12-234
Figure 12.4 CDIF Bus Timing Diagram 12-235
Figure 14.1 USB Host Controller Block Diagram 14-245
Figure 14.2 USB 1.1 Transceiver Block Diagram 14-246
Figure 15.1 USB Controller Block Diagram 15-252
Figure 15.2 USB OTG CSR Memory Map 15-253
Figure 15.3 USB OTG Controller Interrupt Hierarachy 15-271
Figure 15.4 VBUS Control and Interrupts 15-327
Figure 15.5 Transmit Transaction-Level Operation in Slave Mode 15-333
Figure 15.6 Receive Transaction-Level Operation in Slave Mode 15-334
Figure 15.7 Transmit FIFO Write Task in Slave Mode 15-338
Figure 15.8 Receive FIFO Read Task in Slave Mode 15-339
Figure 15.9 Normal Bulk/Control OUT/SETUP and Bulk/Control IN Transactions in Slave Mode 15-340
Figure 15.10 Normal Bulk/Control OUT/SETUP and Bulk/Control IN Transactions in DMA Mode 15-343
Figure 15.11 Normal Interrupt OUT/IN Transactions in Slave Mode 15-347
Figure 15.12 Normal Interrupt OUT/IN Transactions in DMA Mode 15-350
Figure 15.13 Normal Isochronous OUT/IN Transactions in Slave Mode 15-354
Figure 15.14 Normal Bulk/Control OUT/SETUP and Bulk/Control IN Split Transactions in Slave Mode 15-355
Figure 15.15 Normal Bulk/Control OUT/SETUP and Bulk/Control IN Split Transactions in DMA Mode 15-358
Figure 15.16 Normal Interrupt OUT/IN Split Transactions in Slave Mode 15-361
Figure 15.17 Normal Interrupt OUT/IN Split Transactions in DMA Mode 15-364
Figure 15.18 Normal Isochronous OUT/IN Split Transactions in Slave Mode 15-367
Figure 15.19 Normal Isochronous OUT/IN Split Transactions in DMA Mode 15-370
Figure 15.20 Receive FIFO Packet Read in Slave Mode 15-377
Figure 15.21 Processing a SETUP Packet 15-379
Figure 15.22 Slave Mode Bulk OUT Transaction 15-387
Figure 15.23 Slave Mode Bulk IN Transaction 15-396
Figure 15.24 Slave Mode Bulk IN Transfer (Pipelined Transaction) 15-397
Figure 15.25 Slave Mode Bulk IN Two-Endpoint Transfer 15-399
Figure 15.26 Bulk IN Stall 15-400
Figure 15.27 Bulk IN DMA mode with Thresholding 15-401
Figure 15.28 Isochronous IN DMA Mode with Thresholding 15-402
Figure 15.29 Two-Stage Control Transfer 15-405
Figure 15.30 USBTrdTim Max Timing Case 15-408
Figure 15.31 A-Device SRP 15-409

Figure 15.32 B-Device SRP 15-410
Figure 15.33 A-Device HNP 15-411
Figure 15.34 B-Device HNP 15-412
Figure 15.35Host Mode Suspend and Resume With Clock Gating 15-413
Figure 15.36Host Mode Suspend and Remote Wakeup With Clock Gating 15-414
Figure 15.37 Core Interrupt Handler 15-424
Figure 16.1 Example of EHAM usage 16-428
Figure 16.2 Interrupt request using EHCFG.WRIRQ and CSIRQ 16-431
Figure 16.3 Example of writing / reading operation (80 interface, 16bits) 16-434
Figure 16.4 Example of writing / reading operation (68 interface, 16bits) 16-434
Figure 16.5 Access to the On-chip System Bus 16-435
Figure 16.6 Lock tranfser vs Normal transfer (EHRWCS.BSIZE=3) 16-436
Figure 16.7 Example of Writing to the On-chip System Bus (80 interface, 8bits) 16-436
Figure 16.8 Example of Reading from the on-chip System Bus (68 interface, 8bits) 16-437
Figure 16.9 Pseudo code for getting and releasing a semaphore 16-437
Figure 17.1 GPSB Interface Block Diagram 17-439
Figure 17.2 Overall GPSB Block Diagram 17-439
Figure 17.3 Packet Structure 17-448
Figure 17.4 TX/RX Addressing Modes 17-450
Figure 17.5 SPI Timing 0 17-454
Figure 17.6 SPI Timing 1 17-454
Figure 17.7 SPI Timing 2 17-454
Figure 17.8 SPI Timing 3 17-454
Figure 17.9 SSP Timing 17-454
Figure 17.10 Bitstream of MPEG2-TS 17-455
Figure 17.11 MPEG2-TS timing information 17-455
Figure 18.1 TSIF Block Diagram 18-457
Figure 18.2 Timing of TS Data Input (Parallel Mode) 18-457
Figure 18.3 Timing of TS Data Input (Serial Mode) 18-457
Figure 19.1 Remote Control Interface Block Diagram 19-463
Figure 19.2 IR data input example 19-463
Figure 19.3 The overall flow of IR data capture 19-464
Figure 19.4 Data write format in FIFO 19-464
Figure 19.5 The timing diagram about internal counter clock and dur_count value 19-464
Figure 19.6 The minimum or maximum pulse width of IR signal 19-464
Figure 20.1 I2C Block Diagram 20-469
Figure 20.2 I2C Slave Block Diagram 20-469
Figure 21.1 UART Block Diagram 21-481
Figure 21.2 nRTS operation - Case: CPU/DMA is slow 21-483
Figure 21.3 nRTS operation - Case: CPU/DMA is fast 21-483
Figure 21.4 nCTS operation 21-484
Figure 21.5 Operation with GDMA 21-484
Figure 21.6 DMA operation – RX interrupt 21-485
Figure 21.7 DMA operation – TX interrupt 21-486
Figure 21.8 RX operation with DMA and AFC 21-487
Figure 21.9 TX operation with DMA and AFC 21-488
Figure 23.1 GDMA Controller Block Diagram 22-505
Figure 23.2 Relation between Hop and Burst Transfers (If burst size is 4.) 22-513
Figure 23.3 The Example Of Various Types of Transfer. 22-513
Figure 23.4 Data transfer when Channel0 and Channel1 are enabled 22-518
Figure 24.1 RTC Block Diagram 23-519
Figure 24.2 Boot-Up Sequence 23-521
Figure 24.3 The RTC Time Setting Sequence 23-521
Figure 24.4 RTC Alarm Time Setting Sequence 23-522
Figure 24.5 PEND Clear Sequence 23-522
Figure 24.6 RTC Operation Process Flow Chart. 23-523
Figure 24.7 Example of Crystal Oscillator Circuit Connection 23-524
Figure 25.1 ADC Controller Block Diagram 24-533
Figure 25.2 Main waveform 24-534
Figure 26.1 ECC Block Diagram 25-543
Figure 26.2 Example of MLC ECC4 error correction 25-553
Figure 26.3 SLC ECC Encoding Sequence 25-558
Figure 26.4 SLC ECC Decoding Sequence 25-559
Figure 26.5 MLC ECC4/8/12/14/16 Encoding Sequence 25-560
Figure 27.1 MPEFEC frame 26-563
Figure 27.2 MPEFEC buffer write 26-563
Figure 27.3 MPEFEC Erasure 26-564
Figure 27.4 MPEFEC Top Block Diagram 26-565

Tables

Table 4.1 Bus State 4-8
Table 4.2 SMSHC_I Register Map (Base Address = 0xF0590000) 4-10
Table 4.3 PORTCFG Register Map (Base Address = 0xF05F1000) 4-10
Table 4.4 PORTDLY Register Map (Base Address = 0xF05F1004) 4-10
Table 4.5 How to Specify a Data Buffer for DMA Transfers 4-13
Table 4.6 The Values Set to TimeCount and TIMER 4-17
Table 4.7 Conversion of a [PageBuffer] Bank 4-19
Table 4.8 Transfer Protocol Code 4-20
Table 4.9 Interface Mode Configuration 4-25
Table 4.10 DMA Slice Size Configuration 4-27
Table 4.11 Microcode for Interface Mode Switching 4-36
Table 4.12 Instruction ID 4-45
Table 4.13 Access Flag Set to [General Register4,5] 4-47
Table 4.14 How to Specify Absolute Address 4-48
Table 4.15 How to Specify Relative Address 4-48
Table 4.16 Setting of [General Register] 4-49
Table 4.17 Setting of I/O Address 4-49
Table 4.18 Setting of Access Width 4-49
Table 4.19 [General Register] Accessed when Access Width is 1 4-49
Table 4.20 DMA Slice Size of [General Data FIFO] 4-50
Table 4.21 DMA Slice Size of [PageBuffer] 4-50
Table 5.1 Register Map (Base Address = 0xF05A0000) 5-53
Table 5.2 Base Address of Each Slot 5-54
Table 5.3 Channel Control Register Map (Base Address = 0xF05A0800) 5-54
Table 5.4 Command Format 5-59
Table 5.5 Response Register 5-61
Table 5.6 Relation between Transfer Complete and Data Timeout Error 5-71
Table 5.7 Relation between Command Complete and Command Timeout Error 5-71
Table 5.8 Relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error 5-76
Table 5.9 Command Format 5-84
Table 6.1 NAND Flash Controller Register Map (Base Address=0xF05B0000) 6-90
Table 6.2 Page Size of NAND Flash 6-92
Table 7.1 The Address Mapping for Each Address Mode 7-106
Table 7.2 The Base Address for Each Chip Select 7-106
Table 7.3 Memory Controller Register Map (Base Address = 0xF05F0000) 7-107
Table 8.1 The Configuration of GPIO Function for EDI 8-116
Table 8.2 EDI Register Map (BASE ADDRESS : 0xF05F6000) 8-117
Table 8.3 EDI_CSNCFG0[CFGCSn] Configuration 8-118
Table 8.4 EDI_CSNCFG1[CFGCSn] Configuration 8-120
Table 8.5 CFGRDYn Configuration 8-121
Table 9.1 Speed of Data Transfer (Byte per Second) 9-123
Table 9.2 IDE Registers (Base = 0xF0520000) 9-123
Table 10.1 SATA Host Register Map (Base Address = 0xF0560000) 10-157
Table 10.2 I2C Register Setting Examples 10-178
Table 11.1 AUDIO DMA Register Map (Base Address = 0xF0533000) 11-187
Table 11.2 DAI Register Map (Base Address = 0xF0534000) 11-188
Table 11.3 CDIF Register Map (Base Address = 0xF0534080) 11-188
Table 11.4 SPDIF Tx Register Map (Base Address = 0xF0535000) 11-188
Table 11.5 SPDIF Rx Register Map (Base Address = 0xF0535800) 11-188
Table 12.1 DAI Register Map (Base Address = 0xF0537000) 12-227
Table 12.2 CDIF Register Map (Base Address = 0xF0537000) 12-232
Table 13.1 SPDIF Register Map (Base Address = 0xF0538000) 13-237
Table 14.1 USB Host Register Map (Base Address = 0xF0500000) 14-247
Table 14.2 USB 1.1 Host Configuration Register(Base Address = 0xF05F5000) 14-247
Table 15.1 USB Register Map (Base Address = 0xF0550000) 15-253
Table 15.2 USB OTG Register (Base Address = 0xF05F5000) 15-255
Table 15.3 Minimum Duration for Soft Disconnect 15-302
Table 15.4 Interrupt Service Routine for Ping Protocol in Slave Mode 15-337
Table 15.5 Interrupt Service Routine for Bulk/Control OUT/SETUP and Bulk/Control IN Transactions in Slave Mode 15-341
Table 15.6 Interrupt Service Routines for Bulk/Control OUT/SETUP and Bulk/Control IN Transactions in DMA Mode 15-344
Table 15.7 Interrupt Service Routine for Interrupt OUT/IN Transactions in Slave Mode 15-347
Table 15.8 Interrupt Service Routine for Interrupt OUT/IN Transactions in DMA Mode 15-350
Table 15.9 Interrupt Service Routine for Isochronous OUT/IN Transactions in Slave Mode 15-352

Table 15.10 Interrupt Service Routine for Bulk/Control OUT/SETUP and Bulk/Control IN Split Transactions in Slave Mode 15-355
Table 15.11 Interrupt Service Routine for Bulk/Control OUT/SETUP and Bulk/Control IN Split Transactions in DMA Mode 15-358
Table 15.12 Interrupt Service Routine for Interrupt OUT/IN Split Transactions in DMA Mode 15-364
Table 15.13 Interrupt Service Routine for Isochronous OUT/IN Split Transactions in Slave Mode 15-367
Table 16.1 EHI External Interface Pin 16-425
Table 16.2 EHI register map(EHI_Base = 0xF0570000, 0xF0580000) 16-425
Table 16.3 Access to the EHI 16-433
Table 17.1 GPSB Register Map (Base Address = 0xF0536000) 17-440
Table 18.1 GPIO Port Map 18-458
Table 18.2 TSIF Register Map (Base Address = 0xF053B000) 18-459
Table 19.1 Remocon Register Map (Base Address = 0xF05F3000) 19-465
Table 20.1 I2C Register Map (Base Address = 0xF0530000) 20-470
Table 21.1 UART Register Map 21-489
Table 21.2 UART Port Mux Register 21-489
Table 23.1 General DMA Controller Register Map(DMA Base Address = 0xF0540n00, n=0,1,2,3) 22-507
Table 23.2 BASE Address(DMA_BASE) of All GDMA Channles 22-508
Table 24.1 Recommended Oscillator Circuit Constants 23-523
Table 24.2 RTC Register Map (Base Address = 0xF05F2000) 23-525
Table 24.3 RTC Control Register (RTCCON) 23-525
Table 24.4 RTC Interrupt Control Register (INTCON) 23-527
Table 24.5 RTC Alarm Control Register (RTCALM) 23-527
Table 24.6 Alarm Second Data Register (ALMSEC) 23-528
Table 24.7 Alarm Minute Data Register (ALMMIN) 23-528
Table 24.8 Alarm Hour Data Register (ALMHOUR) 23-528
Table 24.9 Alarm Date Data Register (ALMDATE) 23-528
Table 24.10 Alarm Day of Week Data Register (ALMDAY) 23-528
Table 24.11 Alarm Month Data Register (ALMMON) 23-529
Table 24.12 Alarm Year Data Register (ALMYEAR) 23-529
Table 24.13 BCD Second Data Register (BCDSEC) 23-529
Table 24.14 BCD Minute Data Register (BCDMIN) 23-529
Table 24.15 BCD Hour Data Register (BCDHOUR) 23-529
Table 24.16 BCD Date Data Register (BCDDATE) 23-530
Table 24.17 BCD Day of Week Data Register (BCDDAY) 23-530
Table 24.18 BCD Month Data Register (BCDMON) 23-530
Table 24.19 BCD Year Data Register (BCDYEAR) 23-530
Table 24.20 RTC Interrupt Mode Register (RTCIM) 23-530
Table 24.21 RTC Interrupt Pending Register (RTCPEND) 23-531
Table 24.22 RTC Interrupt Status Register (RTCSTR) 23-531
Table 25.1 TSADC Controller Register Map (Base Address = 0xF05F4000) 24-535
Table 25.2 I/O Chart 24-541
Table 25.3 Analog Input Selection Table for 4-wire touch screen panels 24-541
Table 25.4 Analog Input Selection Table for Pressure Measurement 24-542
Table 26.1 ECC Register Map (ECC BASE Address=0xF0539000) 25-544
Table 27.1 MPEFEC Register Map (Base Address = 0xF0510000) 26-565
Table 28.1 IOBUS Configuration Register Map (Base Address = 0xF05F5000) 27-569

1 Introduction

TCC8900 IOBUS provides connections between the internal bus master and external device controller. Also on chip peripherals are located in it. The sort of interfaces supported is as follows : storage interface, audio interface, high speed interface, miscellaneous interface.

[Features]

- STORAGE INTERFACES
  - Memory Stick controller
    - MS/MS PRO/MS PRO-HG
  - Memory Card Interfaces for SD/MMC
    - 1/4 bit SDIO, SDMA, ADMA mode
    - 8bit MMC, CE-ATA mode
  - NAND flash controller
  - SRAM interface
  - External Device Interface
  - UDMA33/66, PIO, IDE Interface for HDD
  - Serial ATA Host Adaptor
    - Gen1(1.5G) or Gen2(3.0G) speed
- AUDIO INTERFACES
  - AUDIO I/O with DMA master
    - AHB master operation
    - I2S Master & Slave Interface
    - S/PDIF RX/TX Interface
  - AUDIO I/O with internal DMA support
    - AHB slave operation
    - Support interfaces to combine with internal DMA controller
    - I2S Master & Slave Interface
    - S/PDIF TX Interface
- HIGH SPEED INTERFACES
  - USB1.1 LS/FS host controller
  - High speed USB2.0 FS/HS OTG controller
  - 2-Channel host port interface via 80/68 compatible – 8 / 16 bits
- MISCELLANEOUS INTERFACE
  - 6-Channel GPSB with TS interface for supporting various serial interfaces
  - 2-Channel dedicated TS parallel interface
  - Remote control interface
  - Configurable 3-Channel I2C
    - 2-Channel Masters
    - 1 Channel I2C Slave
  - 6-Channel UART for serial host interface
    - 3-channel without hardware flow control
    - 3-channel with hardware flow control
- ON-CHIP PERIPHERALS
  - 12 Channel DMA for transferring bulk data
    - Including Hardware DMA Request for Each Channel
  - RTC : Power-Down Mode & Auto-wakeup
  - Touch Screen ADC controller
    - 10/12 bit ADC
  - ECC calculator
  - MPEFEC

2 Bus Architecture

Figure 2.1 shows the I/O bus overall architecture.

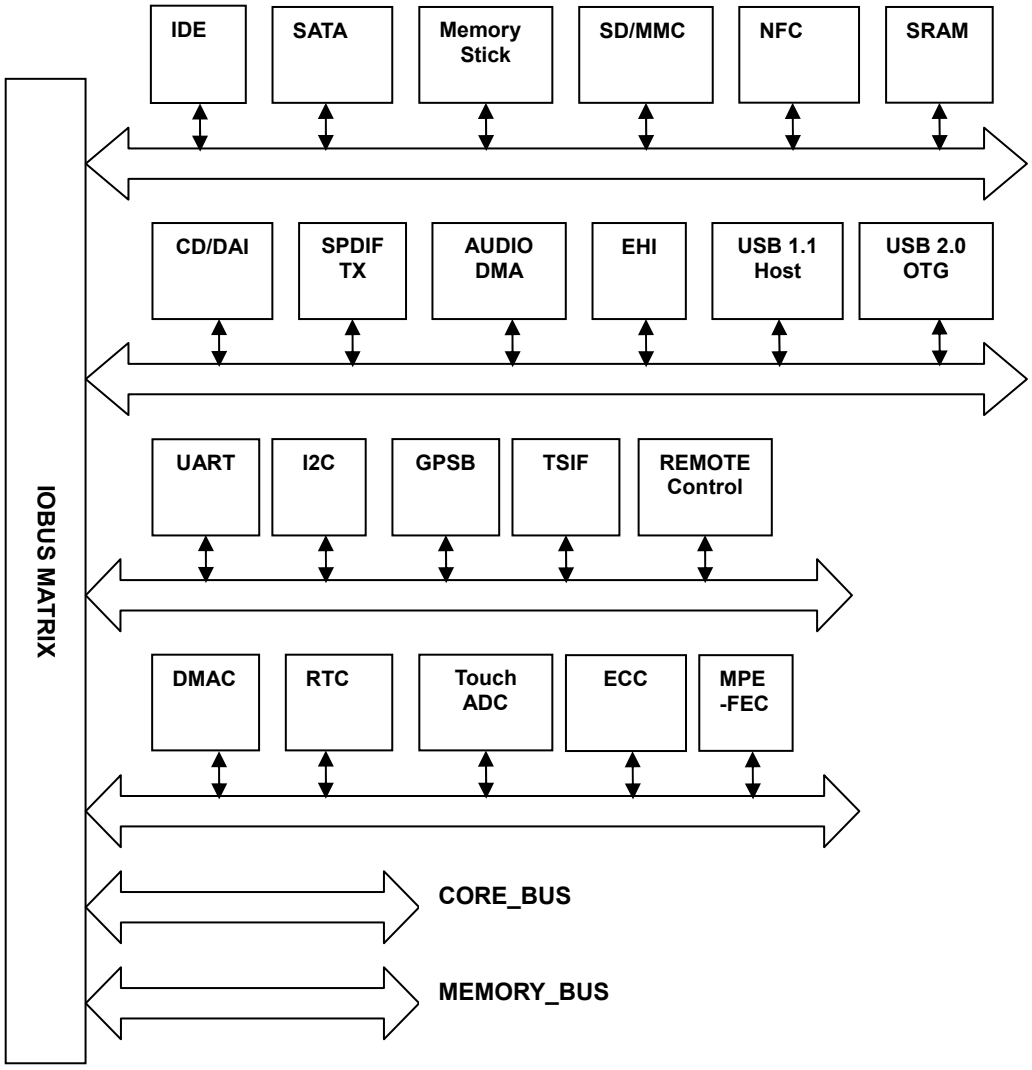


Figure 2.1 The I/O Hardware Bus Architecture

3 Address and Register Map

The TCC8900 has various peripherals for specific interface controllers or on-chip hardware components. These peripherals can be configured appropriately by its own registers that can be accessed through specially allocated address. These address maps are represented in the following table.

Refer to corresponding sections for detail information of each peripheral.

Base Address		Peripherals
0xF0500000		USB1.1 Host Controller
0xF0510000		MPE_FEC Controller
0xF0520000		HDD(IDE) Controller
0xF0530000	0xF0530000	I2C Master Controller 0
	0xF0530040	I2C Master Controller 1
	0xF0530080	I2C Slave Interface
	0xF05300C0	I2C Interrupt Register
0xF0531000		Reserved
0xF0532000	0xF0532000	UART Channel 0 with DMA support
	0xF0532100	UART Channel 1 with DMA support
	0xF0532200	UART Channel 2 with DMA support
	0xF0532300	UART Channel 3 with DMA support
	0xF0532400	UART Channel 4 without DMA support
	0xF0532500	UART Channel 5 without DMA support
	0xF0532600	UART Port Configuration Registers
0xF0533000		Audio DMA Controller
0xF0534000		Audio DAI Controller
0xF0535000		Audio SPDIF TX/RX Controller
0xF0536000	0xF0536000	GPSB channel 0 without DMA support
	0xF0536100	GPSB channel 1 without DMA support
	0xF0536200	GPSB channel 2 without DMA support
	0xF0536300	GPSB channel 3 with DMA support
	0xF0536400	GPSB channel 4 with DMA support
	0xF0536500	GPSB channel 5 with DMA support
	0xF0536800	GPSB Port Configuration Registers
	0xF0536F00	GPSB PID Table
0xF0537000		DAI/CDIF Controller
0xF0538000		SPDIF-TX Controller
0xF0539000		ECC Calculator
-		Reserved
0xF053B000	0xF053B800	TS Interface 0
	0xF053B900	TS Interface 1
	0xF053BA00	TS Interface Port Configuration Registers
0xF0540000	0xF0540000	Central DMA Controller (Channel 0,1,2)
	0xF0540100	Central DMA Controller (Channel 3,4,5)
	0xF0540200	Central DMA Controller (Channel 6,7,8)
	0xF0540300	Central DMA Controller (Channel 9,10,11)
0xF0550000		USB2.0 OTG Controller
0xF0560000		S-ATA Controller
0xF0570000		External Host Interface 0
0xF0580000		External Host Interface 1
0xF0590000		Memory Stick Controller
0xF05A0000	0xF05A0000	SD/MMC Controller 0
	0xF05A0200	SD/MMC Controller 1
	0xF05A0800	SD/MMC Port Configuration Registers
0xF05B0000		NAND Flash Controller
0xF05C0000		SRAM-like Master Interface
0xF05F0000		Static Memory Controller Registers
0xF05F1000		Memory Stick Port Configuration Registers
0xF05F2000		RTC(Real-Time Clock)
0xF05F3000		Remote Control Interface
0xF05F4000		Touch Screen ADC Controller
0xF05F5000		I/O Bus Configuration Registers
0xF05F6000		External Device Interface Control & Port Configuration Registers

4 Memory Stick Host Controller

4.1 Overview

This is a host controller with 32-bit CPU interface that supports Memory Stick Ver. 1.x and Memory Stick PRO-HG Duo. This conforms to “Memory Stick Standard Format Specifications ver. 1.4x” and “Memory Stick Standard Memory Stick PRO-HG Duo Format Specifications ver. 1.0x”

The memory stick host controller has the following features.

- Data transmit/receive FIFO (64 bits x 4)
- Hardware CRC Generator
- Memory Stick Serial Clock (Serial : 20MHz-MAX, Parallel : 60MHz-MAX)
- Burst Transfer via On-Chip DMA Controller

Listed below are major memory stick device and modules supported by this host controller.

- Non Copyright Protected Memory Stick Media
- MagicGate Memory Stick Media
- MagicGate Memory Stick Media with Parallel Transfer Support
- Memory Stick PRO Media
- Memory Stick Micro Media
- Memory Stick PRO-HG Duo Media
- Memory Stick I/O Expansion Module
- Memory Stick PRO I/O Expansion Module+

The host controller is called “smshc_i” standing for **S**ony **M**emory **S**tick **H**ost **C**ontroller with **I-CON**. The smshc_i is a host controller supporting memory stick, equipped with a 32-bit AHB interface.
The smshc_i is composed of two blocks, smshc and I-CON. The former performs communications with memory stick while the latter is a sequencer block which automatically controls the smshc. Each block is mapped to an address which can be accessed by Host CPU via AHB interface.

The I-CON makes it possible to automatically manage the command control sequence for memory stick on behalf of a host CPU. This is how the workload imposed on the host CPU can be reduced. The Figure 4.1 shows this architecture simply.

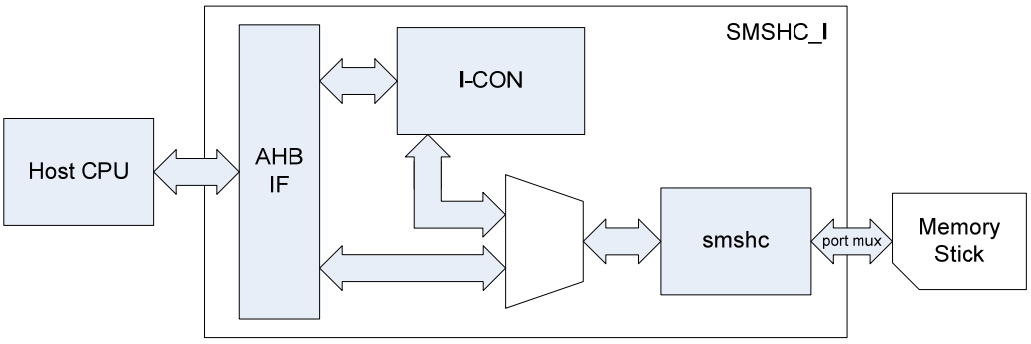


Figure 4.1 The Components in SMSHC_I

smshc Block

- The smshc can be controlled by either a host CPU or the I-CON. However, when it is controlled by one of the two, the other one can not control it.
- The smshc issues TPCs for communicating with memory stick.
- The smshc can select one of the following three interfaces for the communication with memory stick; serial interface, 4-bit parallel interface, or 8-bit parallel interface.
- The smshc can transfer data up to 2048 bytes. That is the maximum size transferable by a single TPC.
- The smshc supports interrupts through 8-channel DATA lines from memory stick.
- FIFO, or [MSHC-Data Register] (64 bits × 4), is built in the smshc.
- Only when the I-CON is not used, the smshc can output a DMA transfer request for [MSHC-Data Register]. The smshc enables a host CPU to set whether or not a DMA transfer request can be output per TPC.

I-CON Block

- The I-CON automatically controls the smshc and can perform command control sequences to memory stick. A host CPU sets command control sequences for memory stick to [Instruction Queue], a buffer for instructions, as microcode.
- The I-CON runs instructions specified by microcode one after another in order.
- A FIFO for sending/receiving general data [General Data FIFO] is built in the I-CON.
- I-CON is equipped with a memory interface for [PageBuffer], a buffer which sends/receives page data.
- I-CON is equipped with a memory interface for [Instruction Queue], an instruction buffer.
- 2 channels are built in the I-CON for data transfer requests.
  (1) Data transfer requests for [PageBuffer]
  (2) Data transfer requests for [General Data FIFO]
- How the I-CON notifies a data transfer request can be changed with a host CPU. Refer to [System Register].

Memory Stick Signals and External Port Mux

Memory Stick Interface signals are as follows:.

- SCLK : Clock output signal to Memory Stick device
- BS : Bus State signal of Memory Stick device
- DATA[7:0] : Data in/output signal for Memory Stick device

The bus state(BS) is classified into four states depending on the attributes and the transfer directions of data on DATA[7:0]. They include BS0 where no packet communication is performed, and BS1 to BS3 where packet communication is being performed. Each data output timing is controlled according to the bus states. The bus states from BS1 to BS3 are defined as a single packet, and each communication should be completed with a packet.

Table 4.1 Bus State

State	BS	Description	
BS0	Low	INT Transfer State : A state in which no packet communication is performed and an INT (interrupt) signal is sent from Memory Stick to a host controller.	
BS1	High	TPC Transfer State : A state in which a TPC is transferred from the host controller to the Memory Stick in order to start packet communication.	
BS2	Low	Read packet	Write packet
		Handshake State : The host controller waits for a RDY signal from the Memory Stick.	Data Transfer State : The host controller transfers data to the Memory Stick.
BS3	High	Read packet :	Write packet :
		Data Transfer State : Memory Stick transfers data to the	Handshake State : The host controller waits for a RDY

		host controller.	signal from Memory Stick.

There are 8 channels for Memory Stick interface in TCC8900. Each Memory Stick interface signal is mapped to one port of them.

Block Diagram

The Figure 4.2 shows the block diagram of memory stick host controller.

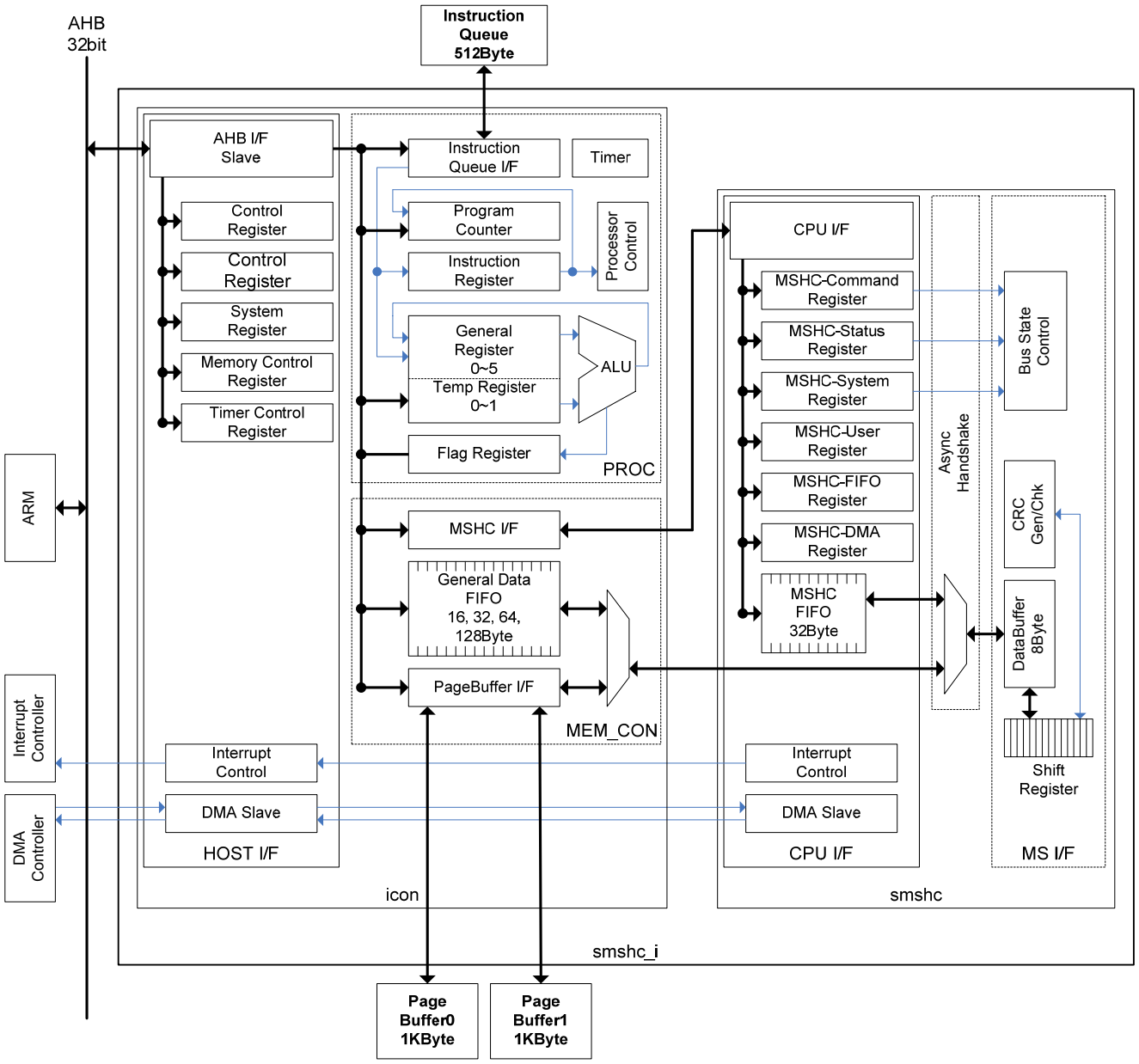


Figure 4.2 Memory Stick Host Controller Block Diagram

4.2 Register Descriptions

Table 4.2 SMSHC_I Register Map (Base Address = 0xF0590000)

Address	Name				Type	Default
	[31:24]	[23:16]	[15:8]	[7:0]		
0x00	Control Register		Program Counter Register		R/W	0x0070_1000
0x04	System Register		-		R/W	0x0800_XXXX
0x08	Flag Register		-		R	0x4000_XXXX
0x0C	Memory Control Register		-		R/W	0x0001_7000
0x10	General Register0		General Register1		R/W	0x8000_9000
0x14	General Register2		General Register3		R/W	0xA000_B000
0x18	General Register4		General Register5		R/W	0xC000_D000
0x1C	Timer Register		-		R	0xE000XXXX
0x20	Instruction Queue				R/W	0xXXXX_XXXX
	LSB(Least Significant Byte) First Mode					
	[15:0]		[31:16]			
	MSB(Most Significant Byte) First Mode					
	[31:16]		[15:0]			
0x24	General Data FIFO				R/W	0x0000_0000
	LSB(Least Significant Byte) First Mode					
	[7:0]	[15:8]	[23:16]	[31:24]		
	MSB(Most Significant Byte) First Mode					
	[31:24]	[23:16]	[15:8]	[7:0]		
0x28	PageBuffer				R/W	0xXXXX_XXXX
	LSB(Least Significant Byte) First Mode					
	[7:0]	[15:8]	[23:16]	[31:24]		
	MSB(Most Significant Byte) First Mode					
	[31:24]	[23:16]	[15:8]	[7:0]		
0x2C	Version Register		-		R	0xXXXX_XXXX
0x30	MSHC-Command Register		-		R/W	0x0000_XXXX
0x34	MSHC-Data Register				R/W	0x0000_0000
	LSB(Least Significant Byte) First Mode					
	[7:0]	[15:8]	[23:16]	[31:24]		
	MSB(Most Significant Byte) First Mode					
	[31:24]	[23:16]	[15:8]	[7:0]		
0x38	MSHC-Status Register		-		R	0x1000_XXXX
0x3C	MSHC-System Register		-		R/W	0x20A5_XXXX
0x40	MSHC-User Custom Register		-		R	0x0220_XXXX
0x44	MSHC-FIFO Control Register		-		R/W	0x0001_XXXX
0x4C	MSHC-DMA Control Register		-		R/W	0x0000_XXXX

Table 4.3 PORTCFG Register Map (Base Address = 0xF05F1000)

Name	Address	Type	Reset	Description
PORTCFG	0x00	R/W	0x00000000	Port Configuration Register

Table 4.4 PORTDLY Register Map (Base Address = 0xF05F1004)

Name	Address	Type	Reset	Description
PORTDLY	0x00	R/W	0x00000000	Port Output Delay Register

Control / Program Counter Register 0xF0590000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								GDIR	GRPN		PDIR	INTC	INT	0	START
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0			1	0				PRGC							

GDIR[23]	General Data FIFO Direction
0	CPU ← [General Data FIFO] ← Reading data from memory stick
1	CPU → [General Data FIFO] → Writing data to memory stick

GRPN[22:21]	General Register for PageBuffer Data Number
00	[General Register0]
01	[General Register1]
10	[General Register2]
11	No writing to [PageBuffer]

PDIR[20]	PageBuffer Direction
0	CPU ← [PageBuffer] ← Reading data from memory stick
1	CPU → [PageBuffer] → Writing data to memory stick

INTC[19]	INT Clear
0	Not clear
1	Interrupt Clear, Automatically back to 0 after clearing the interrupt

INT[18]	Interrupt (Read Only)
0	An interrupt has not occurred in the I-CON block
1	An interrupt has occurred in the I-CON block

START[16]	Start
0	Do not start self-run
1	The I-CON can be a self-run state The address for starting to proceed an instruction is specified by the PRGC bits in [Program Counter Register]

PRGC[7:0]	Program Counter
	These bits are an address pointer for writing/reading/executing each instruction to/from/in [Instruction Queue] To write an instruction, the address to start writing from shall be first set to PRGC After the instruction has been written, these bits can be automatically incremented

System Register **0xF0590004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAIM1	DMAIM0	DMAE1	DMAE0	DMASL[1:0]		GDSZ[1:0]		DMACH	PDSZ[2:0]			0	INTE	0	SRST
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-															

DMAIM1	[31]	DMA1 Interrupt Mode
1	This bit enables the occurrence of an interrupt for a data transfer request (Channel1)	
0	Disable	

DMAIM0	[30]	DMA0 Interrupt Mode
1	This bit enables the occurrence of an interrupt for a data transfer request (Channel0)	
0	Disable	

DMAE1	[29]	DMA1 Enable
1	This bit enables outputting the DMA transfer request signal for a data transfer request (Channel1)	
0	Disable	

DMAE0	[28]	DMA0 Enable
1	This bit enables outputting the DMA transfer request signal for a data transfer request (Channel0)	
0	Disable	

DMASL	[27:26]	DMA Select
	If the number of data buffers to be used is 1 (DMACH =0), these bits select the data buffer where a data transfer request occurs. 00b : [General Data FIFO] 01b : [PageBuffer] 1Xb : [MSHC-Data Register]	

DMACH	[23]	DMA Channel Number
	This bit specifies the number of data buffers to be used 0 : 1 kind 1 : 2 kinds	

The Table 4.5 below shows how to specify a data buffer for DMA transfer. DMA requests for [MSHC-Data Register] are output by setting DMASL=2'b1x only.

Table 4.5 How to Specify a Data Buffer for DMA Transfers

DMACH	DMASL		Channel 0	Channel 1
	[1]	[0]		
0	0	0	[General Data FIFO]	Not Used
0	0	1	[PageBuffer]	Not Used
0	1	X	[MSHC-Data Register]	Not Used
1	X	X	[General Data FIFO]	[PageBuffer]

GDSZ	[25:24]	General Data FIFO DMA Slice Size			
	This bits specify the transfer data size of a single data transfer for [PageBuffer]				
GDSZ	Slice Size	Capacity of General Data FIFO			
		16Byte	32Byte	64Byte	128Byte
00b	4Byte	o	O	o	o
01b	16Byte	x	O	o	o
10b	32Byte	x	X	o	o
11b	64Byte	x	x	x	o

PDSZ	[22:20]	PageBuffer DMA Slice Size
	This bits specify the transfer data size of a single data transfer for [PageBuffer] 000b : 4Byte 001b : 16Byte 010b : 32Byte 011b : 64Byte 100b : 128Byte 101b : 256Byte 110b : Reserved 111b : Reserved	

INTE	[18]	INT Enable
1	This bit enables outputting an interrupt	
0	Disable	

SRST	[16]	Soft Reset
	By setting 1 to this bit, the I-CON block performs a synchronous reset After the synchronous reset, the bit will be returned to 0	

Flag Register **0xF0590008**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MIEF	1	0				DMAF1	DMAF0	FLG	HLTF	ITOF	STPF	EXTS			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-															

All bits of this register are read-only.

MIEF	[31]	MSINT Error Flag
	This bit becomes 1 if an error is detected while checking an INT status with the Instruction 2 to send/receive through [PageBuffer] in Section 8.3.3 TPC Instructions. It is judged that an error has occurred in case of any other interrupt cause than BREQ. It will be cleared to 0 when the START bit of [Control Register] changes from 0 to 1.	

DMAF1	[25]	DMA1 Request Flag
	This bit remains 1 while there is a request through “Data Transfer Request Channel 1.” When DMAE1 of [System Register] is 0, it will be cleared to 0 as soon as data of the slice size has been read/written. However, if there happens a fraction of data smaller than the slice size, this bit will not be cleared to 0 until the fraction is completely read/written. While DMAE1 bit of [System Register] is high, DMAF1 will be cleared, if acknowledge input to DMA channel 1 is set.	

DMAF0	[24]	DMA0 Request Flag
	This bit remains 1 while there is a request through “Data Transfer Request Channel 0.” When DMAE0 of [System Register] is 0, it will be cleared to 0 as soon as data of the slice size has been read/written. However, if there happens a fraction of data smaller than the slice size, this bit will not be cleared to 0 until the fraction is completely read/written. While DMAE0 bit of [System Register] is high, DMAF0 will be cleared, if acknowledge input to DMA channel 0 is set.	

FLG	[23]	Flag
	This flag is used during a self-run. When a Compare instruction is executed, this bit answers 1 for true and 0 for false. On the other hand, when a TPC instruction is executed this bit becomes 1 when a communication error with memory stick occurs. It will be cleared to 0 when the START bit of [Control Register] changes from 0 to 1.	

HLTF	[22]	Halt Flag
	This bit becomes 1 when a Halt instruction is completely processed. It will be cleared to 0 when the START bit of [Control Register] changes from 0 to 1.	

ITOF	[21]	INT Timeout Flag
	This bit becomes 1 when a timeout occurs during a Wait instruction or a TPC instruction. It will be cleared to 0 when the START bit of [Control Register] changes from 0 to 1.	

STPF	[20]	Stop Flag
	While an instruction is being processed (the START bit of [Control Register] is 1), if the process is attempted to stop (0 is set to the START bit), this STPF bit becomes 1 when the stop process is completed. This bit is cleared to 0 when the START bit of [Control Register] changes fro m 0 to 1.	

EXTS	[19:16]	Exit Status
	When a Halt instruction is completed, the exit status will be reflected on these bits. They will be cleared to 0 when the START bit of [Control Register] changes from 0 to 1.	

Memory Control Register **0xF059000C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GFVD								0		PBBC		PBCR	GFCR	PBFUL	PBEMP
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	1	1	1	GFDN											

GFVD	[31:24]	General Data FIFO Valid Data Count (Read-Only)
	These bits specify the number of valid data in [General Data FIFO]. (unit : byte)	

PBBC	[21:20]	PageBuffer Bank Change (R/W)
	These bits change a bank size of [PageBuffer] 00b : 256Byte 01b : Reserved 10b : 1KByte 11b : Reserved	

PBCR	[19]	PageBuffer Clear (R/W)
	[PageBuffer] can be initialized by setting 1 to this bit It automatically returns to 0 after having initialized [PageBuffer].	

GFCR	[18]	General Data FIFO Clear (R/W)
	[General Data FIFO] can be initialized by setting 1 to this bit. It automatically returns to 0 after having initialized [General Data FIFO].	

PBFUL	[17]	PageBuffer Full (Read-Only)
	If [PageBuffer] is a double-buffer structure, this bit becomes 1 when the data size in one of the buffers accessed from a CPU is the same as the PBBC size. When PBFUL=1, writing a value to [PageBuffer] is prohibited.	

PBEMP	[16]	PageBuffer Empty (Read-Only)
	If [PageBuffer] is a double-buffer structure, this bit becomes 1 when the data size in one of the buffers accessed from a CPU is 0. When PBEMP=1, reading a value from [PageBuffer] is prohibited.	

GFDN	[11:0]	General Data FIFO DMA Transfer Data Number (R/W)
	These bits specify the number of accesses when writing data from a host CPU or a DMA controller to [General Data FIFO] during a self-run. (GFDN = Total data / 4Byte) Even when GDIR = 1, if there is no data written from the host CPU or the DMA controller to [General Data FIFO] during a self-run, “0” shall be set to these bits. When GDIR = 0 and a self-run is underway, “0” shall be set to these bits.	

General Register 0/1 **0xF0590010**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
⌙															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
General Register 1															

General Register 2/3 **0xF0590014**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
General Register 2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
General Register 3															

General Register 4/5 **0xF0590018**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
General Register 4															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
General Register 5															

[16:0]	General Register
	Guaranteed Operation of General Register [General Register] is used to mainly perform such functions as follows. -To store immediate values of a Load instruction -To store an expected value for comparing with an immediate value on a Compare instruction -To be used as a loop counter, in which a value is pre-incremented/pre-decremented, on a Compare instruction -To set the total number of pages written into [PageBuffer] on a self-run -To store the register values read/written from/to registers of the smshc If a value has been written to this register through a CPU during a self-run (when START=1), no proper operation is guaranteed.

Timer Register **0xF059001C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	1	1	TIMER												
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-															

TIMER	[28:16]	TIMER (Read-Only)
	These bits can work by using a Wait instruction or setting a timeout with a TPC instruction. The default value will be kept in these bits until another value is set by microcode. After microcode is completed, they will remain as 0. If these bits indicate any other value than 0, it means that it is counting. If a value is set here by microcode, it will start to count down toward 0. When it reaches 0, an event will occur. If waiting for an INT from memory stick through a Wait instruction, the ITOF bit of [Flag Register] becomes 1 as soon as the counted value of these bits reaches 0. Even before the counted value reaches 0, it becomes 0 when an INT occurs from memory stick. In the case of a simple Wait, it stops counting when it reaches 0, and moves on to a next instruction. Table 4.6 shows the values set to TimeCount and those set to the actual TIMER bit through a TPC instruction or a Wait instruction. No timeout will be set if 0000b has been set to TimeCount through a TPC instruction	

Table 4.6 The Values Set to TimeCount and TIMER

TimeCount (in microcode)	Waiting Time (ms) (TIMER/Cycle)	TimeCount (in microcode)	Waiting Time (ms) (TIMER/Cycle)
0000b	0	1000b	1500/Cycle
0001b	2/Cycle	1001b	2000/Cycle
0010b	8/Cycle	1010b	2250/Cycle
0011b	10/Cycle	1011b	3000/Cycle
0100b	15/Cycle	1100b	4500/Cycle
0101b	20/Cycle	1101b	5250/Cycle
0110b	150/Cycle	1110b	6000/Cycle
0111b	200/Cycle	1111b	7000/Cycle

The Waiting Times shown below could be shorter by 1ms at most depending on timing when DVEN rises.
Cycle: DVEN cycle [kHz]

e.g.
In the case that 1 kHz clock has been input to DVEN and the value set in TimeCount is 0010b:
Waiting Time = 8 ÷ 1 = 8 msec

Instruction Queue 0xF0590020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INST0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INST1															

INST0 / INST1	[31:16] / [15:0]	Instruction Queue
	This is a buffer for writing instructions in the Instruction Queue, which can store up to 256 words. It is capable of writing instructions of 256 words. Although the length of an instruction word is 16 bits, it is possible to transfer data from a host both by every 32 and 16 bits. If [Program Counter Register] holds an even-number address, this buffer is accessible both by every 32 and 16 bits. On the other hand, if [Program Counter Register] holds an odd-number address, this buffer is accessible only by every 16bits. Writing/executing an instruction is always performed to the address indicated by the PRGC bits of [Program Counter Register]. Therefore, an appropriate address shall be set in PRGC bits when an instruction is written/executed. When this register is read, the instruction written in the address pointed by PRGC bits will be simultaneously read. - This register shall be read while the START bit of [Control Register] remains 0. - Reading this register is prohibited while START bit is 1.	

General Data FIFO 0xF0590024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GDFIFO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GDFIFO															

GDFIFO	[31:0]	General Data FIFO
	This is a FIFO for sending/receiving general data. Selectable capacities include 16, 32, 64, and 128 bytes. One arbitrary size among them shall be chosen when implemented to an LSI. - While a self-run is being halted, the access size from a CPU is variable among 1, 2, or 4 bytes depending on the HSIZE terminal. - During a self-run, the access size from a CPU or a DMA controller is fixed to 4 bytes. - The data size ranges from 1 to 2048 bytes (selectable by every byte) to be sent/received by a single TPC through [General Data FIFO]. - The data size ranges from 1 to 16,380 bytes to be sent by a single self-run (due to the limitation of GFDN bit width）. - The data size is limitless to be received by a single Self-run.	

PageBuffer **0xF0590028**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PBUF															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBUF															

PBUF	[31:0]	PageBuffer
	This is a buffer for sending/receiving page data which capacity is 2048 bytes - The access size from a CPU or a DMA controller is fixed to 4 bytes. - The maximum data size available to be sent/received by a single TPC through [PageBuffer] is 2048 bytes. - The data size available to be sent/received by a single TPC through [PageBuffer] is selectable from 256, 512, 768, 1024,1280, 1536, 1792, or 2048 bytes. - A bank size, which can divide one of the transfer data sizes, should be set to the PBBC bit of [Memory Control Register]. - PageBuffer cannot be cleared during a self-run	

Table 4.7 Conversion of a [PageBuffer] Bank

PageBuffer Capacity	Conversion of a [PageBuffer] bank (PBBC of [Memory Control Register]	
2048 KByte	How a [PageBuffer] bank is converted varies depending on each TPC.	
	TPC for a 512-byte transfer	: The bank conversion is 256 bytes
	TPC for a 2048-byte transfer	: The bank conversion is 1024 bytes

Version Register **0xF059002C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VER3				VER2				VER1				VER0			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

VER3 to VER0	[31:16]	Version (Read-Only)
	The version of the smshc_i is represented to the bits from VER3 to VER0 by default. How a version number is represented is as follows. e.g. Version 2 . 5 2 c ver3 ver2 ver1 ver0 VER3: 2h VER2: 5h VER1: 2h VER0: Ch (When the value is 0h, the version number does not have a, b, c ...)	

MSHC-Command Register **0xF0590030**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPC				DSL	DSZ										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

TPC	[31:28]	Transfer Protocol Code
	These bits specify a TPC (4 bits).	

Table 4.8 Transfer Protocol Code

TPC	Transfer Protocol Code	
0x2	MS_RD_LDATA	Read Long Data Command
0x3	MS_RD_SDATA	Read Short Data Command
0x4	MS_RD_REG	Read Register Command
0x7	MS_GET_INT	Get Interrupt Command
0xD	MS_WR_LDATA	Write Long Data Command
0xC	MS_WR_SDATA	Write Short Data Command
0xB	MS_WR_REG	Write Register Command
0x8	MS_SET_RW_REG_ADRS	Set R/W Register Address Command.
0xE	MS_SET_CMD	Set Command
0x9	MS_EX_SET_CMD	Exit Set Command

DSL	[27]	Data Select
	This bit selects a send/receive buffer used for communicating with memory stick. 0: Sending/receiving data to/from memory stick through a FIFO of the smshc block 1: Sending/receiving data to/from memory stick through [PageBuffer] or [General Data FIFO]	

DSZ	[26:16]	Data Size
	These bits specify the length of data to be sent/received. The selectable length ranges from 1 to 2048 bytes. However, it is fixed to 2048 bytes when DSZ=0.	

A communication to memory stick can start by writing a value to this register.

Once the communication with memory stick has started, the RDY bit of [MSHC-Status Register] becomes 0. While the communication is underway, the RDY bit of [MSHC-Status Register] remains 0. The same bit will become 1 when the communication is completed.

Writing a value to this register is prohibited during a communication with memory stick (while the RDY bit of [MSHC-Status Register] is 0).

MSHC-Data Register **0xF0590034**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSDATA															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSDATA															

MSDATA	[31:0]	MSHC-Data
	This bit is the access register to a FIFO of the smshc block which can be initialized by setting 1 to the FCLR bit of [MSHC-FIFO Control Register]. The minimum unit size is 8 bytes to access the data in a FIFO of the smshc block. In the case that the data is smaller than 8 bytes, the lower bytes of the data will be invalid. When the internal FIFO is used to transfer data, the unit size is 8 bytes. In the case the actual data is smaller than 8 bytes (GET_INT and SET_RW_REG_ADRS) or indivisible by 8, the data size shall be made a multiple number of 8. 8-bit and 16-bit accesses are prohibited.	

MSHC-Status Register **0xF0590038**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	DRQ	MSINT	RDY	0	0	CRC	TOE	CED	ERR	BRQ	MSINT7	MSINT6	MSINT5	MSINT4	CNK
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-															

DRQ	[30]	DMA Request
	This bit will be set when a data transfer request occurs. This bit can be reset by accessing [MSHC-Data Register]. 1: A data transfer request exists 0: No data transfer request exists	

MSINT	[29]	MS I/F Interrupt
	This bit is set 1 by receiving an interrupt request INT from memory stick. When the SRAC bit of [MSHC-System Register] is 0, the causes of an interrupt request INT from the memory stick will be set to such bits of this register as CED, ERR, BRQ, CNK, MSINT7, MSINT6, MSINT5, and MSINT4. The MSINT bit will be reset by writing a value to [MSHC-Command Register]. 1: An interrupt request INT has been received from memory stick 0: No interrupt request INT has been received from memory stick yet	

RDY	[28]	Ready
	This bit is set 1 when a communication with memory stick is completed. If any error occurs while a protocol is being processed, the error status will be reflected on such bits as CRC and TOE. This bit will be reset by writing any value to [MSHC-Command Register]. 1: Ready to receive a command, or a protocol completed 0: Not ready to receive a command because a communication with memory stick is underway	

CRC	[25]	CRC Error
	If a CRC error occurs, 1 is set to this bit when the protocol is completed. This bit will be cleared by writing any value to [MSHC-Command Register]. 1: A CRC error occurs when receiving data 0: Normal end	

TOE	[24]	Timeout Error
	When a busy state continues longer than the number of clocks set in the BSY bits of [MSHC-System Register], it is considered that some malfunction has happened to memory stick, and the protocol will be consequently terminated. Then, 1 is set to this bit. This bit will be cleared by writing any value to [MSHC-Command Register]. 1: A RDY timeout error occurs at a handshake period in a communication with memory stick 0: Normal end	

CED	[23]	MS Command End
	This bit is effective only at the 4-bit and 8-bit parallel interface mode When a value is set to MSINT bit, the CED bit of the INT register on memory stick will be simultaneously reflected on this bit. This bit will be cleared by writing any value to [MSHC-Command Register].	

ERR	[22]	MS Error
	This bit is effective only at the 4-bit and 8-bit parallel interface mode When a value is set to MSINT bit, the ERR bit of the INT register on memory stick will be simultaneously reflected on this bit. This bit will be cleared by writing any value to [MSHC-Command Register].	

BRQ	[21]	MS Data Buffer Request
	This bit is effective only at the 4-bit and 8-bit parallel interface mode When a value is set to MSINT bit, the BREQ bit of the INT register on memory stick will be simultaneously reflected on this bit. This bit will be cleared by writing any value to [MSHC-Command Register].	

MSINT7 to 4	[20:17]	MS Interrupt 7 to 4
	This bits are effective only at the 8-bit parallel interface mode These bits can be expanded to indicate the causes of an interrupt from memory stick. When a value is set to MSINT bit, the causes of an interrupt status indicated by the bits from DATA7 to DATA4 on memory stick will be simultaneously reflected on these bits. These bits will be cleared by writing any value to [MSHC-Command Register].	

CNK	[16]	MS Command No Acknowledge
	This bit is effective only at the 4-bit and 8-bit parallel interface mode When a value is set to MSINT bit, the CMDNK bit of the INT register on memory stick will be simultaneously reflected on this bit. This bit will be cleared by writing any value to [MSHC-Command Register].	

MSHC-System Register **0xF059003C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RST	INTEN	MSIEN	DRQSL	INTCLR	0			SRAC	EIGHT	MSPIO1	MSPIO0	NOCRC	BSY		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-															

RST	[31]	Synchronous Reset
	When 1 is written to this bit, the smshc block will be rest. It will be cleared to 0 after the internal resetting is completed.	

INTEN	[30]	XINT Enable
	1: Enables the smshc block to output an interrupt request 0: Disables the smshc block to output an interrupt request	

MSIEN	[29]	MS Interrupt Enable
	1: Enables receiving an interrupt request from memory stick 0: Disables receiving an interrupt request from memory stick	

DRQSL	[28]	DRQ Select
	1: Enables the outputting of XINT when the smshc block requests data (DRQ=1) 0: Disables the outputting of XINT when the smshc block requests data (DRQ=1)	

INTCLR	[27]	INT Clear
	An interrupt from the smshc block is cleared by setting 1 to this bit. This bit is cleared to 0 after the interrupt has been cleared.	

SRAC	[23]	Serial Access Mode
	1: Serial interface mode 0: Parallel interface mode	

EIGHT	[22]	Eight Line Parallel Access Mode
	0: 4-bit parallel interface mode 1: 8-bit parallel interface mode	

MSPIO1	[21]	Memory Stick Parallel I/O No.1
	The value of MSPIO1 is output from an external terminal, mspio[1]. When the MSPIO1 bit of [MSHC-System Register] is updated, the MSPIO1 bit of [MSHC-User Custom Register] is also updated. 0: Outputs Low from mspio[1] (Default) 1: Outputs High from mspio[1]	

MSPIO0	[20]	Memory Stick Parallel I/O No.0
	The value of MSPIO0 is output from an external terminal, mspio[0]. When the MSPIO0 bit of [MSHC-System Register] is updated, the MSPIO0 bit of [MSHC-User Custom Register] is also updated. 0: Outputs Low from mspio[0] (Default) 1: Outputs High from mspio[0]	

NOCRC	[19]	No CRC
	When 1 is set to this bit, a write protocol is executed without any CRC data (16 bits) attached to the end of data to be sent. For a read protocol, checking a CRC is performed as usual no matter what the value set to this bit is. This bit shall be 0 unless otherwise specified. 1: Outputting a CRC off 0: Outputting a CRC on	

BSY	[18:16]	Busy Count
	These bits set a RDY timeout period. The maximum BSY period for waiting an RDY signal from memory stick can be calculated by multiplying the value set in this bit x 4 SCLKs When BSY=0, no timeout is detected. To wake up a Memory Stick Ver. 1.X, this bit shall be set 0 before starting a protocol. This is because the RDY timeout is usually longer when waking up a Memory Stick Ver. 1.X. While a protocol is being executed, the value in this bit shall not change.	

Table 4.9 Interface Mode Configuration

SRAC	EIGHT	Interface Mode
0	0	4-bit parallel interface mode
0	1	8-bit parallel interface mode
1	0	Serial interface mode
1	1	Prohibited to be set

MSHC-User Custom Register **0xF0590040**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSPIO[7:2]						MSPIO 1	MSPIO 0	0		EMP	FUL	0			NDLT
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-															

MSPIO	[31:26]	Memory Stick Parallel I/O
	The value of MSPIO[7:2] is output from an external terminal, mspio[7:2].	

MSPIO1	[25]	Memory Stick Parallel I/O No.1
	The value of MSPIO1 is output from an external terminal, mspio[1]. When the MSPIO1 bit of [MSHC-User Custom Register] is updated, the MSPIO1 bit of [MSHC-System Register] is also updated. 0: Outputs Low from mspio[1] 1: Outputs High from mspio[1] (Default)	

MSPIO0	[24]	Memory Stick Parallel I/O No.0
	The value of MSPIO0 is output from an external terminal, mspio[0]. When the MSPIO0 bit of [MSHC-User Custom Register] is updated, the MSPIO0 bit of [MSHC-System Register] is also updated. 0: Outputs Low from mspio[0] (Default) 1: Outputs High from mspio[0]	

EMP	[21]	FIFO Empty
	When 1 is set to the FCLR bit of [MSHC-FIFO Control Register], this bit is also set 1. 1: FIFO of the smshc block is empty 0: Data exists in FIFO of the smshc block	

FUL	[20]	FIFO Full
	When 1 is set to the FCLR bit of [MSHC-FIFO Control Register], this bit is reset to 0. 1: FIFO of the smshc block is full 0: FIFO of the smshc block is not full	

NDLT	[20]	Next Data Latch Timing
	Default = 0 (normal data latch) 0: Latches data at a rising edge right after the data input timing. (Default) 1: Latches data at a rising edge two cycles after the data input timing	

MSHC-FIFO Control Register 0xF0590044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0					AFCLR	FCLR	FDIR	0						DRSZ	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-															

AFCLR	[26]	Auto FIFO Clear
	The FIFO data of the smshc block will be automatically initialized by setting 1 to this bit. When both a protocol and a data transfer are completed, this bit will be initialized.	

FCLR	[25]	FIFO Clear
	The FIFO data in the smshc block will be initialized by setting 1 to this bit. It will be cleared to 0 when initializing the FIFO data is completed.	

FDIR	[24]	FIFO Direction
	When a protocol starts, the value of TPC[3] of [MSHC-Command Register] will be reflected on this bit. 1: The direction of FIFO in the smshc block shall be the direction for sending data (CPU → FIFO of the smshc block → memory stick) 0: The direction of FIFO in the smshc block shall be the direction for receiving data (CPU ← FIFO of the smshc block ← memory stick)	

DRSZ	[17:16]	Data Request Size
	These bits set a slice size for a data transfer of FIFO in the smshc block.	

Table 4.10 DMA Slice Size Configuration

DRSZ[1:0]	DMA Slice Size
00b	4Byte
01b	8Byte
10b	16Byte
11b	Prohibited to be set

MSHC-DMA Control Register **0xF059004C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAON															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

DMAON	[31:16]	DMA Request Enable
	These bits enable outputting a DMA transfer request signal depending on each value set in the TPC bit of [MSHC-Command Register]. 0: When a data transfer request occurs, xdrq is not asserted 1: When a data transfer request occurs, xdrq is asserted e.g. DMAON[15]=1: DMA transfer request signal will be valid when issuing TPC = 1111b DMAON[13]=1: DMA transfer request signal will be valid when issuing TPC = 1101b (WRITE_LONG_DATA) DMAON[7]=0: DMA transfer request signal will be invalid when issuing TPC = 0111b (GET_INT)	

PORTCFG Register **0xF05F1000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BM			DVEN	-										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-													PORT		

BM	[30]	Byte Order Mode Selection
	Select Byte order 0:LSB(Least Significant Byte) first mode (default) 1:MSB(Most Significant Byte) first mode	

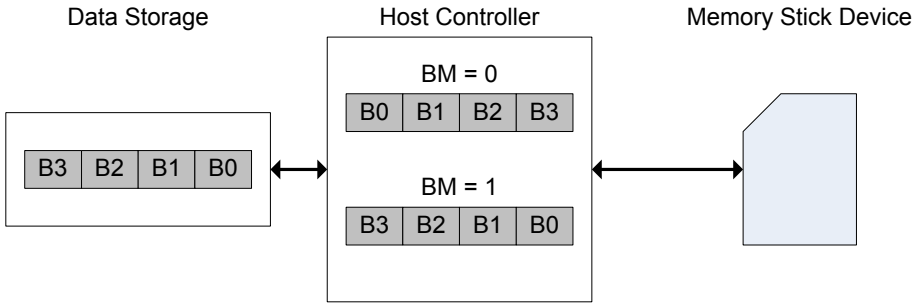


Figure 4.3 Memory Stick Byte Order Mode

DVEN	[27]	DVEN
	Timer count enable signal (1KHz to 2KHz) The maximum frequency will be half of HCLK frequency. For more about the operation of this bit, refer to the description of [Timer Register].	

PORT	[2:0]	Port Number
	It specifies the port number to be used by the memory stick host controller. (0~5)	

PORTDLY Register **0xF05F1004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-		DLY_OEN			DLY_BS			DLY_DATA7			DLY_DATA6			DLY_DATA5	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DLY_DATA5	DLY_DATA4			DLY_DATA3			DLY_DATA2			DLY_DATA1			DLY_DATA0		

DATA0-7	[23:0]	DATA[7:0] Output Delay Configuration
	It controls output delays of each Memory Stick DATA signal. 3 bits are assigned and it ranges from 0 to 7. 0: no delay 1~7: about max. 500ps* DLY_DATAn	

BS	[26:24]	BS Output Delay Configuration
	It controls output delays of each Memory Stick BS signal. 3 bits are assigned and it ranges from 0 to 7. 0: no delay 1~7: about max. 500ps* DLY_BSn	

OEN	[23:0]	OEN Output Delay Configuration
	It controls output delays of all Memory Stick output signal. 3 bits are assigned and it ranges from 0 to 7. 0: no delay 1~7: about max. 500ps* DLY_OEN	

4.3 Operation

4.3.1 Operation during Reset

When 1 is set to the SRST bit of [System Register], the registers and operational sequences inside the I-CON block will be initialized. After this synchronous reset is completed, the SRST bit will be automatically cleared to 0.When 1 is set to the RST bit of [MSHC-System Register], the registers and operational sequences inside the smshc block will be initialized. After this synchronous reset is completed, the RST bit will be automatically cleared to 0.

If a reset is performed during a communication with memory stick, the bus state of host controller might disagree to that of the memory stick. In addition, when such a reset occurs in the middle of a communication with memory stick, the power supply of the memory stick shall be also reset.

4.3.2 Operation Flow

The smshc_i can achieve a communication with memory stick by starting up the smshc and issuing a TPC. There are two methods to start up the smshc; one is through a host CPU while the other one is through the I-CON.

4.3.3 Communication Method to Start up smshc through Host CPU

A TPC and the data size to be sent are written to [MSHC-Command Register] by a CPU. This is how the smshc can start communicating with memory stick (issuing a TPC).

The basic flow is as follows:

(3) Writes values to [MSHC-Command Register]
(4) Accesses [MSHC-Data Register] (Write / Read)
(5) Completes a TPC （[MSHC-Status Register] RDY = 1）

Described below are the basic explanation of write TPCs and read TPCs.

Write TPC

In the case that a write TPC is issued, the smshc communicates with memory stick if a TPC, the size of data to be transferred, and the data itself are set by a host CPU. The smshc block automatically reverses the TPC in BS1, adds CRC data in BS2, and detects a BSY/RDY signal in BS3. Transferring data can be achieved using [MSHC-Data Register] (FIFO) or using either [PageBuffer] or [General Data FIFO]. The selection of a method to write data can be set to the DSL bit of [MSHC-Command Register].
When a TPC is completed, the RDY bit of [MSHC-Status Register] will be set.

After having written data to [MSHC-Command Register], writing any value to the following registers is prohibited until the TPC is completed ([MSHC-Status Register] RDY = 0); [MSHC-Command Register], [MSHC-System Register], [MSHC-FIFO Control Register], and [MSHC-DMA Request Control Register]. If written, proper operation cannot be guaranteed.

To transfer data through [PageBuffer] or [General Data FIFO], microcode with a TPC instruction, which utilizes [PageBuffer] or [General Data FIFO] respectively, shall be executed. Neither [PageBuffer] nor [General Data FIFO] can be used without a self-run.

- When [MSHC-Data Register] is used, data of the FIFO size (32 bytes) can be stored in [MSHC-Data Register] in advance.
- If no transfer data can be prepared in time, SCLK (Memory Stick I/F clock) will stop to wait for the data to be transferred.
- If no RDY signal is returned from the memory stick resulting in a timeout during BS3, not only the TOE bit of [MSHC-Status Register] but also the RDY bit will be set. The duration of the RDY timeout can be set at the BSY bits of [MSHC-System Register].

Read TPC

In the case that a read TPC is issued, the smshc communicates with memory stick if a TPC and the size of data to be transferred are set by a host CPU. The smshc automatically reverses the TPC in BS1, detects a BSY/RDY signal in BS2, and checks CRC data in BS3. Reading data can be achieved using [MSHC-Data Register](FIFO) or using either [PageBuffer] or [General Data FIFO]. The selection of a method to read data can be set to the DSL bit of [MSHC-Command Register].
When a TPC is completed, the RDY bit of [MSHC- Status Register] will be set.

After having written a value to [MSHC-Command Register], writing any value to the following registers is prohibited until the TPC is completed ([MSHC-Status Register] RDY = 0); [MSHC-Command Register], [MSHC-System Register], [MSHC-FIFO Control Register], and [MSHC-DMA Request Control Register]. If written, proper operation cannot be guaranteed.

To transfer data through [PageBuffer] or [General Data FIFO], microcode with a TPC instruction, which utilizes [PageBuffer] or [General Data FIFO] respectively, shall be executed. Neither [PageBuffer] nor [General Data FIFO] can be used without a self-run.

- If no transfer data can be prepared in time, SCLK (Memory Stick I/F clock) will stop to wait for the data to be transferred.
- If no RDY signal is returned from the memory stick resulting in a timeout during BS2, not only the TOE bit of [MSHC-Status Register] but also the RDY bit will be set. The duration of a RDY timeout can be set at the BSY bits of [MSHC-System Register].
- During BS3, if a CRC calculation of data results in an inconsistency, not only the RDY bit of [MSHC-Status Register] but also the CRC bit will be set. This announces the lack of data reliability.

Shown below is an example of a communication with Memory Stick PRO. This example illustrates the flow of a READ_DATA command. A host CPU directly accesses the smshc registers (0x30 to 0x4C). This method makes it easier to issue TPCs, however, more workload will be imposed on the host CPU.

The ▭ portion shows a single cycle of TPC issuance or the process for an interrupt from Memory Stick PRO.

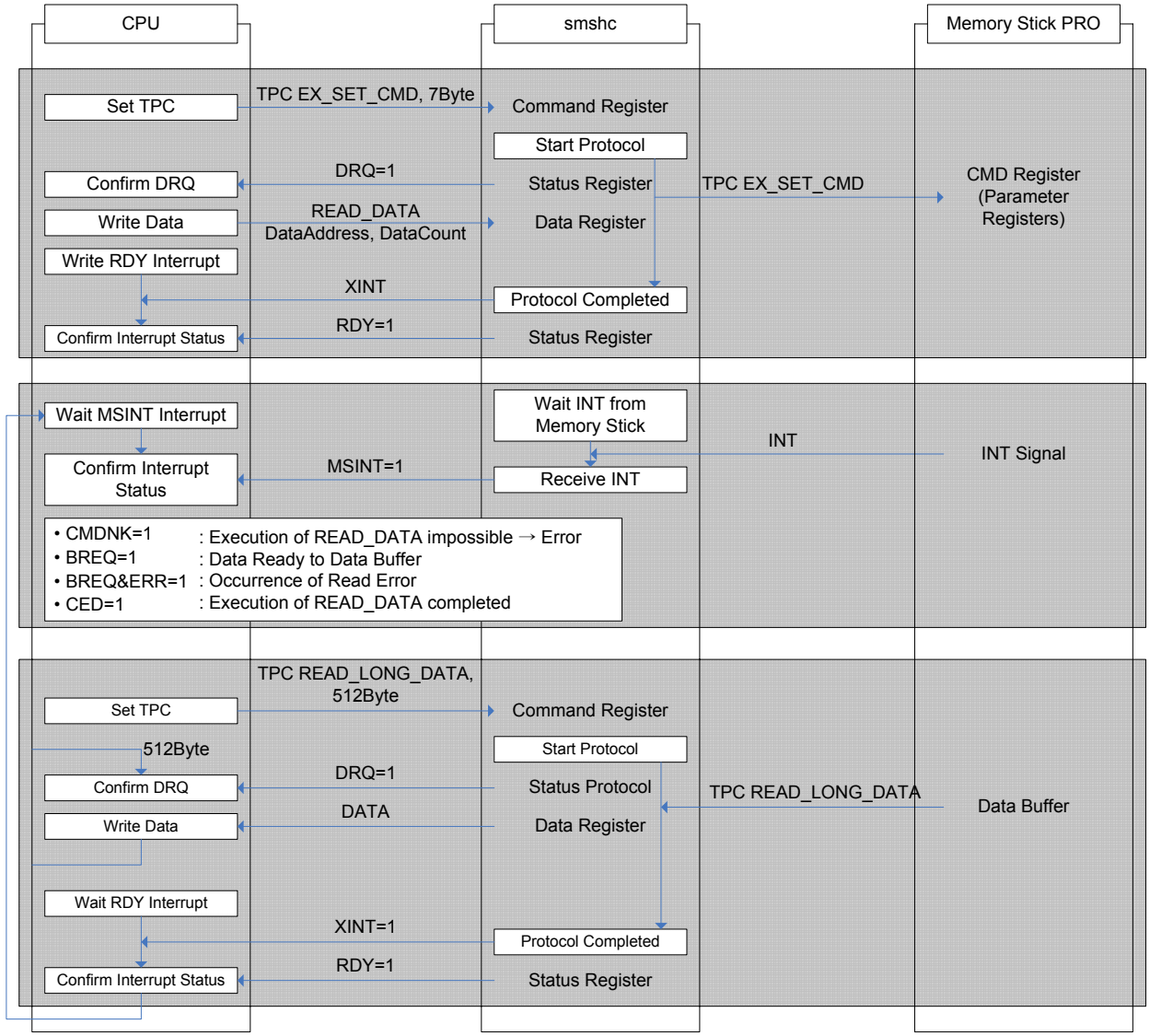


Figure 4.4 Communication Example for Starting up smshc through CPU

4.3.4 Communication Method to Start up smshc through I-CON

The I-CON accesses the smshc registers (0x30 to 0x4C). The details of instructions for starting up the smshc are all gathered into the I-CON as microcode. The I-CON can start up the smshc and handle interrupts on behalf of a host CPU. This method can decrease the workload imposed on the host CPU, however, microcode are required.
Shown below is an example of a communication with Memory Stick PRO. This example illustrates the flow of a READ_DATA command.

The  portion shows a single cycle of TPC issuance or the process for an interrupt from Memory Stick PRO.

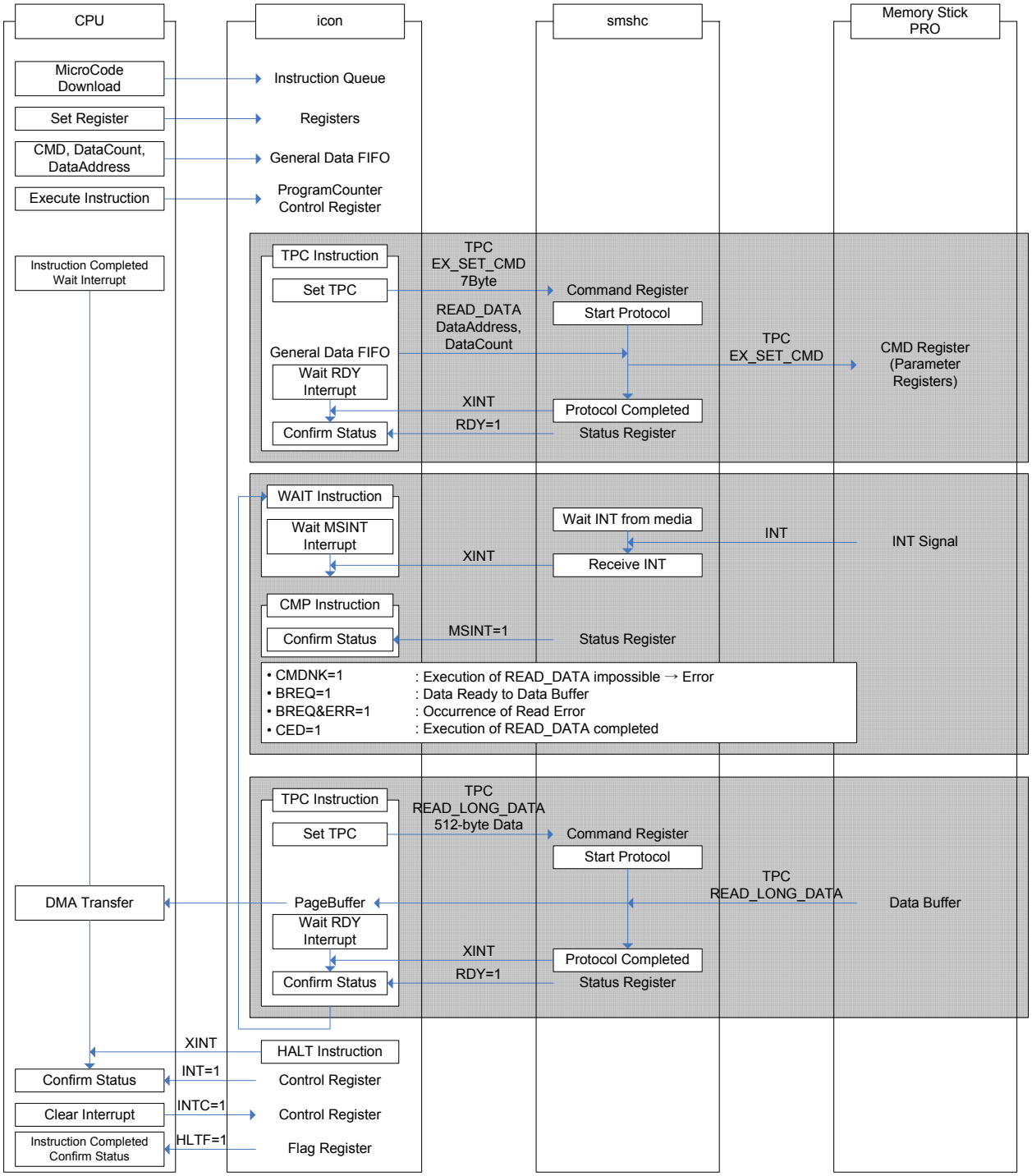


Figure 4.5 Communication Example for starting up smshc through I-CON

4.3.5 Interface Mode Switching Sequence

The host controller supports three interface mode, serial, 4-bit parallel, 8-bit parallel. The interface mode on the host controller shall be switched after the interface mode on the memory stick has been successfully completed through a WRITE_REG TPC.(a RDY signal is received.) The sequence of switching interface mode is described in this section..

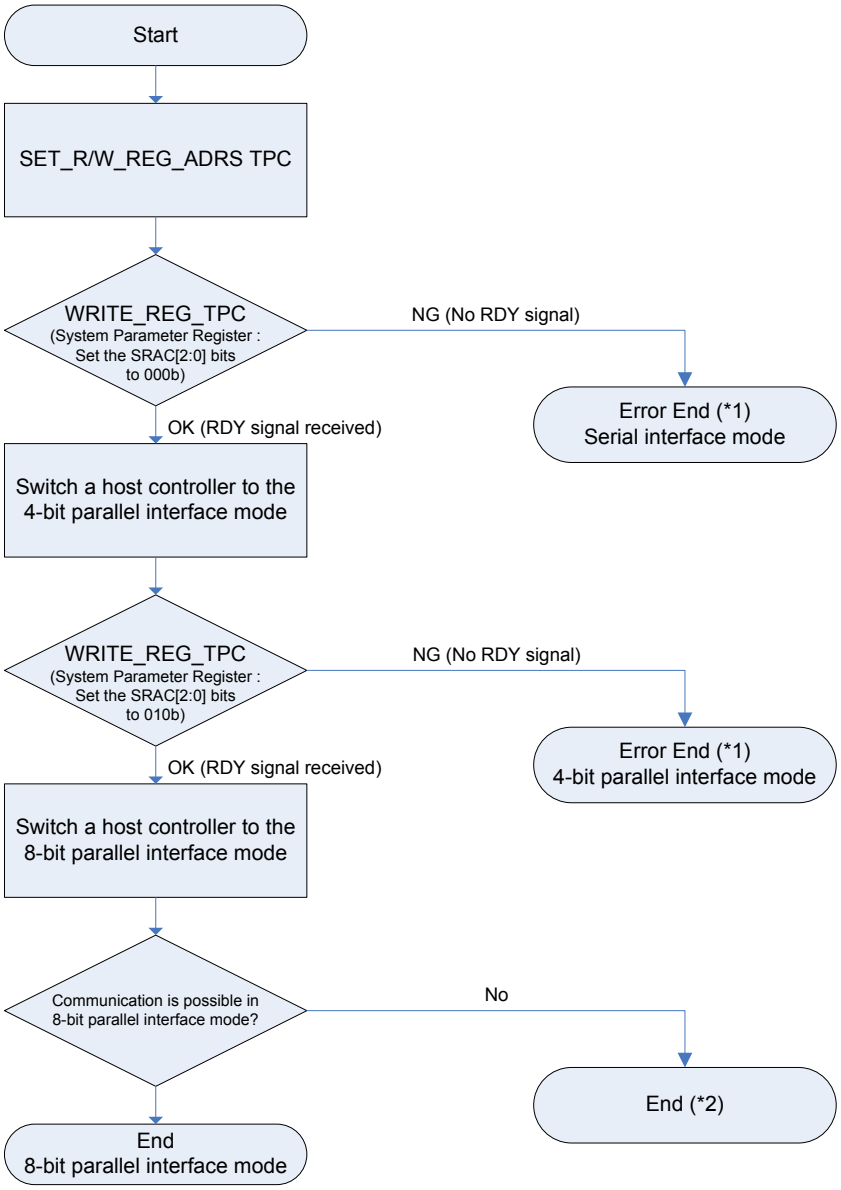


Figure 4.6 Sequence to Switch Interface Mode

Switching Interface Mode through CPU

This test accesses the smshc controller's registers, and issues TPCs after switching to a 4-bit parallel interface mode. Specifically, it issues SET_RW_REG_ADRS as a write TPC and GET_INT as a read TPC; checks for an interrupt after a TPC transfer is complete; clears an interrupt when it occurs; and checks RDY, TOE and CRC of the MSHC-Status Register. The setting flow (1)~(14) is for 4-bit interface mode and (15)~(26) is for checking the previous settings .

- Enable external interrupts to occur from smshc_i

(1) Set the INTEN bit of the MSHC-System Register to enable external interrupts to occur from smshc_i.

- Switch to 4-bit parallel interface.

(2) In the MSHC-Command Register, set as follows: TPC=SET_RW_REG_ADRS and DSZ=0x4.
(3) Continue to read the DRQ bit of the MSHC-Status Register until it is set to 1.
(4) Write 8-byte data (0x0101_1001, 0x0000_0000) to the MSHC-Data Register.
(5) Wait for an XINT interrupt to occur.
(6) Read the MSHC-Status Register to make sure everything is completed successfully.
(7) Set the INTCLR bit of the MSHC-System Register to clear the XINT interrupt.
(8) In the MSHC-Command Register, set as follows: TPC=WRITE_REG and DSZ=0x1.
(9) Continue to read the DRQ bit of the MSHC-Status Register until it is set to 1.
(10) Write 8-byte data (0x0000_0000, 0x0000_0000) to the MSHC-Data Register.
(11) Wait for an XINT interrupt to occur.
(12) Read the MSHC-Status Register to make sure everything is completed successfully.
(13) Set the INTCLR bit of the MSHC-System Register to clear the XINT interrupt.
(14) In the MSHC-System Register, set as follows: SRAC=0.

Additional steps below are to be done in 8-bit parallel interface setting mode after 4-bit interface mode settings.

- Switch to 8-bit parallel interface.

(15) In the MSHC-Command Register, set as follows: TPC=WRITE_REG and DSZ=0x1.
(16) Continue to read the DRQ bit of the MSHC-Status Register until it is set to 1.
(17) Write 8-byte data (0x4000_0000, 0x0000_0000) to the MSHC-Data Register.
(18) Wait for an XINT interrupt to occur.
(19) Read the MSHC-Status Register to make sure everything is completed successfully.
(20) Set the INTCLR bit of the MSHC-System Register to clear the XINT interrupt.
(21) In the MSHC-System Register, set as follows: SRAC=0 and EIGHT=1.

Following steps (22)~(33) checks if mode switching is completed or not.

- Transfer a write TPC.

(22) In the MSHC-Command Register, set as follows: TPC=SET_RW_REG_ADRS and DSZ=0x4.
(23) Continue to read the DRQ bit of the MSHC-Status Register until it is set to 1.
(24) Write arbitrary 8-byte data to the MSHC-Data Register.
(25) Wait for an XINT interrupt to occur.
(26) Read the MSHC-Status Register to make sure everything is completed successfully.
(27) Set the INTCLR bit of the MSHC-System Register to clear the XINT interrupt.

- Transfer a read TPC.

(28) In the MSHC-Command Register, set as follows: TPC=GET_INT and DSZ=0x1.
(29) Continue to read the DRQ bit of the MSHC-Status Register until it is set to 1.
(30) Read the MSHC-Data Register. The expected value is as follows: 0x8000_0000, 0x0000_0000.
(31) Wait for an XINT interrupt to occur.
(32) Read the MSHC-Status Register to make sure everything is completed successfully.

(33) Set the INTCLR bit of the MSHC-System Register to clear the XINT interrupt.

Switching Interface Mode through I-CON - 8bit Parallel Interface Mode

To operate with I-CON, microcode must be prepared in [Instruction Queue]..If START bit of [Control Register] is set, the microcode is read and executed in the order saved. The detailed setting flow described in the table below.

Table 4.11 Microcode for Interface Mode Switching

Program Counter	Microcode Instruction	Description
0x0000	0x2C20	TPC instruction to send an immediate value; uses [Data Register] of smshc.
0x0001	0x8004	SET_RW_REG_ADRS | SIZE=0x4
0x0002	0x0101	RD_ADRS=0x1; RD_SIZE=0x1
0x0003	0x1001	WR_ADRS=0x10; WR_SIZE=0x1
0x0004	0x2720	Instruction to send a TPC to memory stick using [General Data FIFO]
0x0005	0xB001	WRITE_REG | SIZE=0x1
0x0006	0x60BC	Stores the value of [General Register0] to the [System Register] of smshc
0x0007	0x0001	HALT instruction; EXIT Code=0x1

In switching interface mode through CPU, mode setting data, to be transferred to external memory stick, would be moved from common memory space with a help from DMA controller. The common memory space should be kept occupied previously. 4-bit parallel interface mode setting sequence is as follows.

- Enable external interrupts to occur from smshc_i.
- Write microcode instructions to the Instruction Queue.(refer to Table 4.11)
- Write in the common memory space the data to be sent by WRITE_REG to switch to a 4-bit parallel interface.
- Perform settings to start DMA0.

(34) The slice size is 4 bytes, and therefore memory-to-peripheral transfer shall be performed once.

- Start smshc_i (to switch to 4-bit parallel interface).

(35) Set the Memory Control Register as follows: GFDN=0x1.
(36) Write 0x6005_0000 in the General Data Registers 0 to 1.
(37) Write 0x0000_0000 in General Data Registers 2 to 3.
(38) Set the System Register as follows: DMAE0=1, DMACH=0, DMASL = 00b, GDSZ=00b, INTE=1 and DMAE1=1.
(39) Set Program Counter Register as follows: PRGC=0x00.
(40) Set Control Register as follows: GDIR=1 and START=1.
(41) Wait for an XINT interrupt to occur. DMA data transfer is being performed...
(42) Check the Flag Register to make sure data is set as follows: HLTF=1, EXTS=0x1.
(43) Set the Control Register as follows: INTC=1.

Additional steps below are to be done in 8-bit parallel interface setting mode after 4-bit interface mode settings

- Write in the Shared SRAM the data to be sent by WRITE_REG to switch to an 8-bit parallel interface.
- Perform settings to start DMA0.

(44) The slice size is 4 bytes, and therefore memory-to-peripheral transfer shall be performed once.

- Start smshc_i (to switch to 8-bit parallel interface).

(45) Set the Memory Control Register as follows: GFDN=0x1 and GFCR=1.
(46) Write 0x6045_0000 in General Data Registers 0 to 1.
(47) Write 0x0000_0000 in General Data Registers 2 to 3.
(48) Set the System Register as follows: DMAE0=1, DMACH=0, DMASL = 00b, GDSZ=00b, INTE=1 and DMAE1=1.
(49) Set Program Counter Register as follows: PRGC=0x00.
(50) Set the Control Register as follows: GDIR=1 and START=1.

(51) Wait for an XINT interrupt to occur. DMA data transfer is being performed...
(52) Check the Flag Register to make sure data is set as follows: HLTF=1, FLG=1 and EXTS=0x1.
(53) Set the Control Register as follows: INTC=1.

4.4 Timing Diagram

4.4.1 Serial Interface Mode

TPC Transfer State (BS1)

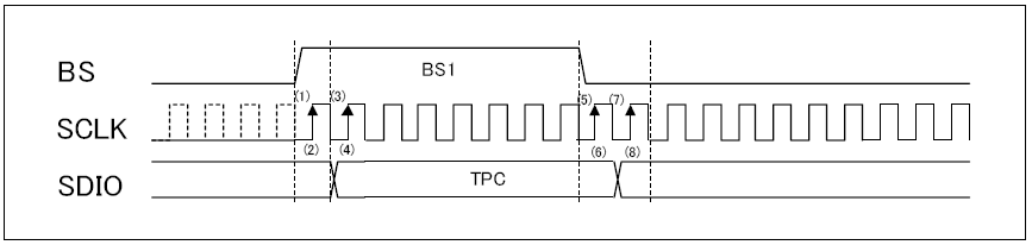


Figure 4.7 Serial Protocol TPC Transfer State (BS1)

Operation	Description
Host	The host controller outputs High at point (1) on BS above, and starts the output of SCLK before point (2). Starts output the MSB of a TPC on the SDIO at point (3). Switches the BS to Low at the same time the LSB of the TPC is output at point (5). BS2 starts from point (7).
Memory Stick	The Memory Stick detects BS as High at point (2) above, and SDIO is put in a High Impedance (input) state between point (2) and point (3). Receive the MSB of the TPC at the next point (4). Detects the BS as Low at the same time the LSB of the TPC is received at point (6) and starts BS2 from point (7).

Data Transfer State (Read Packet : BS3)

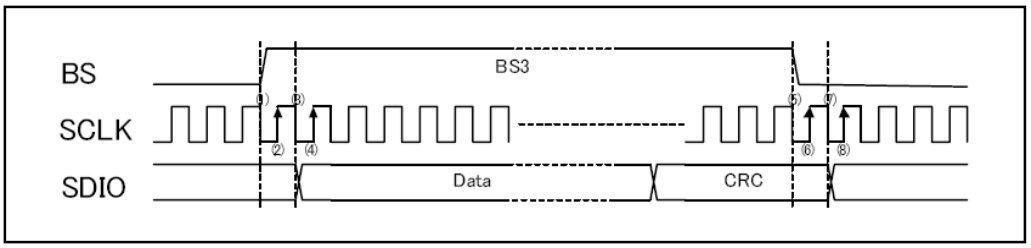


Figure 4.8 Serial Protocol Data Transfer State (BS3)

Operation	Description
Host	In a Read Packet, after switching the BS to High at point (1) above, the host controller receives the MSB of the first data at point (4). After switching the BS to Low at point (5), the LSB of the CRC is received at point (6) and the Read Packet completes.
Memory Stick	Detects the Bs as High at point (2) above, and starts outputting the MSB of the first data at point (3). After outputting the LSB of the CRC at point (5), the BS is detected as Low and then enters into BS0 from point (7).

Data Transfer State (Write Packet : BS2)

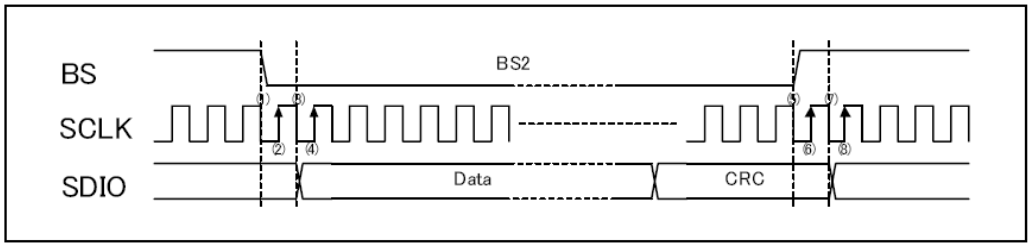


Figure 4.9 Serial Protocol Data Transfer State (BS2)

Operation	Description
Host	For a Write Packet, after switching the BS to Low at point (1) above, transferring the MSB of the first data is started at point (3) with a 1 SCLK delay. Transfers the 16-bit CRC for all the data immediately after the LSB of the last data. Switches the BS to High at the same time the LSB of the CRC is sent at point (5), and then starts detecting the RDY signal of BS3 from point (8).
Memory Stick	The Bs is detected as Low at point (2) above, and receiving of the signal will start with the MSB of the first data at point (4). The BS is detected as High simultaneously to receiving the LSB of the CRC at point (6) and then starts outputting the BSY/RDY signal of BS3 from point (7).

Handshake State (Read Packet : BS2 / Write Packet : BS3)

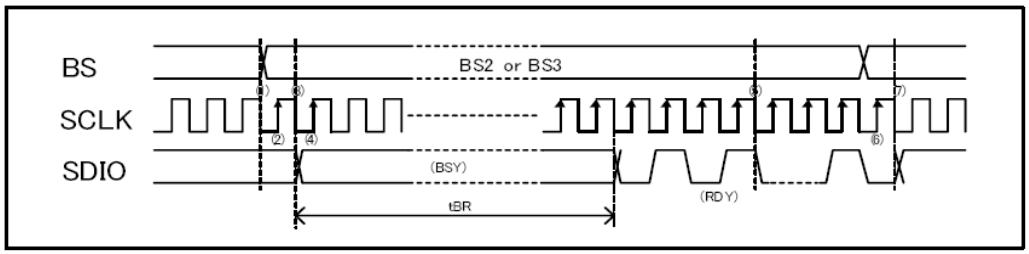


Figure 4.10 Serial Protocol Handshake State

Operation	Description
Host	After switching the BS at point (1) above, starts to detect the RDY signal from point (4). After receiving the RDY signal, which reversing for each SCLK, for more than or equal to 4 SCLKs, switches the BS signal at the falling edge of the SCLK after point (5) and the next Bus State begins with point (7).
Memory Stick	After detecting the switch of Bus State at point (2) above, starts to output of the BSY/RDY signal from point (3). After the Memory Stick has completed its internal processing and outputs the RDY signal, it detects the switch of Bs at point (6) and then the next BS starts from point (7).

INT Transfer State (BS0)

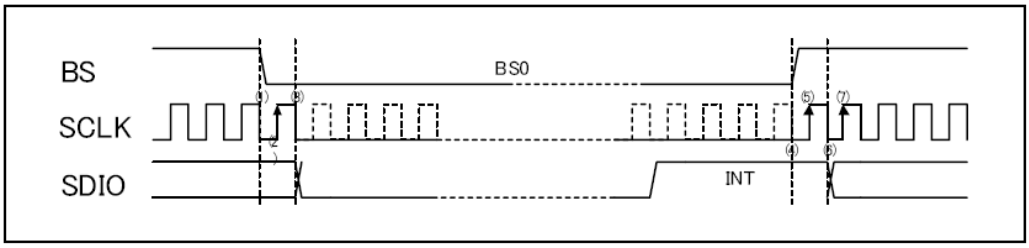


Figure 4.11 Serial Protocol INT Transfer State

Operation	Description
Host	SCLK can be halted after the point (3) above, if necessary. (SCLK is Low) SDIO shall be in a Hi-Z state (input) after point (3). If an INT signal (SDIO is High) if detected during the INT Transfer State period, the host controller shall confirm the cause of interrupt by read out INT Register of a Memory Stick.

Memory Stick	Detects BS as Low at point (2) above, and starts the INT Transfer State at point (3). SDIO shall be in the output status after point (3), and outputs Low until the INT signal is output. The cause of interrupt shall be reflected to the INT Register and the INT signal (SDIO is High) shall be output for the period of the INT Transfer State.

4.4.2 4-bit Parallel Interface Mode

TPC Transfer State (BS1)

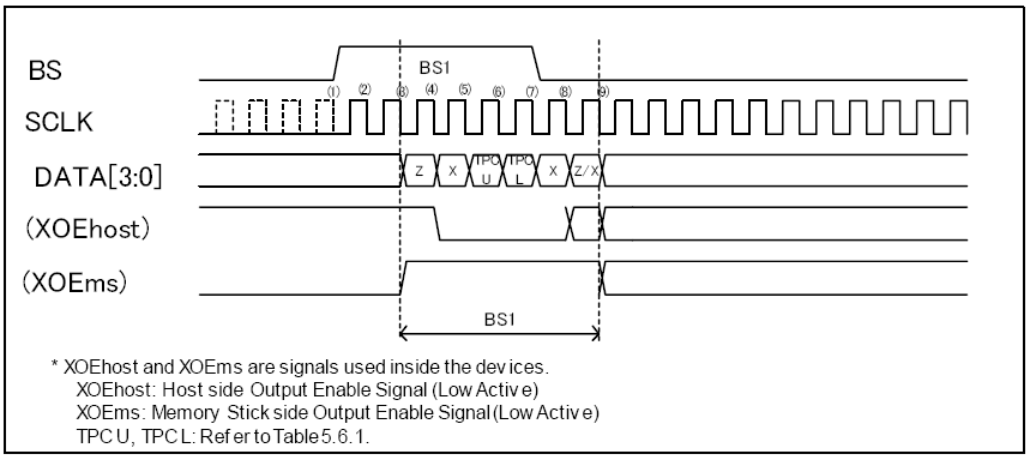


Figure 4.12 4-bit Parallel Protocol TPC Transfer State

Operation	Description
Host	It outputs the BS as High at the above point (1), and starts the provision of SCLK before point (2). It starts the output of the upper 4 bits of the TPC on DATA[3:0] at point (5). Switches the BS to Low at point (7) and then starts BS2 from point (9).
Memory Stick	It detects the BS as High at the above point (2), and receives the upper 4 bits of the TPC at point (6). Receives the lower 4 bits at point (7). The BS is detected as Low at point (8), and starts BS2 from point (9).

Data Transfer State (Read Packet : BS3)

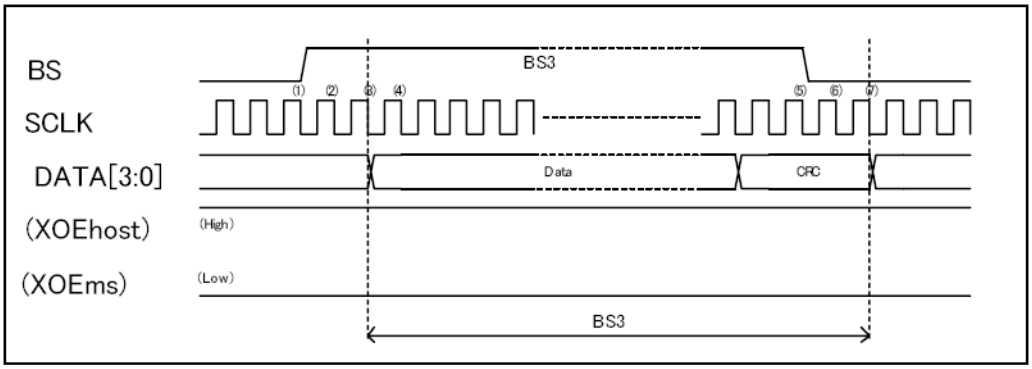


Figure 4.13 4-bit Parallel Protocol Data Transfer State (BS3)

Operation	Description
Host	For a Read Packet, after switching the BS to High at point (1) above, receiving the signal will start with the upper 4 bits of the first data at point (4), with a delay of 3 SCLKs. After switching the BS to Low at point (5), the lower 4 bits of the 16 bit CRC are received at point (7), and the Read Packet completes.
Memory Stick	The BS is detected as High at point (2) above, and the upper 4 bits of the first data are output at point (3). At point (6), the Bs is detected as Low and the lower 4 bits of the 16-bit CRC are output. The transition to BS0 is made at point (7).

Data Transfer State (Write Packet : BS2)

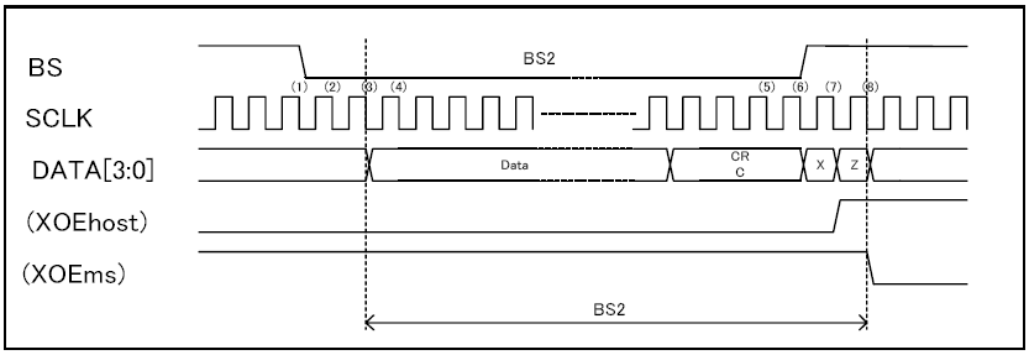


Figure 4.14 4-bit Parallel Protocol Data Transfer State (BS2)

Operation	Description
Host	For a Write Packet, after switching the BS to Low at point (1) above, starts to transfer the signal with the upper 4 bits of the first data at point (3), with a delay of 2 SCLKs. The 16-bit CRC for all of the data is transferred from the next SCLK after the last data. The BS is switched to High at point (6).
Memory Stick	The BS is detected as Low at point (2) above, and starts to receive the signal with the upper 4 bits of the first data at point (4). After receiving the lower 4 bits of the 16-bit CRC at point (6), the BS is detected as High at point (7), and the BSY signal or RDY signal of BS3(Handshake State) starts outputting from point (8).

Handshake State (Read Packet : BS2 / Write Packet : BS3)

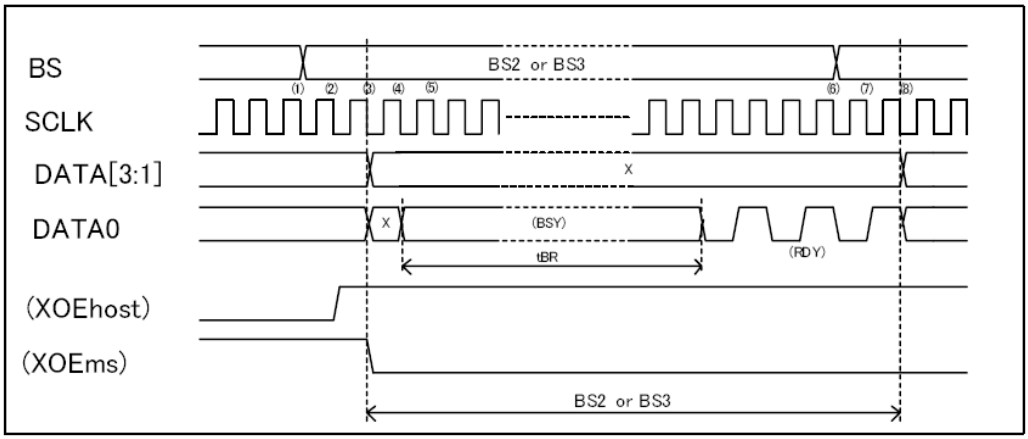


Figure 4.15 4-bit Parallel Protocol Handshake State

Operation	Description
Host	After switching the BS at point (1) above, the detection of a RDY signal is started from point (5). After receiving the RDY signal, which is reversed for each SCLK, for more than or equal to 4 SCLKs, the BS signal is switched at the falling edge of the SCLK after point (6), and the next Bus State begins at point (8).
Memory Stick	After detecting the switch of BS at point (2) above, BSY/RDY signal are output at point (3). After outputs the RDY signal by the completion of internal processing, the Memory Stick detects the switching of BS at point (7), and starts the next Bus State at point (8).

INT Transfer State (BS0)

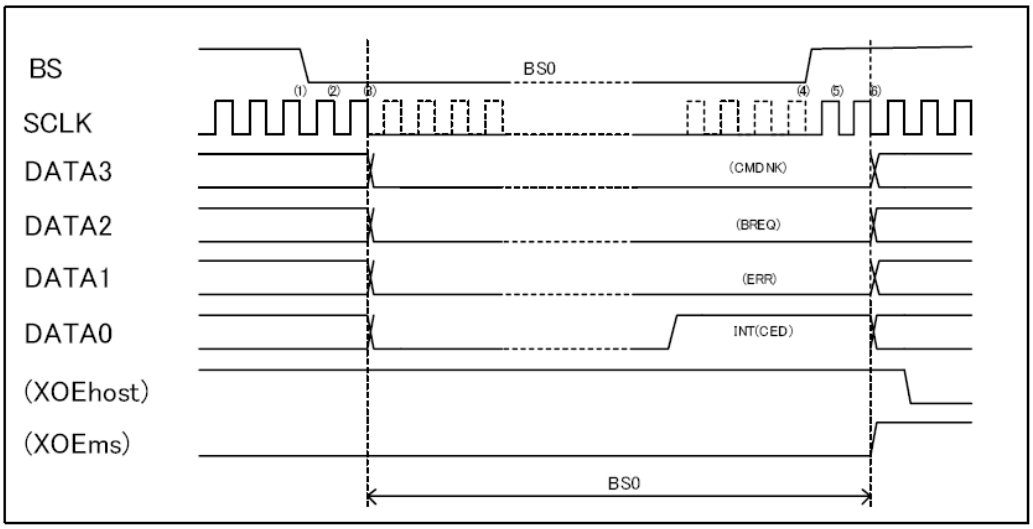


Figure 4.16 4-bit Parallel Protocol INT Transfer State

Operation	Description
Host	SCLK can be halted after point (3). If one of DATA[3:0] becomes High in the INT Transfer State, it will be detected as an INT signal. Confirms the interrupt factor from signals lines in High Level.
Memory Stick	It detects BS0 at point (2) above, and starts the INT transfer State at point (3). After point (3), DATA[3:0] shall be in the output status, and output 0 until the INT signals are output. The cause of interrupt shall be reflected to the INT Register and the INT signals are output on DATA[3:0]

4.4.3 8-bit Parallel Interface Mode

TPC Transfer State (BS1)

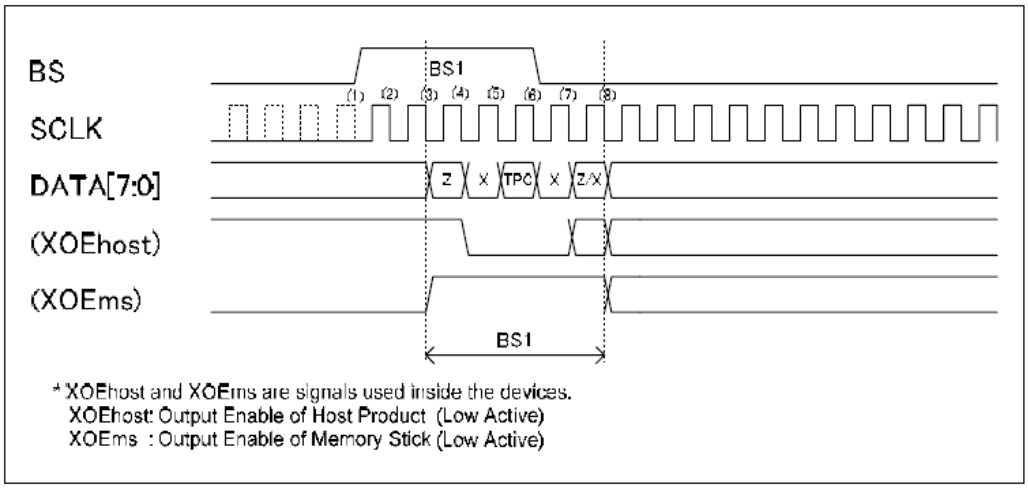


Figure 4.17 8-bit Parallel Protocol TPC Transfer State

Operation	Description
Host	A host controller outputs the BS as High at (1) above, and starts to supply the SCLK before (2). It outputs a TPC on the DATA[7:0] at (5), switches the BS to Low at (6), and starts BS2 from (8).
Memory Stick	Memory Stick detects that the BS is High at (2) and receives the TPC at (6). It detects that the BS is Low at (7), and starts BS2 from (8).

Data Transfer State (Read Packet : BS3)

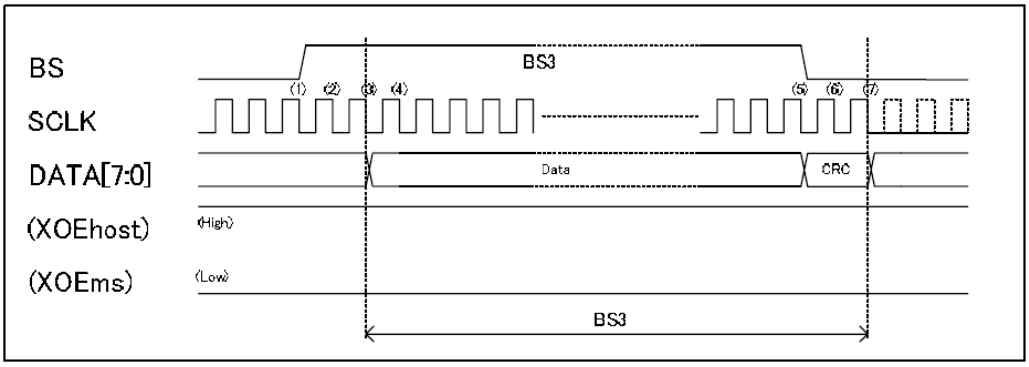


Figure 4.18 8-bit Parallel Protocol Data Transfer State (BS3)

Operation	Description
Host	In a read packet, after switching the BS to High at (1) above, the host controller starts to receive data at (4) which is 3 SCLKs behind from (1). After switching the BS to Low at (5), the host controller receives the lower 8 bits out of a 16-bit CRC at (7) so that the read packet will end.
Memory Stick	Memory Stick detects the BS as High at (2), and starts to output data from (3). It detects the BS as Low at (6) and shifts the state to BS0 at (7).

Data Transfer State (Write Packet : BS2)

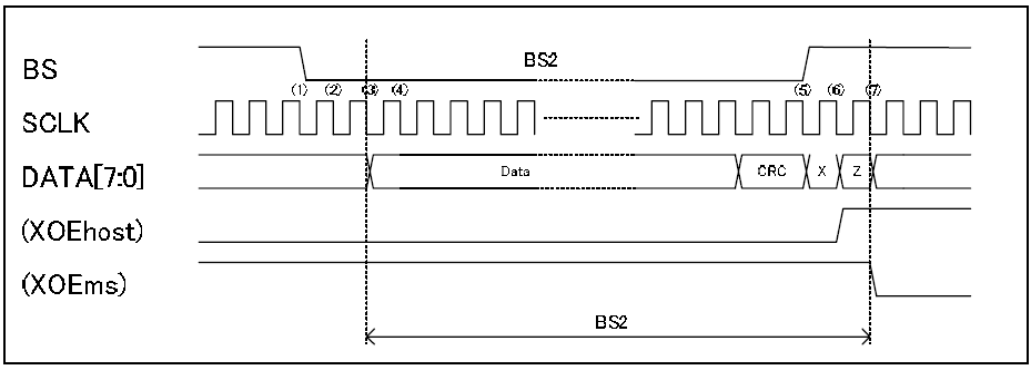


Figure 4.19 8-bit Parallel Protocol Data Transfer State (BS2)

Operation	Description
Host	In a write packet, after switching the BS to Low at (1) above, the host controller starts to transfer data at (3) which is 2 SCLKs behind from (1). The host controller transfers the 16-bit CRC for all of the data from the next SCLK after the last data and switches the BS to High at (5)
Memory Stick	The Memory Stick detects the BS as Low at (2), and starts to receive data at (4). After receiving the lower 8 bits out of a 16-bit CRC at (5), the Memory Stick detects the BS as High at (6), and starts to output s BSY signal or a RDY signal in BS3 (a handshake state) from (7).

Handshake State (Read Packet : BS2 / Write Packet : BS3)

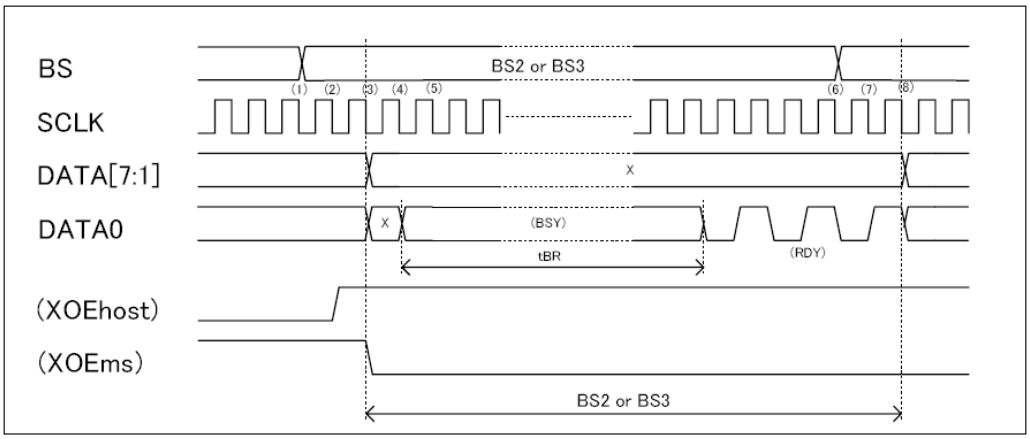


Figure 4.20 8-bit Parallel Protocol Handshake State

Operation	Description
Host	After switching the BS at (1) above, the host controller starts to detect a RDY signal from (5). After receiving the RDY signal the host controller switches the BS signal at a falling edge of the SCLK after (6), and starts the next bus state from (8).
Memory Stick	At (2), Memory Stick detects that the BS has been switched, and then it starts to output a BSY signal or a RDY signal from (3). The Memory Stick outputs a RDY signal after its internal process has completed. It then detects that the BS has been switched again at (7), and starts the next bus state from (8).

INT Transfer State (BS0)

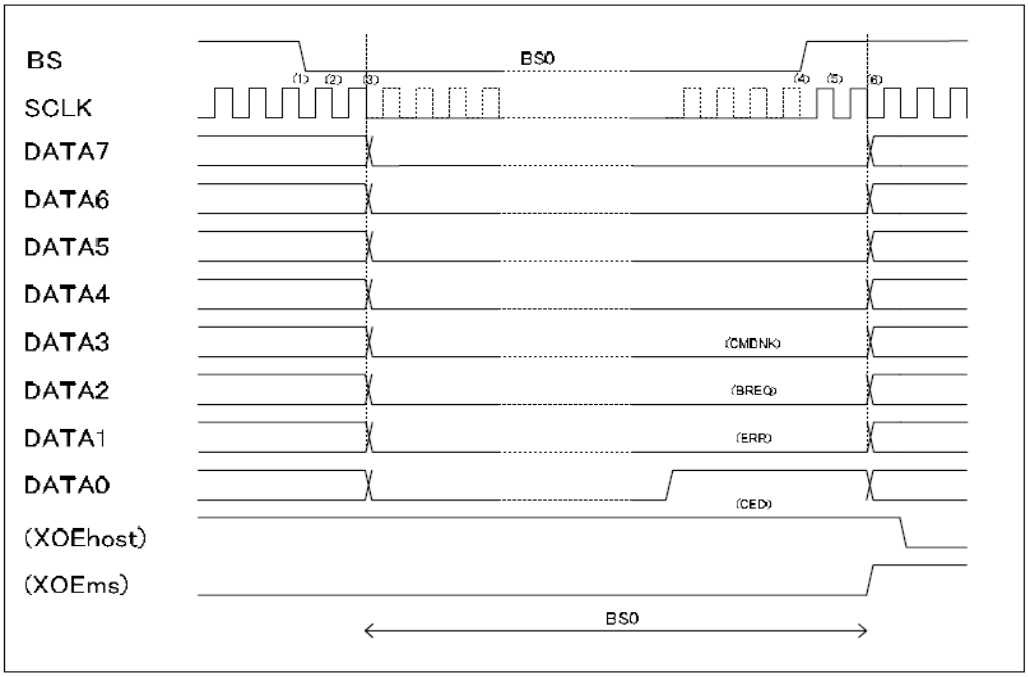


Figure 4.21 8-bit Parallel Protocol INT Transfer State

Operation	Description
Host	A host controller can stop the SCLK after (3). If any of the DATA[3:0] becomes High in this bus state, data is detected as an INT signal. The host controller identifies the interrupt factor through the signal line with High.
Memory Stick	Memory Stick detects BS0 at (2) above, and starts an INT transfer state from (3). After (3), the Memory Stick sets the output mode on the DATA[7:0], and outputs Low until an INT signal is output. The cause of an interrupt is reflected to the INT register and the INT signal is

	output on DATA[3:0] (High).

4.5 I-CON Block

The main function of the I-CON implemented on the smshc_i is to control the smshc on behalf of a host CPU. This chapter explains about the control of the smshc, self-run, microcode instructions, and data transfer.

4.5.1 Control of smshc

The I-CON performs a self-run. As the “Terminology” page says at the very beginning of this document that “self-run” is a state in which the I-CON performs microcode and controls the smshc, its flow can be detailed as follows:

(54) A host CPU writes microcode to [Instruction Queue].
The host CPU sets 1 to the START bit of [Control Register].
(55) The I-CON will be in a self-run mode. It reads, decodes, and executes microcode.
(fetch → decode → execute)

The above tasks performed by a self-run include a control of the smshc.

4.5.2 Self-run

Starting of Self-run

The I-CON starts instructions specified by microcode by writing 1 to the START bit of [Control Register] through a host CPU. The address where the instruction starts from can be specified by the PRGC bits of [Program Counter Register]. To communicate with memory stick using a self-run, the INTEN bit of [MSHC-System Register] shall remain 1.
The following operations are prohibited during a self-run.

- Writing data to an address which does not belong to any of Group A to E listed below
- Changing a bit which belongs to neither Group A nor B
- Reading neither Group F nor G

Group A to E represent as below.

A : START and INTC bits of [Control Register]
B : SRST bit of [System Register]
C : [Program Counter Register]
D : [General Data FIFO]
E : [PageBuffer]
F : [Instruction Queue]
G : [MSHC Register]

Halt of Self-run

During a self-run, if a halt instruction is executed, or such a bit as MIEF, FLG, or ITOF of [Flag Register] changes from 0 to 1 due to a communication problem, the self-run automatically stops and an interrupt occurs. After the halt of the self-run, the START bit of [Control Register] returns to 0.

Termination of Self-run

If 0 is written to the START bit of [Control Register] while it is 1, a self-run of the I-CON can be terminated on purpose as soon as the instruction underway at the moment is completed. When the self-run is terminated, [Program Counter Register] also stops and indicates the address of an instruction next to the one that has been just completed. After the ongoing instruction is completed, the STPF bit of [Flag Register] becomes 1 and an interrupt occurs due to the aborted self-run.

- Even though 1 is written back to the START bit after the termination process has been completed, there is no guarantee that it can lead a resumption of the self-run.
- It requires an asynchronous reset or a synchronous reset to start it over.

4.5.3 Instruction

Overview

Every instruction is 16-bit variable. The value shown in parentheses attached to each instruction represents its length. (1 word is equivalent to 16 bits.)
The structure of each instruction code is as follows.

3 bits (Instruction ID)			3bits (Instruction word length -1)				10 bits (varies depending on each instruction)				
MSB											LSB

Table 4.12 Instruction ID

Instruction ID	Instruction	Description
000	Wait / Halt	A simple wait for a specified period To wait for an INT from memory stick To halt
001	TPC	To send an specified TPC to communicate with memory stick
010	Load	To load an immediate value or the value of an [MSHC Register] onto [General Register]
011	Store	To store the value of [General Register] or an immediate value into [MSHC Registers] or [PageBuffer]
100	Compare	To compare between immediate value and that of [General Register] or [MSHC Registers]
101	Jump	True/false/unconditional jump Where to jump is specified by an absolute address/relative address
110	-	-
111	Nop	No Operation

Wait/Halt Instructions (1 Word)

This type of instructions waits for an INT from memory stick, conducts a wait for a predetermined period, and halts an instruction (Halt). For setting up TimeCount, refer to [Timer Register (0x1C)].

Instruction Code	Note	Description
000_000_11_xxxx_tttt	t:TimeCount	To wait for an INT from memory stick (with a timeout) In case of a timeout, 1 is set to the ITOF bit of [Flag Register] and the instruction is terminated.
000_000_01_xxxx_xxxx		To wait for an INT from memory stick (without a timeout)
000_000_10_xxxx_tttt	t:TimeCount	A simple wait for a specified period
000_000_00_xxxx_CCCC	C:ExitCode	To halt an instruction Consequently, ExitCode will be set to the EXTS bit of [Flag Register].

TPC Instructions (2 to 8 Words)

A TPC instruction issues a TPC to the smshc. There are seven types of TPC instruction.

(56) Instruction 1 to send a TPC through [MSHC-Data Register]
→ Data to be sent is recorded in microcode
(57) Instruction to receive a TPC through [MSHC-Data Register]
→ Data to be received is read by a Load instruction through [MSHC-Data Register]
(58) Instruction to send/receive a TPC through [General Data FIFO]
→ A data transfer request occurs during a self-run
(59) Instruction 1 to send/receive a TPC through [PageBuffer]
→ A data transfer request occurs during a self-run
(60) Instruction 2 to send a TPC through [MSHC-Data Register]
→ Data to be sent is 5-byte data including the value of [General Register]
(61) Instruction 3 to send a TPC through [MSHC-Data Register]
→ Data to be sent is 9-byte data including the value of [General Register]
(62) Instruction 2 to send/receive through [PageBuffer]
→ A data transfer request occurs during a self-run

A TPC timeout can be determined by the lower 4 bits of an instruction code (tttt). However, if tttt is 0000b, there should not be any TPC timeout. For how to specify TimeCount, refer to [Timer Register (0x1C)]. How to handle a TPC communication error can be specified by the E bit of an instruction code.

E=0: ignore communication errors
E=1: In case of an error occurrence, a self-run is terminated

No.	Instruction Code	Note	Description
(1)	001_LLL_000_xE_xttttt	L:Length t:TimeCount T:TPC S:DataSize	To send an immediate value to memory stick through [MSHC-Data Register] The maximum size is 12 bytes for the data to be sent as an immediate value.
(2)	001_001_001_xE_xtttt TTTT_0SSS_SSSS_SSSS	t: TimeCount T: TPC S:DataSize [Byte]	To wait for an INT from memory stick (without a timeout)
(3)	001_001_110_xE_xtttt TTTT_1SSS_SSSS_SSSS	t: TimeCount T: TPC S: DataSize [Byte]	To send/receive data to/from memory stick through [General Data FIFO] For the data size to be sent/received, refer to [General Data FIFO (0x24)] .
(4)	001_001_111_xE_xtttt TTTT_1SSS_SSSS_SSSS	t: TimeCount T: TPC S: DataSize [Byte]	To send/receive data to/from memory stick through [PageBuffer] Data size to be sent: 512 bytes, or 2 Kbytes Data size to be received: 512 bytes, 2 Kbytes, or any integer multiple of 4 bytes
(5)	001_001_010_xE_xtttt TTTT_0SSS_SSSS_SSSS	t: TimeCount T: TPC S: DataSize [Byte]	Block Address send mode To send data to memory stick through [MSHC-Data Register] To send the total 5-byte data of {0x00,[GeneralRegister2,3]}. This instruction code is used to make a Logical/physical transportation table.
(6)	001_001_011_xE_xtttt TTTT_0SSS_SSSS_SSSS	t: TimeCount T: TPC S: DataSize [Byte]	Expanded Block Address send mode To send data to memory stick through [MSHC-Data Register] To send the total 9-byte data of {0x00,[General Register2,3,4,5]}.
(7)	001_001_100_xE_xtttt TTTT_1SSS_SSSS_SSSS	t: TimeCount T: TPC S: DataSize [Byte]	Expanded TPC instruction To send/receive data to/from memory stick through [PageBuffer] This instruction code is used to receive an INT from memory stick, to check the interrupt state, and to send a TPC. Data size to be sent: 512 bytes, or 2 Kbytes Data size to be received: 512 bytes or 2 Kbytes

During the process of an expanded TPC instruction, a self-run is terminated regardless of the E bit when a communication error occurs through GET_INT TPC (TOC =1 or CRC = 1 in [MSHC-Status Register]). Regardless of the E bit, a communication error will be reflected to the FLG bit of [Flag Register].

Load Instructions (1 to 2 Words)

A load instruction writes specified data to General Registers. There are two types of load instructions.

(63) Instruction to read a value from one of MSHC Registers (0x30 to 0x4F) and write it to one of [General Register0 to 5]
(64) Instruction to read an immediate value of microcode and writes it to one of [General Register0 to 5]

Only when reading [MSHC-Data Register] (0x34), an access width can be selected from either 4 bytes or 2 bytes.

If 4 bytes is selected, the upper 2 bytes of [MSHC-Data Register] are written to one of [General Register0, 2, 4] while the lower 2 bytes of [MSHC-Data Register] to the corresponding [General Register1, 3, 5].

No.	Instruction Code	Note	Description
(1)	010_000_RR_1AAA_AAAW	R: Register Number, A: I/O address, W: Access Width (Refer to Table 4.16 ,	To write the values of an specified [MSHC Register] to a designated [General Register]

		Table 4.17, Table 4.18.)	
(2)	010_001_RR_0xxx_xxxx (immediate)	R: Register Number (Refer to Table 4.16)	To write immediate values to a designated [General Register]

Store Instructions (1 to 2 Words)

Store instruction writes the values of a General Register or the immediate values of microcode to a specified place. There are three types of store instructions.

(65) To read the values of a General Register, and write them to [PageBuffer]
(66) To read the values of a General Register, and write them to an [MSHC Register]
(67) To read the immediate values of microcode, and write them to [PageBuffer]

Only when writing values to [MSHC-Data Register], the access width can be selected from either 4 bytes or 2 bytes.

If 4 bytes is selected, any value among [General Register0, 2, 4] is written to the upper 2 bytes of [MSHC-Data Register] while the value of the corresponding [General Register1, 3, 5] to the lower 2 bytes of [MSHC-Data Register].

In case of writing values to any other register than [MSHC-Data Register], the access width shall be 2 bytes.
In case of writing values to [PageBuffer], the access width shall be always 4 bytes.
In case of writing values to [PageBuffer], a store instruction is allowed only when the PDIR bit of [Control Register] is 1.

No.	Instruction Code	Note	Description
(1)	011_000_RR_1AAA_AAAW	R: Register Number, A: I/O address, W: Access Width (Refer to Table 4.16 , Table 4.17, Table 4.18.)	To write the values of a [General Register] to [PageBuffer] To write the values of a [General Register] to an specified [MSHC Register]
(2)	011_001_xx_0010_1000 (immediate)		To write immediate values to [PageBuffer]

Table 4.13 Access Flag Set to [General Register4,5]

Instruction Code	Description
011_000_01_1111_1100	To set an access flag to [General Register4, 5]
011_000_00_1111_1100	To cancel an access flag on [General Register4, 5]

Compare Instructions (2 to 3 Words)

This type of instructions compares an expected value and that of a General Register or an MSHC Register, and reflects the result to the FLG bit of [Flag Register]. When comparing with the value of an MSHC Register, the value shall be first loaded to a General Register before the comparison.

If loaded from [MSHC-Data Register], the access width can be selected from either 4 bytes or 2 bytes. However, even when 4 bytes is selected, it is only the upper 2 bytes that are to be compared.

To compare values effectively, each bit can be masked. 1 shall be set to all the bits to be masked. The results from the comparison will be reflected to the FLG bit of [Flag Register]. When true, the bit is set to 1 while it is set to 0 for false.

In addition, a pre-increment/pre-decrement by 1 or a pre-increment/pre-decrement by 4 to the value of a General Register is available to perform a loop process. In the case of a pre-increment/pre-decrement by 1, the register value wraps around when it overflows or underflows. If two General Registers, General Register A and General Register B as shown in the table below, are compared, a pre-increment/pre-decrement by 1 or a pre-increment/pre-decrement by 4 is executed to General Register A.

PPP	Description
000	No action
010	No action
100	+ 1 to the value of a [General Register] before executing microcode
110	- 1 to the value of a [General Register] before executing microcode
001	No action
011	No action

101	+ 4 to the value of a [General Register] before executing microcode
111	- 4 to the value of a [General Register] before executing microcode

Instruction Code	Note	Description
100_001_RR_0PPP_0xxx (Expected values)	R: Register Number (Refer to Table 4.16 , Table 4.17, Table 4.18.)	To compare an immediate value and that of [General Register] (No bit mask exists)
100_001_RR_1AAA_AAAW (Expected values)	R: Register Number A: I/O address W: Access Width (Refer to Table 4.16 , Table 4.17, Table 4.18.)	To compare an immediate value and that of [General Register] after automatically loaded (No bit mask exists)
100_010_RR_0PPP_0xxx (Expected values) (Bit mask)	R: Register Number (Refer to Table 4.16)	To compare an immediate value and that of [General Register] (Bit masks exist)
100_010_RR_1AAA_AAAW (Expected values) (Bit mask)	R: Register Number A: I/O address W: Access Width (Refer to Table 4.16 , Table 4.17, Table 4.18.)	To compare an immediate value and that of [General Register] after automatically loaded (Bit masks exist)
100_000_AA_0PPP_1xBB	A: [General Register-A] B: [General Register-B] (Refer to Table 4.16)	To compare two [General Register] with one another (No bit mask exists) Pre-increment /Pre-decrement are executed to [General Register-A].
100_001_AA_0PPP_1xBB (Bit mask)	A: [General Register-A] B: [General Register-B] (Refer to Table 4.16)	To compare two [General Register] with one another (Bit masks exist) Pre-increment /Pre-decrement are executed to [General Register-A].

Jump Instructions (1 Word)

Whether a true jump, false jump, or unconditional jump is performed depending on the FLG bit value of [Flag Register].
The address for a jump can be selected from either an absolute address or relative address. A relative address is represented with a 2's complement.

R=0: absolute address
R=1: relative address (address represented with a 2's complement)

Instruction Code	Note	Description
101_000_00_RAAA_AAAA	A: Address	True for a jump
101_000_01_RAAA_AAAA	A: Address	False for a jump
101_000_10_RAAA_AAAA	A: Address	Unconditional

The range of an address is between 0 and 127 for an absolute address while between -64 and +63 for a relative address.

Table 4.14 How to Specify Absolute Address

RAAA_AAAA	Instruction Queue Address Pointer (Absolute Address)
0000_0000	0
0000_0001	1
0000_0010	2
....	....
0111_1101	125
0111_1110	126
0111_1111	127

Table 4.15 How to Specify Relative Address

RAAA_AAAA	Instruction Queue Range of Address Pointer (Relative Address)

1011_1111	+63
1011_1110	+62
1011_1101	+61
....	....
1000_0001	+`
1000_0000	0
1111_1111	-1
....	....
1100_0010	-62
1100_0001	-63
1100_0000	-64

Nop Instructions (1 Word)

No operation

Instruction Code	Note	Description
111_000_xx_xxxx_xxxx		No operation

Table 4.16 Setting of [General Register]

Register Number	Register	Note
00	[General Register0]	
01	[General Register1]	
10	[General Register2]	When the access flag to [General Register4, 5] has been set, [General Register4] is accessed. The access flag can be set by a store instruction.
11	[General Register3]	When the access flag to [General Register4, 5] has been set, [General Register5] is accessed. The access flag can be set by a store instruction.

Table 4.17 Setting of I/O Address

Register Number	Register	Note
0x28	[PageBuffer]	[PageBuffer] can be specified only by a store instruction.
0x30	[MSHC-Command Register]	
0x34	[MSHC-Data Register]	
0x38	[MSHC-Status Register]	
0x3C	[MSHC-System Register]	
0x40	[MSHC-User Custom Register]	
0x44	[MSHC_FIFO Control Register]	
0x48	Reserved	An access is prohibited.
0x4C	[MSHC-DMA Control Register]	

Table 4.18 Setting of Access Width

W	Access Width
0	2 bytes
1	4 bytes

Table 4.19 [General Register] Accessed when Access Width is 1

Register Number	[General Register] to be accessed	Note
00	[General Register0,1]	
01	-	Setting is prohibited.
10	[General Register2,3]	When the access flag to [General Register4, 5] has been set, [General Register4,5] is accessed.
11	-	Setting is prohibited.

4.5.4 Data Transfer

Direction of Data Transfer

During a self-run (START=1 of [Control Register]) of the smshc_i, the data transfer direction of [PageBuffer] and [General Data FIFO] cannot be changed.

- The data transfer direction of [General Data FIFO] is specified by the GDIR bit of [Control

Register].
- The data transfer direction of [PageBuffer] is specified by the PDIR bit of [Control Register].

Therefore, during a self-run (START=1);

Changing GDIR is prohibited.
Changing PDIR is prohibited.

Also, while a self-run is halted (START=0);

Changing GDIR is prohibited if any data remains in [General Data FIFO].
Changing PDIR is prohibited if any data remains in [PageBuffer].

Number of Data Transfers

1. Number of DMA transfers of data written to [General Data FIFO]

When data is written to [General Data FIFO] from a host CPU or a DMA controller during a self-run, the number of transfers shall be set to the GFDN bits of [Memory Control Register].

- The transfer unit is 4 bytes.
- Also refer to [General Data FIFO (0x24)].

2. Total number of pages written to [PageBuffer]

When data is written to [PageBuffer] from a host CPU or a DMA controller during a self-run, the total number of pages shall be set to [General Register] specified by GRPN.

- The transfer unit is 512 bytes. However, if the size of data transferred to memory stick is not divisible by 512, it shall be rounded up to a multiple that can be divided by 512.
- Also refer to [PageBuffer (0x28)].

Slice Size

To access [General Data FIFO] and [PageBuffer] from a host CPU or a DMA controller during a self-run, the data size shall be only a multiple of Word （32bit）. Listed below are the slice sizes.

- “Slice Size” is the size of transfer data per XDRQ assertion. It shall be set depending on the specifications of a DMA controller.
- If the slice size of the smshc_i does not agree with that of a DMA controller, a proper operation cannot be achieved.

Table 4.20 DMA Slice Size of [General Data FIFO]

GDSZ[1:0]	Slice Size	Capacity of General Data FIFO			
		16 bytes	32 bytes	64 bytes	128 bytes
00	4 bytes	o	o	o	o
01	16 bytes	x	o	o	o
10	32 bytes	x	x	o	o
11	128 bytes	x	x	x	o

When the data of a fraction smaller than a slice size is read/written from/to [General Data FIFO];

- In case of DMAE0/1=1: The data transfer request is cleared as soon as DMA acknowledgement[Channel0/1] is asserted after the fraction data has been read/written.
- In case of DMAE0/1=0: The data transfer request is cleared after the fraction data has been read/written.

Table 4.21 DMA Slice Size of [PageBuffer]

PDSZ[2:0]	Slice Size
000	4 bytes
001	16 bytes

010	32 bytes
011	64 bytes
100	128 bytes
101	256 bytes
110	Reserved
111	Reserved

Data of a fraction smaller than a slice size:

To write it to [PageBuffer];

- Prohibited. The data size shall be always equal to that of the slice size.

To read it from [PageBuffer];

- In case of DMAE0/1=1: The data transfer request is cleared as soon as DMA acknowledgements[Channel0/1] are asserted after the fraction data has been read.
- In case of DMAE0/1=0: The data transfer request is cleared after the fraction data has been read.

Data Transfer Request of General Data FIFO

1. Reading Data from [General Data FIFO]

When GDIR = 0, data is being read from [General Data FIFO].

- During a self-run:
  When data stored in [General Data FIFO] reaches the volume of the slice size, a data transfer request occurs.
  On the other hand, while data stored in [General Data FIFO] remains smaller than the slice size, no data transfer request occurs.

After a self-run, if any data is left over in [General Data FIFO], a data transfer request occurs.

Once the data transfer has been completed, no more data transfer request occurs.

2. Writing Data to [General Data FIFO]

When GDIR=1, data is being written to [General Data FIFO].

- During a self-run:
  If there remains a space equivalent to the slice size or larger in [General Data FIFO], and also the number of transfers is less than the value set in the GFDN bits, a data transfer request occurs. When the number of transfers specified in the GFDN bits of [Memory Control Register] is completed, no more data transfer request will occur.

- While a self-run is being halted:
  No data transfer request occurs.

Data Transfer Request of PageBuffer

1. Reading Data from [PageBuffer]

When PDIR = 0, data is being read from [PageBuffer].

- During a self-run:
  When data stored in [PageBuffer] reaches the volume of the PBBC size, a data transfer request occurs.

Until data of the PBBC size has been completely read, data transfer requests will occur multiple times depending on a slice size. If the data written in [PageBuffer] is smaller than the PBBC size, no data transfer request will occur.

After a self-run, if any data is left over in [PageBuffer], a data transfer request occurs.

Once the data transfer request is completed, no more data transfer request will occur.

2. Writing Data to [PageBuffer]

When PDIR=1, data is being written to [PageBuffer].

- During a self-run:
  If there remains a space equivalent to the PBBC size or larger in [PageBuffer], and also the number of transferred pages is less than the total number of transfer pages, a data transfer request occurs.

Data transfer requests for writing data of the PBBC size will occur multiple times depending on a slice size.

Once the total transfer page data has been transferred, no more Data transfer request occurs.

A total data volume written to [PageBuffer] shall be a multiple of the PBBC size ×2.

If data to be transferred is not a multiple of the PBBC size ×2, it shall be so adjusted by adding arbitrary data to the end of the data.

Such added data is likely to be left over in [PageBuffer] when a self-run is completed. In this case, an initialization is required through a reset of I-CON or the PBCR bit of [Memory Control Register].

- While a self-run is being halted:
  No Data transfer request occurs.

5 SD/SDIO/MMC/CE-ATA Host Controller

5.1 Overview

The TCC8900 complies with following versions of specification concerned.

- SD Host Controller Specification Version 2.0
- SDIO Card Specification Version 2.0
- SD Memory Card Specification Draft Version 2.0
- SD Memory Card Security Specification Version 1.0
- MMC Specification Version 3.31, 4.2,,4.3 and MMC Plus and MMC Mobile
- CE-ATA Digital Protocol Revision 1.1

There are two host controllers in TCC8900 and each of them has two slot in it. Each slot has a set of registers independently. So those can be assigned to separate host controllers at the same time. For more effective data transaction, two 1K dual port FIFO are used. You may access those using inherent SDMA and ADMA. No DMA access is also available.

Refer to the SD Host Controller Standard Specification V2.00 of SD Association to know more specific explanation.

5.2 Block Diagram of Each Host Controller

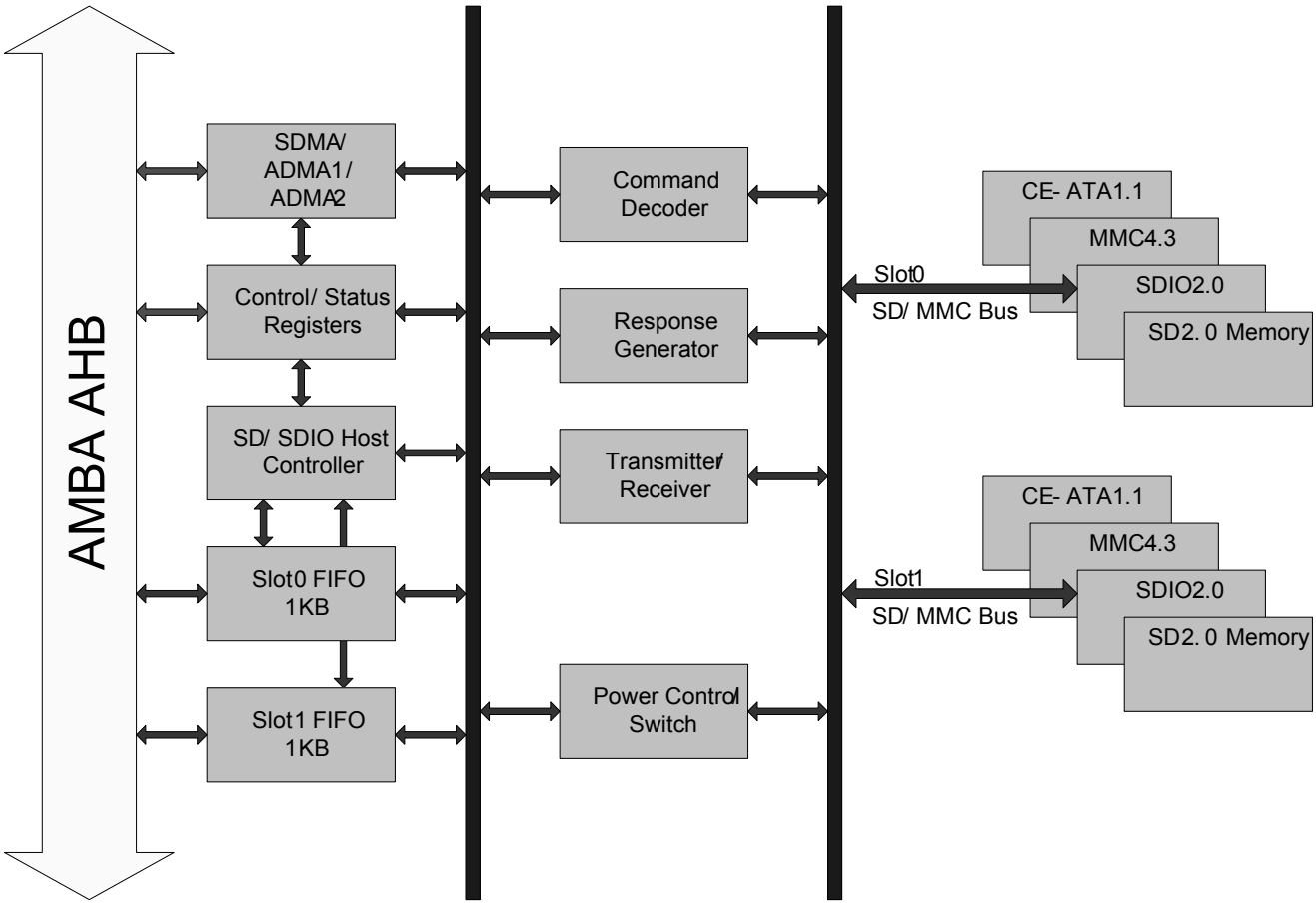


Figure 5.1 SD/SDIO/MMC/CE-ATA Block Diagram

5.3 Register Description

Table 5.1 Register Map (Base Address = 0xF05A0000)

Name	Address	Type	Reset	Description
SDMA	0x000	R/W	0x0000	SDMA System Address
BSIZE	0x004	R/W	0x0000	Block Size
BCNT	0x006	R/W	0x0000	Block Count
ARG	0x008	R/W	0x0000	Argument
TMODE	0x00C	R/W	0x0000	Transfer Mode
CMD	0x00E	R/W	0x0000	Command
RESP0	0x010	R	0x0000	Response0

RESP1	0x012	R	0x0000	Response1
RESP2	0x014	R	0x0000	Response2
RESP3	0x016	R	0x0000	Response3
RESP4	0x018	R	0x0000	Response4
RESP5	0x01A	R	0x0000	Response5
RESP6	0x01C	R	0x0000	Response6
RESP7	0x01E	R	0x0000	Response7
DATAL	0x020	R/W	-	Buffer Data Port(Low)
DATAH	0x022	R/W	-	Buffer Data Port(High)
STATEL	0x024	R	0x0000	Present State(Low)
STATEH	0x026	R	0x0000	Present State(High)
CONTL	0x028	R/W	0x0000	Power Control / Host Control
CONTH	0x02A	R/W	0x0000	Wakeup Control / Block Gap Control
CLK	0x02C	R/W	0x0000	Clock Control
TIME	0x02E	R/W	0x0000	Software Reset / Timeout Control
STSL	0x030	R	0x0000	Normal Interrupt Status(Low)
STSH	0x032	R	0x0000	Normal Interrupt Status(High)
STSENL	0x034	R/W	0x0000	Normal Interrupt Status Enable(Low)
STSENH	0x036	R/W	0x0000	Normal Interrupt Status Enable(High)
INTENL	0x038	R/W	0x0000	Normal Interrupt Signal Enable(Low)
INTENH	0x03A	R/W	0x0000	Normal Interrupt Signal Enable(High)
CMD12ERR	0x03C	R	0x0000	Auto CMD12 Error Status
CAPL	0x040	R	0x30B0	Capabilities(Low)
CAPH	0x042	R	0x69EF	Capabilities(High)
CURL	0x048	R	0x0001	Maximum Current Capabilities(Low)
CURH	0x04A	R	0x0000	Maximum Current Capabilities(High)
FORCEL	0x050	W	0x0000	Force event for AutoCmd12 Error
FORCEH	0x052	W	0x0000	Force event for Error Interrupt Status
ADMAERR	0x054	R/W	0x0000	ADMA Error Status
ADDR0	0x058	R/W	0x0000	ADMA Address[15:0]
ADDR1	0x05A	R/W	0x0000	ADMA Address[31:16]
ADDR2	0x05C	R/W	0x0000	ADMA Address[47:32]
ADDR3	0x05E	R/W	0x0000	ADMA Address[63:48]
SLOT	0x0FC	R	0x0000	Slot Interrupt Status
VERSION	0x0FE	R	0x0002	Host Controller Version

The address map of registers is arranged by the half word(16bits) to specify those functions. But The TCC8900 accesses data by the word(32bits). Those are only for 1 slot. There are 4 groups of registers for each slot. The Table 5.2 shows base address of each slot.

Table 5.2 Base Address of Each Slot

CORE No.	SLOT No.	Base Address
0	0	0xF05A_0000
	1	0xF05A_0100
1	2	0xF05A_0200
	3	0xF05A_0300

Table 5.3 Channel Control Register Map (Base Address = 0xF05A0800)

Name	Address	Type	Reset	Description
SDPORTCTRL	0x00	R/W	0x0000	SD/MMC port control register
SDPORTDLY0	0x04	R/W	0x0000	SD/MMC output delay control register
SDPORTDLY1	0x08	R/W	0x0000	SD/MMC output delay control register
SDPORTDLY2	0x0C	R/W	0x0000	SD/MMC output delay control register
SDPORTDLY3	0x10	R/W	0x0000	SD/MMC output delay control register

The TCC8900 has 8 external ports for SD/MMC interface. Each slot can be connected to the one of these ports.

SDPORTCTRL **0xF05A0800**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CD3	CD2	CD1	CD0	WP3	WP2	WP1	WP0								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	SLOT3				SLOT2				SLOT1				SLOT0		

BIT	NAME	R/W	RESET	DESCRIPTION
31	CD3	R/W	0	Card Detection for SLOT 3 This bit is connected to card detection signal of SLOT3
30	CD2	R/W	0	Card Detection for SLOT 2 This bit is connected to card detection signal of SLOT2
29	CD1	R/W	0	Card Detection for SLOT 1 This bit is connected to card detection signal of SLOT1
28	CD0	R/W	0	Card Detection for SLOT 0 This bit is connected to card detection signal of SLOT0
27	WP3	R/W	0	Write Protect for SLOT 3 This bit is connected to write protect signal of SLOT3
26	WP2	R/W	0	Write Protect for SLOT 2 This bit is connected to write protect signal of SLOT2
25	WP1	R/W	0	Write Protect for SLOT 1 This bit is connected to write protect signal of SLOT1
24	WP0	R/W	0	Write Protect for SLOT 0 This bit is connected to write protect signal of SLOT0
14-12	SLOT3	R/W	0	These bits specify what ports are used for SLOT3. 0-7: Port Number
10-8	SLOT2	R/W	0	These bits specify what ports are used for SLOT2. 0-7: Port Number
6-4	SLOT1	R/W	0	These bits specify what ports are used for SLOT1. 0-7: Port Number
2-0	SLOT0	R/W	0	These bits specify what ports are used for SLOT0. 0-7: Port Number
Others	-	R	0	Read as 0

SDPORTDLY0 **0xF05A0804**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-		OEN			CMD			DATA7			DATA6			DATA5	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA5	DATA4			DATA3			DATA2			DATA1			DATA0		

SDPORTDLY1 **0xF05A0808**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-		OEN			CMD			DATA7			DATA6			DATA5	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA5	DATA4			DATA3			DATA2			DATA1			DATA0		

SDPORTDLY2 **0xF05A080C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-		OEN			CMD			DATA7			DATA6			DATA5	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA5	DATA4			DATA3			DATA2			DATA1			DATA0		

SDPORTDLY3 **0xF05A0810**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-		OEN			CMD			DATA7			DATA6			DATA5	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA5	DATA4			DATA3			DATA2			DATA1			DATA0		

DATA0-7	[23:0]	DATA[7:0] Output Delay Configuration
	It controls output delays of each DATA signal. 3 bits are assigned and it ranges from 0 to 7. 0: no delay 1~7: about max. 500ps* DATAn	

CMD	[26:24]	Command Output Delay Configuration
	It controls output delays of Command signal. 3 bits are assigned and it ranges from 0 to 7. 0: no delay 1~7: about max. 500ps* CMD	

OEN	[23:0]	OEN Output Delay Configuration
	It controls output delays of Output Enable signal. 3 bits are assigned and it ranges from 0 to 7. 0: no delay 1~7: about max. 500ps* OEN	

SDMA System Address 0xF05A0*n*[1]00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDMA[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SDMA[15:0]															

SDMA	[31:0]	SDMA System Address
N	System memory address for a DMA transfer This contains system memory address for a SDMA transfer. This should be initialized before starting a SDMA transaction. After SDMA has stopped, the next system address of the next contiguous data position can be read from this register. The SDMA transfer waits at the every boundary specified by the Host SDMA Buffer Boundary in the Block Size register.	

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

Block Count and Size

0xF05A0*n*[1]04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCNT[15:0]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BSIZE[12]	SDMABUF[2:0]			BSIZE[11:0]											

BCNT	[31:16]	Block Count for Current Transfer
N	This is enabled when Block Count Enable in the Transfer Mode register is set to 1 and is valid only for multiple block transfers. 0000h - Stop Count 0001h - 1 block 0002h - 2 blocks --- --- FFFFh - 65535 blocks		

BSIZE	[15], [11:0]	Transfer Block Size
N	The block size for block data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. Values ranging from 1 up to the maximum buffer size can be set 0000h - No Data Transfer 0001h - 1 Byte 0002h - 2 Bytes 0003h - 3 Bytes 0004h - 4 Bytes --- --- 01FFh - 511 Bytes 0200h - 512 Bytes --- --- 0800h - 2048 Bytes 1000h - 4096 Bytes		

SDMABUF	[14:12]	Host SDMA Buffer Size
N	These bits specify the size of contiguous buffer in the system memory. The SDMA transfer shall wait at the every boundary specified by these fields and the Host Controller generates the DMA Interrupt to request the Host Driver to update the SDMA System Address register. At the end of transfer, the Host Controller may issue or may not issue DMA Interrupt. In particular, DMA Interrupt shall not be issued after Transfer Complete Interrupt is issued. 000b - 4KB(Detects A11 Carry out) 001b - 8KB(Detects A12 Carry out) 010b - 16KB(Detects A13 Carry out) 011b - 32KB(Detects A14 Carry out) 100b - 64KB(Detects A15 Carry out) 101b -128KB(Detects A16 Carry out) 110b - 256KB(Detects A17 Carry out) 111b - 512KB(Detects A18 Carry out)		

[1] n = 0 or 1 for two slots in core0, 2 or 3 for two slots in core1.

Argument 0xF05A0*n*[1]08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARG[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ARG[15:0]															

ARG	[31:0]	Command Argument
N	This is specified as bit39-8 of Command-Format. See the Table 5.4	

Table 5.4 Command Format

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	'1'
Description	start bit	transmission bit	response index	Card status	CRC7	end bit

Command and Transfer Mode 0xF05A0*n*[1]0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		CMDINDEX						CTYPE		DATSEL	CICHK	CRCHK		RTYPE	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								SPI	ATACMD	MS	DIR		ACMD12	BCNTEN	DMAEN

BIT	NAME	R/W	RESET	DESCRIPTION
29-24	CMDINDEX	R/W	0	Command Index This bit shall be set to the command number (CMD0-63, ACMD0-63).
23-22	CTYPE	R/W	0	Command Type There are three types of special commands. - Abort(11b) CMD12, CMD52 for writing 'I/O Abort' in CCCR - Resume(10b) CMD52 for writing "Function Select" in CCCR - Suspend(00b) CMD52 for writing "Bus Suspend" in CCCR - Normal(00b) all other commands
21	DATSEL	R/W	0	Data Present Select 1: This bit is set to 1 to indicate that data is present and shall be transferred using the DAT line. 0: No Data Present
20	CICHK	R/W	0	Command Index Check Enable 1: If this bit is set to 1, the HC shall check the index field in the response to see if it has the same value as the command index. 0: Disable
19	CRCHK	R/W	0	Command CRC Check Enable 1: If this bit is set to 1, the HC shall check the CRC field in the response. 0: Disable

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

17:16	RTYPE	R/W	0	Response Type Response Type Select 00 - No Response 01 - Response length 136 (R2) 10 - Response length 48 (R3, R4, R1, R5, R7) 11 - Response length 48 check Busy after response (R1b, R5b)
7	SPI	R/W	0	SPI Mode 1: SPI Mode 0: SD Mode
6	ATACMD	R/W	0	CMD Completion Enable for CE-ATA Device will send command completion signal for CE-ATA Device will not send command completion signal for CE-ATA
5	MS	R/W	0	Multi/Single Block Select This bit is set when issuing multiple-block transfer commands using DAT line. For any other commands, this bit shall be set to 0. 1: Multiple Block 0: Single Block
4	DIR	R/W	0	Data Transfer Direction Select 1: Read (Card to Host) 0: Write (Host to Card)
2	m	R/W	0	Auto CMD12 Enable 1: When this bit is set to 1, the HC shall issue CMD12 automatically when last block transfer is completed. 0: Disable
1	BCNTEN	R/W	0	Block Count Enable 1: enable the Block count register, which is only relevant for multiple block transfers. 0: Disable
0	DMAEN	R/W	0	DMA Enable DMA can be enabled only if DMA Support bit in the Capabilities register is set. 1: A DMA operation shall begin when the HD writes to the upper byte of Command register (0x00F). 0: No data transfer or Non-DMA data transfer

Response1/0 **0xF05A0*n*[1]10**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESPONSE1[15:0]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESPONSE0[15:0]															

Response3/2 **0xF05A0*n*[1]14**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESPONSE3[15:0]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESPONSE2[15:0]															

Response5/4 **0xF05A0*n*[1]18**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

RESPONSE5[15:0]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESPONSE4[15:0]															

Response7/6 **0xF05A0*n*[1]1C**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESPONSE7[15:0]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESPONSE6[15:0]															

RESPONSE	**[127:0]**	**Command Response**
N	command responses from the SD Bus	

Table 5.5 Response Register

Kind of Response	Meaning of Response	Response Field	Response Register
R1, R1b(normal response)	Card Status	R[39:8]	RESPONSE[31:0]
R1b(Auto CMD12 response)	Card Status for Auto CMD12	R[39:8]	RESPONSE[127:96]
R2(CID, CSD register)	CID or CSD reg. incl.	R[127:8]	RESPONSE[119:0]
R3(OCR register)	OCR register for memory	R[39:8]	RESPONSE[31:0]
R4(OCR register)	OCR register for I/O etc	R[39:8]	RESPONSE[31:0]
R5, R5b	SDIO response	R[39:8]	RESPONSE[31:0]
R6(Published RCA response)	New published RCA[31:16] etc	R[39:8]	RESPONSE[31:0]

Buffer Data Port

0xF05A0*n*[1]20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA[15:0]															

DATA	[31:0]	Buffer Data
1	The Host Controller Buffer can be accessed through this 32-bit Data Port Register.	

Present State

0xF05A*n*[1]24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			DAT[7:4]				CMD	DAT[3:0]				SDWP	SDCD	CDST	CDIN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				RDEN	WREN	RDACT	WRACT	Reserved					DATACT	NODAT	NOCMD

DAT	[28:25], [23:20]	Data Line Signal Level
	This status is used to check DAT[7:0] line level to recover from errors, and for debugging.	

CMD	[24]	CMD Line Signal Level
	This status is used to check CMD line level to recover from errors, and for debugging.	

SDWP	[19]	Write Protect Switch Pin Level
1	This bit reflects the SDWP# pin. Write enabled	
0	Write protected	

SDCD	[18]	Card Detect Pin Level
1	This bit reflects the inverse value of the SDCD# pin. Card present.	
0	No Card present.	

CDST	[17]	Card State Stable
1	If this bit is set to 1, it means the Card Detect Pin Level is stable. No Card or Inserted.	
0	Reset of Debouncing.	

CDIN	[16]	Card Inserted
1	This bit indicates whether a card has been inserted.	
0	Reset or Debouncing or No Card.	

RDEN	[11]	Buffer Read Enable
1	This status is used for non-DMA read transfers. If this bit is 1, readable data exists in the buffer.	
0	Read Disable	

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

WREN	[10]	Buffer Write Enable
1	This status is used for non-DMA read transfers. If this bit is 1, data can be written to the buffer.	
0	Write Disable	

RDACT	[9]	Read Transfer Active
1	This status is used for detecting completion of a read transfer.	
0	No valid data	

WRACT	[8]	Write Transfer Active
1	This status indicates a write transfer is active.	
0	No valid data	

DATACT	[2]	DAT Line Active
1	This bit indicates whether one of the DAT line on SD bus is in use.	
0	DAT line inactive.	

NODAT	[1]	Command Inhibit(DAT)
1	This status bit is generated if either the DAT Line Active or the Read transfer Active is set to 1. It cannot issue command which uses the DAT line	
0	It can issue command which uses the DAT line	

NOCMD	[0]	Command Inhibit(CMD)
1	indicates the CMD line is in use.	
0	indicates the CMD line is not in use and the HC can issue a SD command using the CMD line.	

Wakeup/Block Gap/Power/Host Control 0xF05A0*n*[1]28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					WKOUT	WKIN	WKINT					BGINT	RDWAIT	CONREQ	BGSTOP
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				VOLTSEL			POW	DETSEL	DETCD	SD8	SELDMA		HS	SD4	LED

WKOUT	[26]	Wakeup Enable On Card Removal
1	This bit enables wakeup event via Card Removal assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit.	
0	Disable	

WKIN	[25]	Wakeup Enable On Card Insertion
1	This bit enables wakeup event via Card Insertion assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit.	
0	Disable	

WKINT	[24]	Wakeup Enable On Card Interrupt
1	This bit enables wakeup event via Card Interrupt assertion in the Normal Interrupt Status register. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1	
0	Disable	

BGINT	[19]	Interrupt At Block Gap
1	This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer.	
0	Disable	

RDWAIT	[18]	Read Wait Control
1	If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using DAT[2] line.	
0	Disable	

CONREQ	[17]	Continue Request
1	This bit is used to restart a transaction which was stopped using the Stop At Block Gap Request.	
0	Stop is ignored	

BGSTOP	[16]	Stop At Block Gap Request
1	This bit is used to stop executing a transaction at the next block gap for non-DMA,SDMA and ADMA transfers. The Host Driver shall leave this bit set to 1 until the Transfer Complete is set to 1.	
0	Keep transferring	

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

VOLTSEL	[11:9]	SD Bus Voltage Select
N	By setting these bits, the HD selects the voltage level for the SD card. Only 3.3V is supported. *Others will be supported in the future.* 111b - 3.3 V(TYP.) 110b - 3.0 V(TYP.) 101b - 1.8 V(TYP.) 100b - 000b – Reserved	

POW	[8]	SD Bus Power
1	Before setting this bit, the SD host driver shall set SD Bus Voltage Select. *This will be supported in the future.*	

DETSEL	[7]	Card Detect Signal Selection
1	The card detect test level is selected for test purpose.	
0	SDCD# is selected for normal use	

DETCD	[6]	Card Detect Test Level
1	This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted,	
0	No Card	

SD8	[5]	SD 8bit Mode
1	8 bit mode is selected	
0	8 bit mode is not selected	

SELDMA	[4:3]	DMA Select
N	One of supported DMA modes can be selected. Use of selected DMA is determined by DMA Enable of the Transfer Mode register. 00 - SDMA is selected 01 - 32-bit Address ADMA1 is selected 10 - 32-bit Address ADMA2 is selected 11 - 64-bit Address ADMA2 is selected	

HS	[2]	High Speed Enable
1	If this bit is set to 1, the HC outputs CMD line and DAT lines at the rising edge of the SD clock (up to 50 MHz)	
0	The Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock (up to 25MHz).	

SD4	[1]	Data Transfer Width
1	4 bit mode	
0	1 bit mode	

LED	[0]	LED Control
-	This bit is used to caution the user not to remove the card while the SD card is being accessed. *This will be supported in the future*.	

Software Reset/Timeout/Clock Control **0xF05A0*n*[1]2C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					RSTDAT	RSTCMD	RSTALL	Reserved				TIMEOUT[3:0]			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SDCLKSEL[7:0]								Reserved					SCKEN	CLKRDY	CLKEN

RSTDAT	[26]	Software Reset for DAT Line
1	Only part of data circuit is reset. The following registers and bits are cleared by this bit: Buffer Data Port Register - Buffer is cleared and Initialized. Present State register - Buffer read Enable - Buffer write Enable - Read Transfer Active - Write Transfer Active - DAT Line Active - Command Inhibit (DAT) Block Gap Control register - Continue Request - Stop At Block Gap Request Normal Interrupt Status register - Buffer Read Ready - Buffer Write Ready - Block Gap Event - Transfer Complete	
0	Keep working	

RSTCMD	[25]	Software Reset For CMD Line
1	Only part of command circuit is reset. The following registers and bits are cleared by this bit: Present State register - Command Inhibit (CMD) Normal Interrupt Status register - Command Complete	
0	Keep working	

RSTALL	[24]	Software Reset For All
1	This reset affects the entire HC except for the card detection circuit. If this bit is set to 1, the host driver should issue reset command and reinitialize the SD card.	
0	Keep working	

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

TIMEOUT	[19:16]	Data Timeout Counter Value
N	This value determines the interval by which DAT line time-outs are detected 1111 - Reserved 1110 - TMCLK * 2^27 ---------- ---------- 0001 - TMCLK * 2^14 0000 - TMCLK * 2^13	

SDCLKSEL	[15:8]	SDCLK Frequency Select
N	This register is used to select the frequency of the SDCLK pin. 80h - base clock divided by 256 40h - base clock divided by 128 20h - base clock divided by 64 10h - base clock divided by 32 08h - base clock divided by 16 04h - base clock divided by 8 02h - base clock divided by 4 01h - base clock divided by 2 00h - base clock(10MHz-63MHz) According to the Physical Layer Specification, the maximum SD Clock frequency is 25 MHz in normal speed mode and 50MHz in high speed mode, and shall never exceed this limit.	

SCKEN	[2]	SD Clock Enable
1	SD Clock is enabled.	
0	The Host Controller shall stop SDCLK when writing this bit to 0. SDCLK Frequency Select can be changed when this bit is 0.	

CLKRDY	[1]	Internal Clock Stable
1	This bit is set to 1 when SD clock is stable after writing to Internal Clock Enable in this register to 1. The SD Host Driver shall wait to set SD Clock Enable until this bit is set to 1.	
0	Not ready	

CLKEN	[0]	Internal Clock Enable
1	Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the Host Controller shall set Internal Clock Stable in this register to 1. This bit shall not affect card detection.	
0	Stop	

Normal Interrupt Status

0xF05A0*n*[1]30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VENDOR[3:0]				Reserved		ADMA	ACMD12	CLIMIT	DATEND	DATCRC	DATTIME	CINDEX	CMDEND	CMDCRC	CMDTIME
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERR	Reserved						CDINT	CDOUT	CDIN	RDRDY	WRRDY	DMA	BLKGAP	TDONE	CDONE

VENDOR	[31:28]	Vendor Specific Error Status
N	Additional status bits can be defined in this register by the vendor.	

ADMA	[25]	ADMA Error
1	This bit is set when the Host Controller detects errors during ADMA based data transfer. In addition, the Host Controller generates this Interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor.	
0	No Error	

ACMD12	[24]	Auto CMD12 Error
1	This bit is set when detecting that one of the bits in Auto CMD12 Error Status register has changed from 0 to 1. This bit is set to 1, not only when the errors in Auto CMD12 occur but also when Auto CMD12 is not executed due to the previous command error..	
0	No Error	

CLIMIT	[23]	Current Limit Error
1	Reading 1 means the HC is not supplying power to SD card due to some failure. *This will be supported in the future*.	
0	No Error	

DATEND	[22]	Data End Bit Error
1	Occurs when detecting 0 at the end bit position of read data which uses the DAT line or the end bit position of the CRC status.	
0	No Error	

DATCRC	[21]	Data CRC Error
1	Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC Status having a value of other than “010”.	
0	No Error	

DATTIME	[20]	Data Timeout Error
1	Occurs when detecting one of following timeout conditions. 1) Busy Timeout for R1b, R5b type. 2) Busy Timeout after Write CRC status 3) Write CRC status Timeout 4) Read Data Timeout	
0	No Error	

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

CINDEX	[19]	Command Index Error
1	Occurs if a Command Index error occurs in the Command Response.	
0	No Error	

CMDEND	[18]	Command End Bit Error
1	Occurs when detecting that the end bit of a command response is 0.	
0	No Error	

CMDCRC	[17]	Command CRC Error
1	Command CRC Error is generated in two cases. 1) If a response is returned and the Command Time-out Error is set to 0, this bit is set to 1 when detecting a CRC error in the command response 2) The HC detects a CMD line conflict by monitoring the CMD line when a command is issued.	
0	No Error	

CMDTIME	[16]	Command Timeout Error
1	This bit is set only if the no response is returned within 64 SDCLK cycles from the end bit of the command. If the Host Controller detects a CMD line conflict, this bit shall be set without waiting for 64 SD clock cycles because the command will be aborted by the Host Controller.	
0	No Error	

ERR	[15]	Error Interrupt
1	If any of the bits in the Error Interrupt Status Register are set, then this bit is set. Therefore the HD can test for an error by checking this bit first.	
0	No Error	

CDINT	[8]	Card Interrupt
1	Card interrupt is generated. In 1-bit mode, the Host Controller shall detect the Card Interrupt without SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the Host System. It is necessary to define how to handle this delay.	
0	No card interrupt	

CDOUT	[7]	Card Removal
1	This status is set if the Card Inserted in the Present State register changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed	
0	Card State Stable or Debouncing	

CDIN	[6]	Card Insertion
1	This status is set if the Card Inserted in the Present State register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed.	
0	Card State Stable or Debouncing	

RDRDY	[5]	Buffer Read Ready
1	This status is set if the Buffer Read Enable changes from 0 to 1 and it is ready to read buffer.	
0	Not ready to read buffer.	

WRRDY	[4]	Buffer Write Ready
1	This status is set if the Buffer Write Enable changes from 0 to 1 and it is ready to read buffer.	
0	Not ready to read buffer.	

DMA	[3]	DMA Interrupt
1	This status is set if the Host Controller detects the Host DMA Buffer boundary during transfer. In case of ADMA, by setting Int field in the descriptor table, Host Controller generates this interrupt. This interrupt shall not be generated after the Transfer Complete.	
0	No DMA Interrupt	

BLKGAP	[2]	Block Gap Event
1	If the Stop At Block Gap Request in the Block Gap Control register is set, this bit is set when both a read / write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this bit is not set to 1. - Read Transaction: This bit is set at the falling edge of the DAT Line Active Status (When the transaction is stopped at SD Bus timing. - Write Transaction: This bit is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing).	
0	No Block Gap Event	

TDONE	[1]	Transfer Complete
1	This bit is set when a read / write transaction is completed. - Read Transaction: This bit is set at the falling edge of Read Transfer Active Status. This interrupt is generated in two cases. The first is when a data transfer is completed as specified by data length (After the last data has been read to the Host System). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request in the Block Gap Control register(After valid data has been read to the Host System). - Write Transaction: This bit is set at the falling edge of the DAT Line Active Status. This interrupt is generated in two cases. The first is when the last data is written to the SD card as specified by data length and the busy signal released. The second is when data transfers are stopped at the block gap by setting Stop At Block Gap Request in the Block Gap Control register and data transfers completed. (After valid data is written to the SD card and the busy signal released). - Write Transaction: This bit is set when busy is de-asserted.	
0	Not complete	

Table 5.6 Relation between Transfer Complete and Data Timeout Error

Transfer Complete	Data Timeout Error	Meaning of the status
0	0	Interrupted by another factor
0	1	Timeout occur during transfer
1	Don't Care	Command Execution complete

CDONE	[0]	Command Complete
1	This bit is set when get the end bit of the command response (Except Auto CMD12).	
0	No Command Complete	

Table 5.7 Relation between Command Complete and Command Timeout Error

Command Complete	Command Timeout Error	Meaning of the status
0	0	Interrupted by another factor
Don't Care	1	Response not received within 64 SDCLK cycles
1	0	Response received

Normal Interrupt Status Enable **0xF05A0*n*[1]34**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VENDOR						ADMA	ACMD12	CLIMIT	DATEND	DATCRC	DATTIME	CINDEX	CMDEND	CMDCRC	CMDTIME
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							CDINT	CDOUT	CDIN	RDRDY	WRRDY	DMA	BLKGAP	TDONE	CDONE

VENDOR	[31:28]	Vendor Specific Error Status Enable
1	Enabled	
0	Masked	

ADMA	[25]	ADMA Error Status Enable
1	Enabled	
0	Masked	

ACMD12	[24]	Auto CMD12 Error Status Enable
1	Enabled	
0	Masked	

CLIMIT	[23]	Current Limit Error Status Enable
1	Enabled	
0	Masked	

DATEND	[22]	Data End Bit Error Status Enable
1	Enabled	
0	Masked	

DATCRC	[21]	Data CRC Error Status Enable
1	Enabled	
0	Masked	

DATTIME	[20]	Data Timeout Error Status Enable
1	Enabled	
0	Masked	

CINDEX	[19]	Command Index Error Status Enable
1	Enabled	
0	Masked	

CMDEND	[18]	Command End Bit Error Status Enable
1	Enabled. If this bit is set to 0, the Host Controller shall clear interrupt request to the System. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The Host Driver should clear the Card Interrupt Status Enable before servicing the Card Interrupt and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.	
0	Masked	

CMDCRC	[17]	Command CRC Error Status Enable
1	Enabled	
0	Masked	

CMDTIME	[16]	Command Timeout Error Status Enable
1	Enabled	
0	Masked	

CDINT	[8]	Card Interrupt Status Enable
1	Enabled	
0	Masked	

CDOUT	[7]	Card Removal Status Enable
1	Enabled	
0	Masked	

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

CDIN	[6]	Card Insertion Status Enable
1	Enabled	
0	Masked	

RDRDY	[5]	Buffer Read Ready Status Enable
1	Enabled	
0	Masked	

WRRDY	[4]	Buffer Write Ready Status Enable
1	Enabled	
0	Masked	

DMA	[3]	DMA Interrupt Status Enable
1	Enabled	
0	Masked	

BLKGAP	[2]	Block Gap Event Status Enable
1	Enabled	
0	Masked	

TDONE	[1]	Transfer Complete Status Enable
1	Enabled	
0	Masked	

CDONE	[0]	Command Complete Status Enable
1	Enabled	
0	Masked	

Normal Interrupt Signal Enable

0xF05A0*n*[1]38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VENDOR						ADMA	ACMD12	CLIMIT	DATEND	DATCRC	DATTIME	CINDEX	CMDEND	CMDCRC	CMDTIME
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR							CDINT	CDOUT	CDIN	RDRDY	WRRDY	DMA	BLKGAP	TDONE	CDONE

VENDOR	[31:28]	Vendor Specific Error Signal Enable
1	Enabled	
0	Masked	

ADMA	[25]	ADMA Error Signal Enable
1	Enabled	
0	Masked	

ACMD12	[24]	Auto CMD12 Error Signal Enable
1	Enabled	
0	Masked	

CLIMIT	[23]	Current Limit Error Signal Enable
1	Enabled	
0	Masked	

DATEND	[22]	Data End Bit Error Signal Enable
1	Enabled	
0	Masked	

DATCRC	[21]	Data CRC Error Signal Enable
1	Enabled	
0	Masked	

DATTIME	[20]	Data Timeout Error Signal Enable
1	Enabled	
0	Masked	

CINDEX	[19]	Command Index Error Signal Enable
1	Enabled	
0	Masked	

CMDEND	[18]	Command End Bit Error Signal Enable
1	Enabled	
0	Masked	

CMDCRC	[17]	Command CRC Error Signal Enable
1	Enabled	
0	Masked	

CMDTIME	[16]	Command Timeout Error Signal Enable
1	Enabled	
0	Masked	

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

CDINT	[8]	Card Interrupt Signal Enable
1	Enabled	
0	Masked	

CDOUT	[7]	Card Removal Signal Enable
1	Enabled	
0	Masked	

CDIN	[6]	Card Insertion Signal Enable
1	Enabled	
0	Masked	

RDRDY	[5]	Buffer Read Ready Signal Enable
1	Enabled	
0	Masked	

WRRDY	[4]	Buffer Write Ready Signal Enable
1	Enabled	
0	Masked	

DMA	[3]	DMA Interrupt Signal Enable
1	Enabled	
0	Masked	

BLKGAP	[2]	Block Gap Event Signal Enable
1	Enabled	
0	Masked	

TDONE	[1]	Transfer Complete Signal Enable
1	Enabled	
0	Masked	

CDONE	[0]	Command Complete Signal Enable
1	Enabled	
0	Masked	

Auto CMD12 Error Status **0xF05A0*n*[1]3C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								NOCMD			INDEX	ENDBIT	CRC	TIMEOUT	NORUN

NOCMD	[7]	Command Not Issued
1	Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error (D04 - D01)	
0	No Error	

INDEX	[4]	Auto CMD12 Index Error
1	This bit is set if the Command Index error occurs in response to a command.	
0	No Error	

ENDBIT	[3]	Auto CMD12 End Bit Error
1	This bit is set when detecting that the end bit of command response is 0.	
0	No Error	

CRC	[2]	Auto CMD12 CRC Error
1	This bit is set when detecting a CRC error in the command response.	
0	No Error	

TIMEOUT	[1]	Auto CMD12 Timeout Error
1	This bit is set if the no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to1, the other error status bits (D04-D02) are meaningless.	
0	No Error	

NORUN	[0]	Auto CMD12 Not Executed
1	Setting this bit to 1 means the HC cannot issue Auto CMD12 to stop memory multiple block transfer due to some error. If this bit is set to 1, other error status bits (D04-D01) are meaningless.	
0	Executed	

Table 5.8 Relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Kinds of Error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

Present State **0xF05A0*n*[1]40**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIBLK	SPI	BUS64	INT	V18	V30	V33	RESUME	SDMA	HS		ADMA2	EXTBUS	MAXBLK	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		BASECLK						TUNIT		TIMEOUTCLK					

SPIBLK	[30]	SPI Block Mode
1	Supported	
0	Not supported	

SPI	[29]	SPI Mode
1	Supported	
0	Not supported	

BUS64	[28]	64bit System Bus Support
1	Setting 1 to this bit indicates that the Host Controller supports 64-bit address descriptor mode and is connected to 64-bit address system bus.	
0	Does not support 64 bit system address	

INT	[27]	Interrupt Mode
1	Supported	
0	Not supported	

V18	[26]	Voltage Support 1.8V
1	Supported	
0	Not supported	

V30	[25]	Voltage Support 3.0V
1	Supported	
0	Not supported	

V33	[24]	Voltage Support 3.3V
1	Supported	
0	Not supported	

RESUME	[23]	Suspend / Resume Support
1	Supported	
0	Not supported	

SDMA	[22]	SDMA Support
1	Supported	
0	Not supported	

HS	[21]	High Speed Support
1	Supported	
0	Not supported	

ADMA2	[19]	ADMA2 Supported
1	Supported	
0	Not supported	

EXTBUS	[18]	Extended Media Bus Support
1	Supported	
0	Not supported	

MAXBLK	[17:16]	Max Block Length
N	This value indicates the maximum block size that the Host Driver can read and write to the buffer in the Host Controller. The buffer shall transfer this block size without wait cycles. It is noted that transfer block length shall be always 512 bytes for SD Memory Cards regardless of this field. 00 - 512 byte	

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

	01 - 1024 byte 10 - 2048 byte 11 - 4096 byte

BASECLK	[13:8]	Base Clock Frequency For SD Clock
N	This value indicates the base (maximum) clock frequency for the SD clock. Unit values are MHz. If the real frequency is 16.5MHz, the lager value shall be set 01 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value Not 0 - 1 MHz to 63 MHz 000000b - Get information via another method.	

TUNIT	[7]	Timeout Clock Unit
1	This bit shows the unit of base clock frequency used to detect Data Timeout Error. The unit is MHz.	
0	The unit is KHz.	

TIMEOUTCLK	[5:0]	Timeout Clock Frequency
N	This bit shows the base clock frequency used to detect Data Timeout Error. The Timeout Clock Unit defines the unit of this field's value. Timeout Clock Unit =0 [KHz] unit: 1KHz to 63KHz Timeout Clock Unit =1 [MHz] unit: 1MHz to 63MHz Not 0 - 1KHz to 63Khz or 1MHz to 63MHz 000000b - Get Information via another method	

Maximum Current Capabilities 0xF05A0*n*[1]48

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								MAXCURV18[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MAXCURV30[7:0]								MAXCURV33[7:0]							

MAXCURV18	[23:16]	Maximum Current for 1.8V
	Maximum Current for 1.8V *This will be supported in the future.*	

MAXCURV30	[15:8]	Maximum Current for 3.0V
	Maximum Current for 3.0V *This will be supported in the future.*	

MAXCURV33	[7:0]	Maximum Current for 3.3V
	Maximum Current for 3.3V *This will be supported in the future.*	

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

Force Event for Error Interrupt / Auto CMD12 Error Status 0xF05A0*n*[1]50

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VENDOR[3:0]				Reserved		ADMA	ACMD12	CLIMIT	DATEND	DATCRC	DATTIME	CINDEX	CMDEND	CMDCRC	CMDTIME
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved								NOCMD	Reserved		INDEX	ENDBIT	CRC	TIMEOUT	NORUN

VENDOR	[31:28]	Force Event For Vendor Specific Error Status
1	Generate interrupt	
0	No interrupt	

ADMA	[25]	Force Event For ADMA Error
1	Generate interrupt	
0	No interrupt	

ACMD12	[24]	Force Event For Auto CMD12 Error
1	Generate interrupt	
0	No interrupt	

CLIMIT	[23]	Force Event For Current Limit Error
1	Generate interrupt	
0	No interrupt	

DATEND	[22]	Force Event For Data End Bit Error
1	Generate interrupt	
0	No interrupt	

DATCRC	[21]	Force Event For Data CRC Error
1	Generate interrupt	
0	No interrupt	

DATTIME	[20]	Force Event For Data Timeout Error
1	Generate interrupt	
0	No interrupt	

CINDEX	[19]	Force Event For Command Index Error
1	Generate interrupt	
0	No interrupt	

CMDEND	[18]	Force Event For Command End Bit Error
1	Generate interrupt	
0	No interrupt	

CMDCRC	[17]	Force Event For Command CRC Error
1	Generate interrupt	
0	No interrupt	

CMDTIME	[16]	Force Event For Command Timeout Error
1	Generate interrupt	
0	No interrupt	

NOCMD	[7]	Force Event For Command Not Issued
1	Generate interrupt	
0	No interrupt	

INDEX	[4]	Force Event For Auto CMD12 Index Error
1	Generate interrupt	
0	No interrupt	

ENDBIT	[3]	Force Event For Auto CMD12 End Bit Error
1	Generate interrupt	
0	No interrupt	

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

CRC	[2]	Force Event For Auto CMD12 CRC Error
1	Generate interrupt	
0	No interrupt	

TIMEOUT	[1]	Force Event For Auto CMD12 Timeout Error
1	Generate interrupt	
0	No interrupt	

NORUN	[0]	Force Event For Auto CMD12 Not Executed
1	Generate interrupt	
0	No interrupt	

ADMA Error Status 0xF05A0*n*[1]54

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													LEN	ERRSTATE	

LEN	[2]	ADMA Length Mismatch Error
1	This error occurs in the following 2 cases. (1) While Block Count Enable being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length. (2) Total data length can not be divided by the block length.	
0	No Error	

ERRSTATE	[1:0]	ADMA Error State
N	This field indicates the state of ADMA when error is occurred during ADMA data transfer. This field never indicates "10" because ADMA never stops in this state. 00 - ST_STOP (Stop DMA) Points next of the error descriptor 01 - ST_FDS (Fetch Descriptor) Points the error descriptor 10 - Never set this state (Not used) 11 - ST_TFR (Transfer Data) Points the next of the error descriptor	

ADMA System Address0 0xF05A0n58

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[15:0]															

ADMA System Address1 0xF05A0n5C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[47:32]															

ADDR	[63:0]	ADMA System Address
N	This register holds byte address of executing command of the Descriptor table. 32-bit Address Descriptor uses lower 32-bit of this register. At the start of ADMA, the Host Driver shall set start address of the Descriptor table. The ADMA increments this register address, which points to next line, when every fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register shall hold valid Descriptor address depending on the ADMA state. The Host Driver shall program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. ADMA2 ignores lower 2-bit of this register and assumes it to be 00b.	

Refer to the SD Host Controller Standard Specification V2.00 of SD Association to know how to make the descriptor table.

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

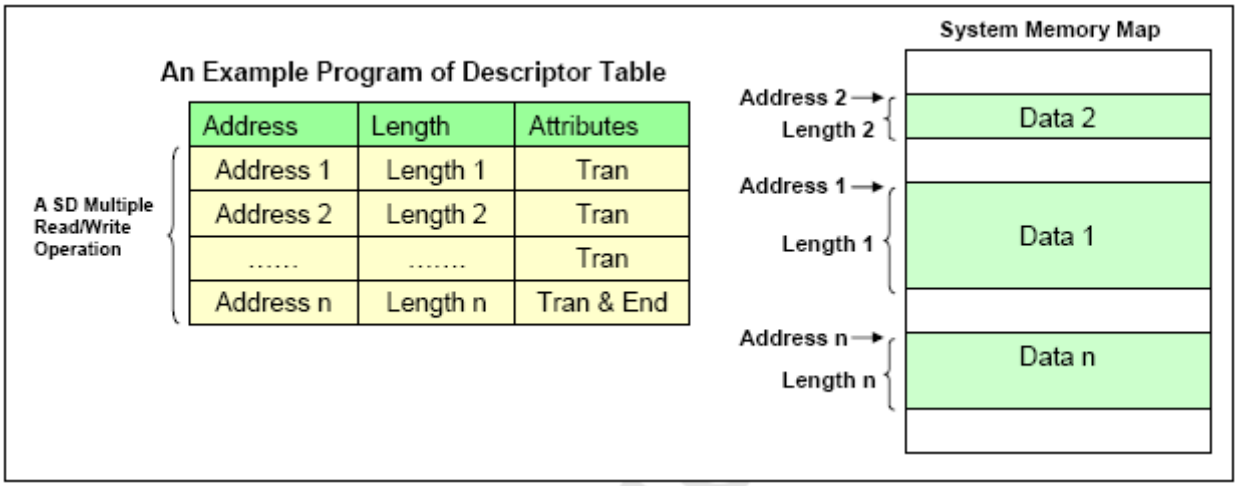


Figure 5.2 An Example of ADMA2 Data Transfer

Figure 5.2 shows a typical ADMA2 descriptor program. The data area is sliced in various lengths and each slice is placed somewhere in system memory. The Host Driver describes the Descriptor Table with set of address, length and attributes. Each sliced data is transferred in turns as programmed in descriptor.

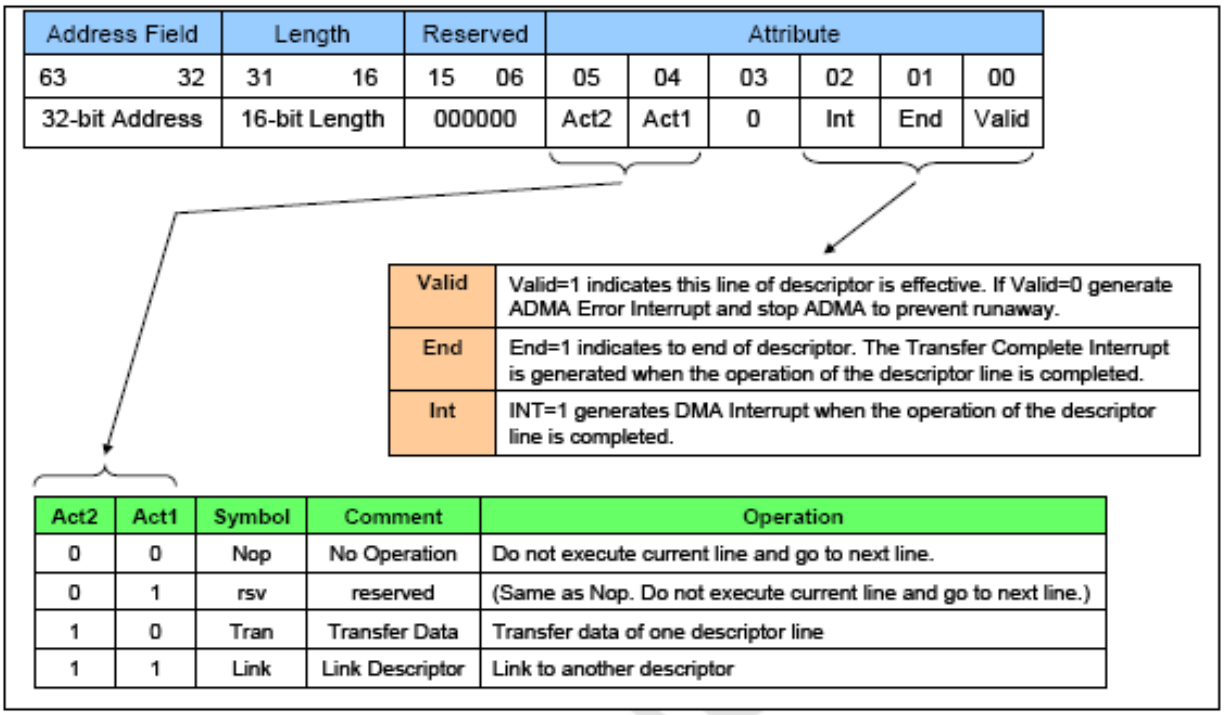


Figure 5.3 32-bit Address Descriptor Table

SPI Interrupt Support **0xF05A0*n*[1]F0**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SPIINT[7:0]							

SPIINT	[7:0]	SPI Interrupt Support
N	This bit is set to indicate the assertion of interrupts in the SPI mode at any time, irrespective of the status of the card select (CS) line. If this bit is zero, then SDIO card can only assert the interrupt line in the SPI mode when the CS line is asserted.	

Host Controller Version / Slot Interrupt Status **0xF05A0*n*[1]FC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								SLOTINT[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VENDOR[7:0]								SPEC[7:0]							

SLOTINT	[23:16]	Interrupt Signal For Each Slot
N	These status bit indicate the logical OR of Interrupt signal and Wakeup signal for each slot. Bit 00 - Slot 1 Bit 01 - Slot 2 Bit 02 - Slot 3 ----- ----- Bit 07 - Slot 8	

VENDOR	[15:8]	Vendor Version Number
N	This status is reserved for the vendor version number. The HD should not use this status.	

SPEC	[7:0]	Specification Version Number
N	This Status indicates the Host Controller Spec Version. 00 - SD Host Specification version 1.0 01 - SD Host Specification version 2.00 including only the feature of theTest Register 02 - SD Host Specification version 2.00 including the feature of the Test Register and ADMA others – Reserved	

5.4 Timing Diagram

5.4.1 Command Format

All commands have a fixed code length of 48 bits, needing a transmission time of 1.92 μs @ 25 MHz and 0.96 μs @ 50 MHz.

Table 5.9 Command Format

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	'1'
Description	Start bit	Transmission bit	Command index	Argument	CRC7	End bit

A command always starts with a start bit (always 0), followed by the bit indicating the direction of transmission (host = 1). The next 6 bits indicate the index of the command, this value being interpreted as a binary coded number (between 0 and 63). Some commands need an argument (e.g. an address), which is coded by 32 bits. A value denoted by 'x' in the table above indicates this variable is dependent on the command. All commands are protected by a CRC. Every command codeword is terminated by the end bit (always 1).

5.4.2 Timing of the Transaction in SD Mode

[1] n = 0 or 1 for two slots in core0, n = 2 or 3 for two slots in core1.

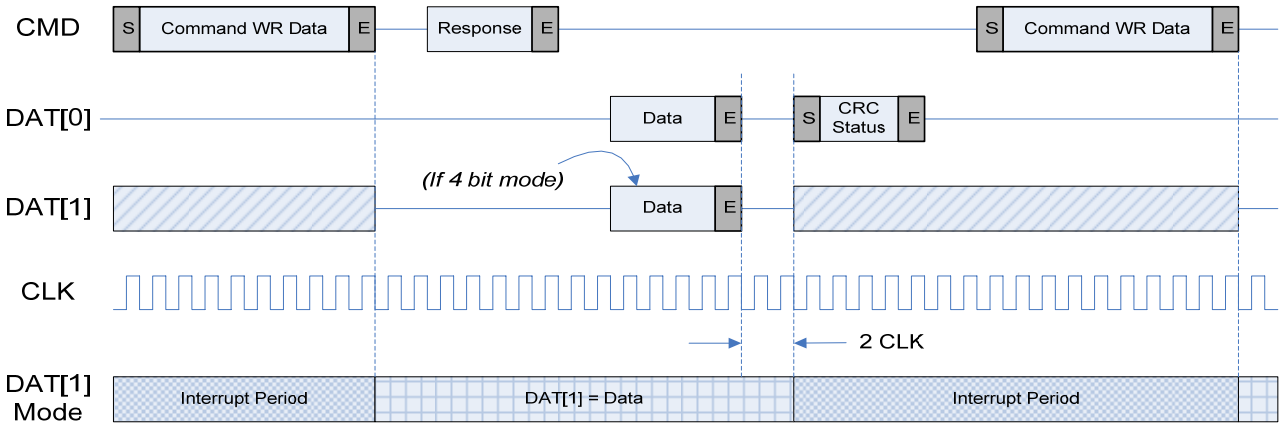


Figure 5.4 SDIO/SD – Write Interrupt Cycle Timing

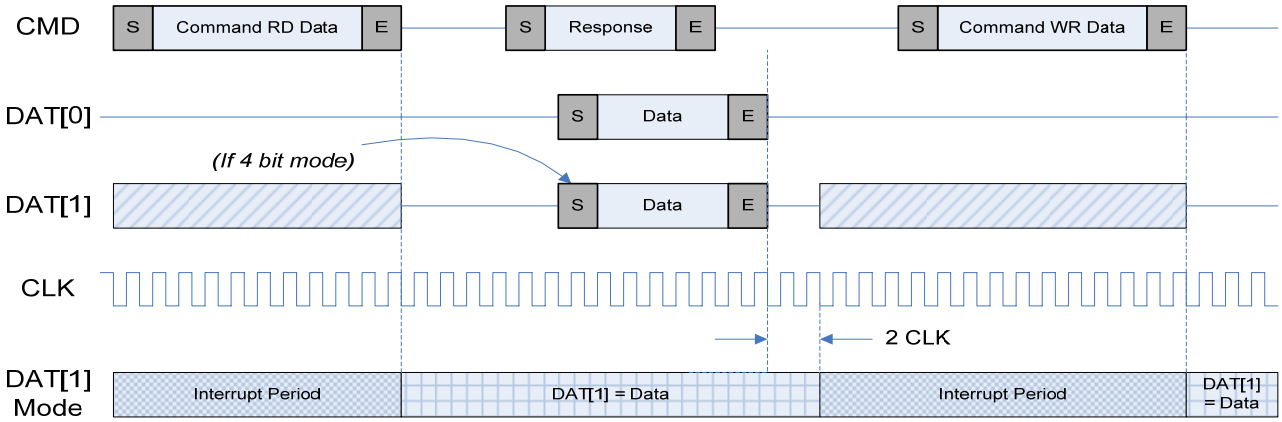


Figure 5.5 SDIO/SD – Read Interrupt Cycle Timing

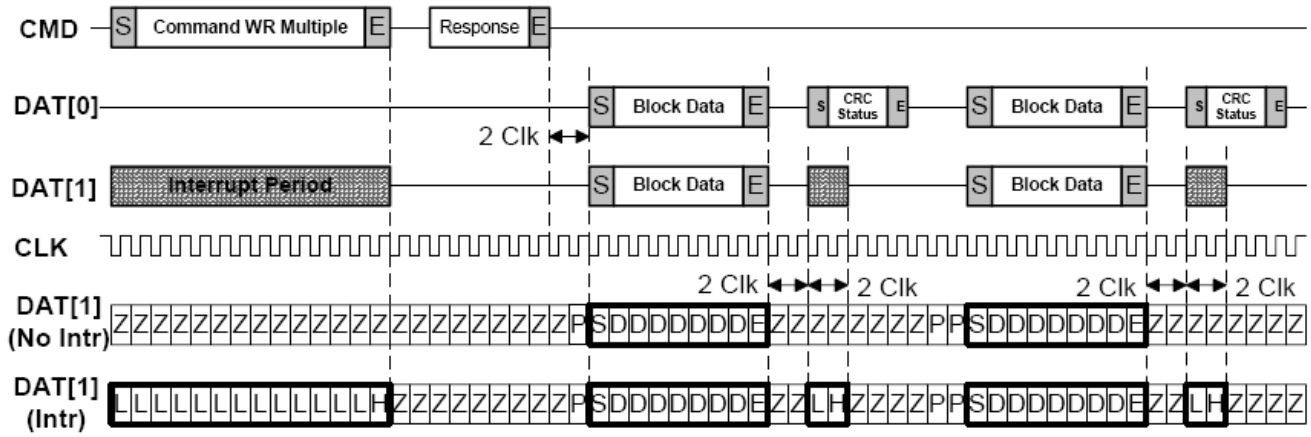


Figure 5.6 SDIO/SD – Multiple Block 4-Bit Write Interrupt Cycle Timing

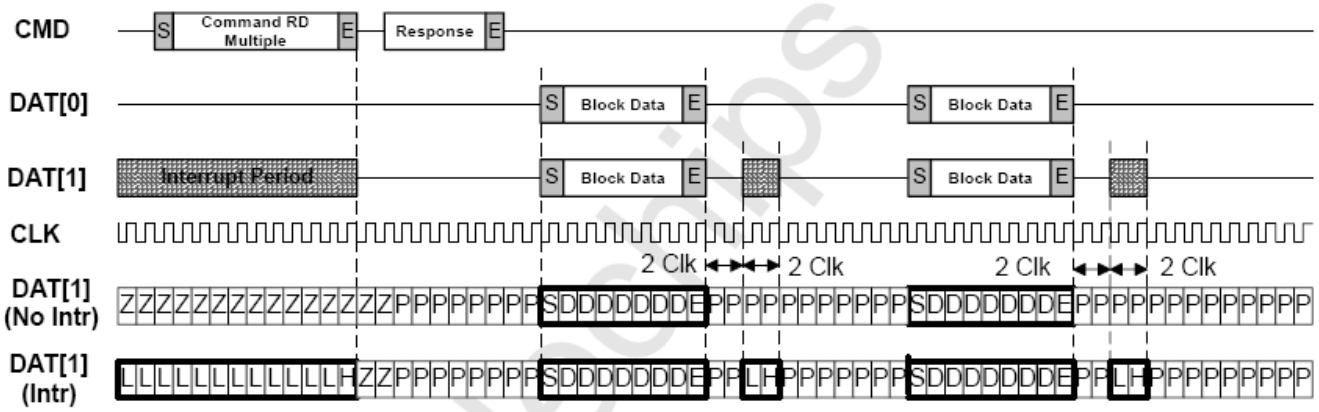


Figure 5.7 SDIO/SD – Multiple Block 4-Bit Read Interrupt Cycle Timing

5.4.3 Timing of the Transaction in CE-ATA Mode

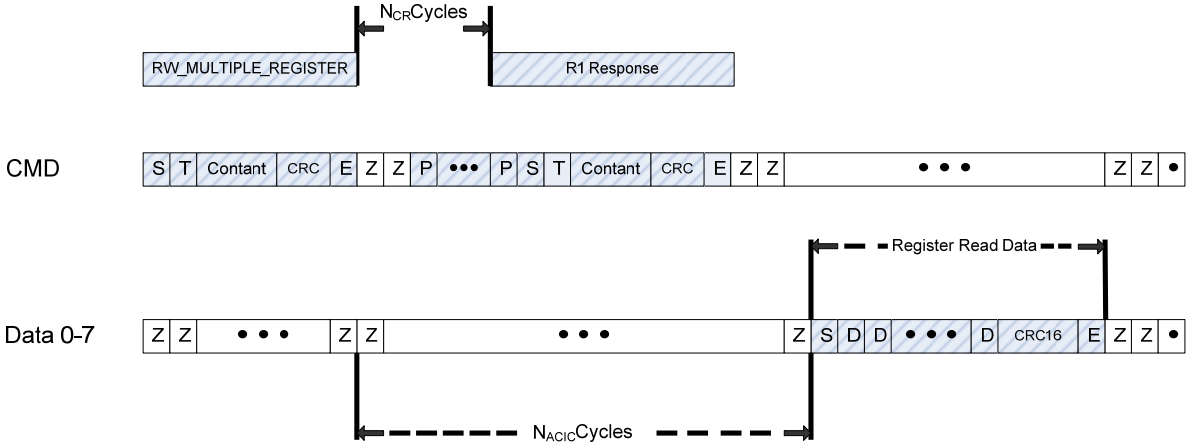


Figure 5.8 Multiple Register Read (CMD60)Transaction

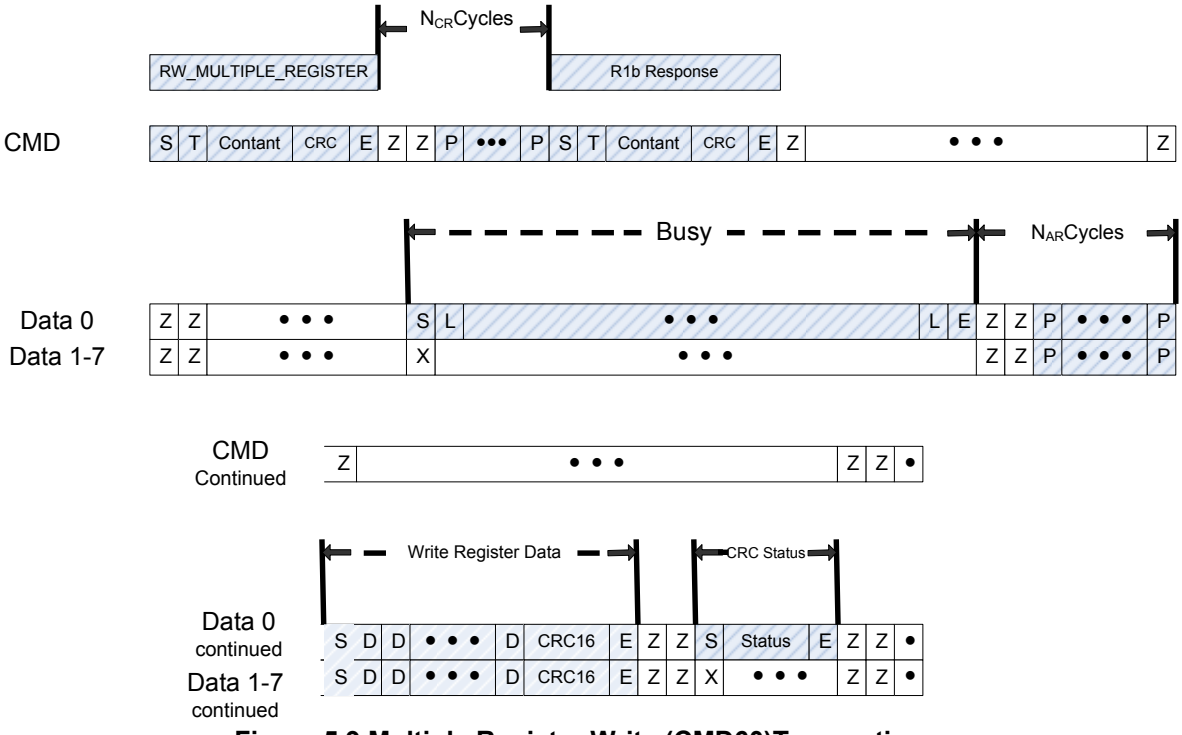


Figure 5.9 Multiple Register Write (CMD60)Transaction

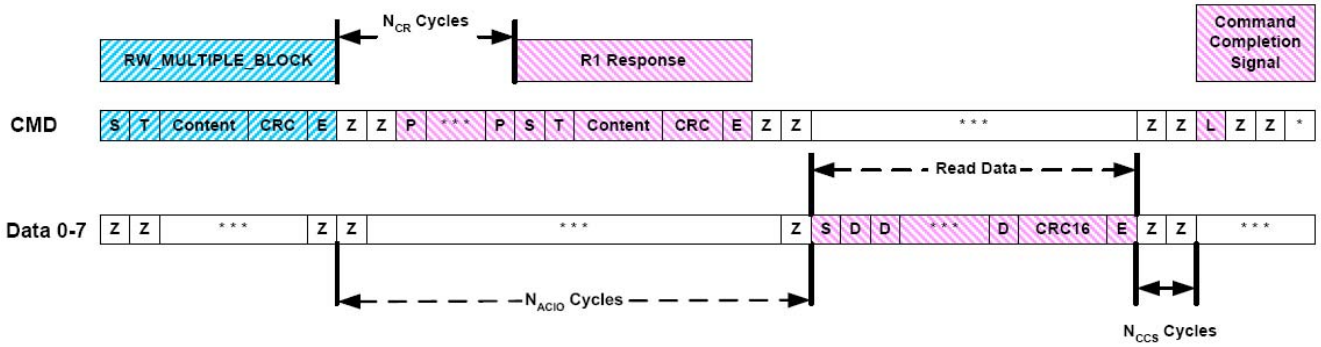


Figure 5.10 Single Block Read (CMD61) Transaction

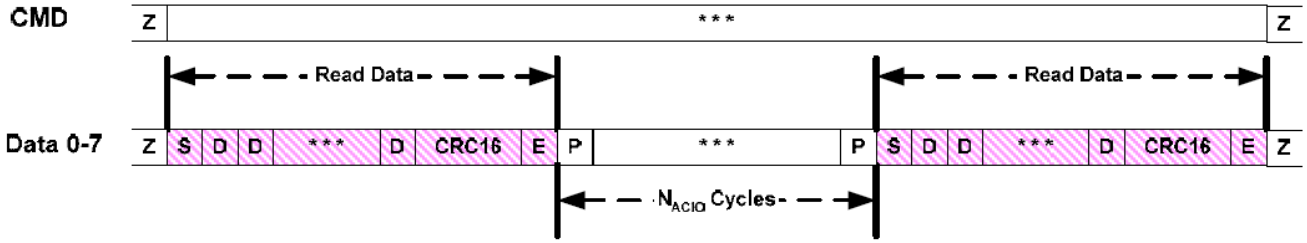
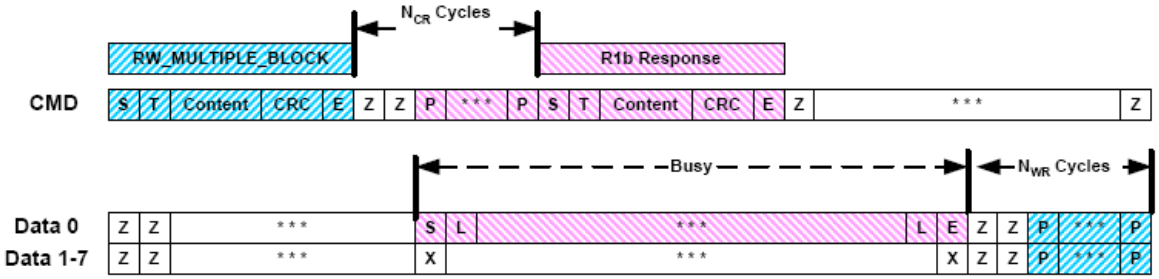


Figure 5.11 Multiple Block Read (CMD61) Transaction



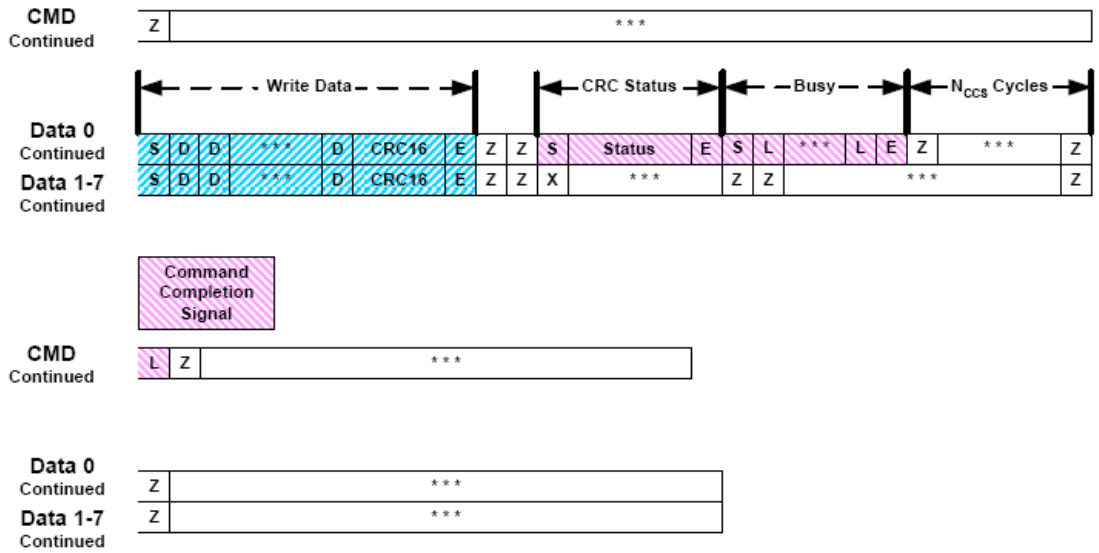


Figure 5.12 Single Block Write (CMD61) Transaction

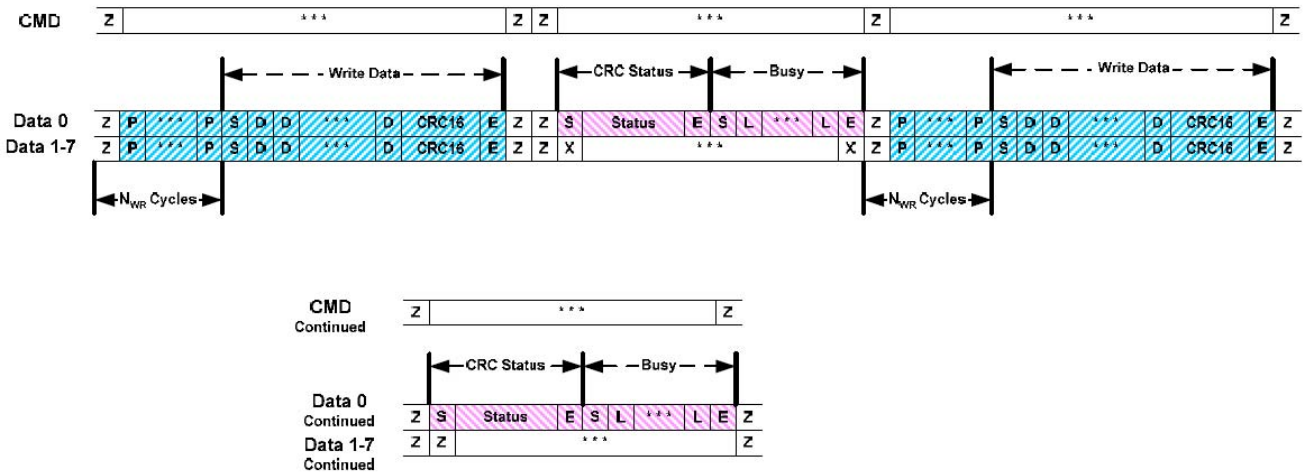


Figure 5.13 Multiple Block Write (CMD61) Transaction

6 NFC

6.1 Function Description

TCC8900 has a 8/16 bit NAND flash controller which can support various type of NAND flash such as 532(512 + 16)bytes/page, 2112(2048+64)bytes/page, 4314(4096 + 218)bytes/page NAND and so on.

The supportable configuration for external NAND flash is as follows.

Case1) one NAND flash of 8-bit bus-width.
Case2) one NAND flash of 16-bit bus-width
Case3) two NAND flashes of 8-bit bus-width with the same chip enable signal

The supportable main features are as follows

- 8 / 16 bit bus width
- 1/2/4/8 burst program / read operation by General DMA
- 1/2/4/8 burst program / read operation by CPU
- single address / linear address
- single 8 bit or 16 bit data read/write
- multiple 8 bit or 16 bit data read/write
- single word data read/write
- 4 chip selection / 1 ready port
- 16 X 32 bit FIFO (MAX burst size = 8 burst)

The block diagram of NFC is in Figure 6.1. All control and data I/O signals are connected to external NAND flash through EDI block. In some cases, after GPIO_B[31:0] port selection, the EDI block should be configured additionally.

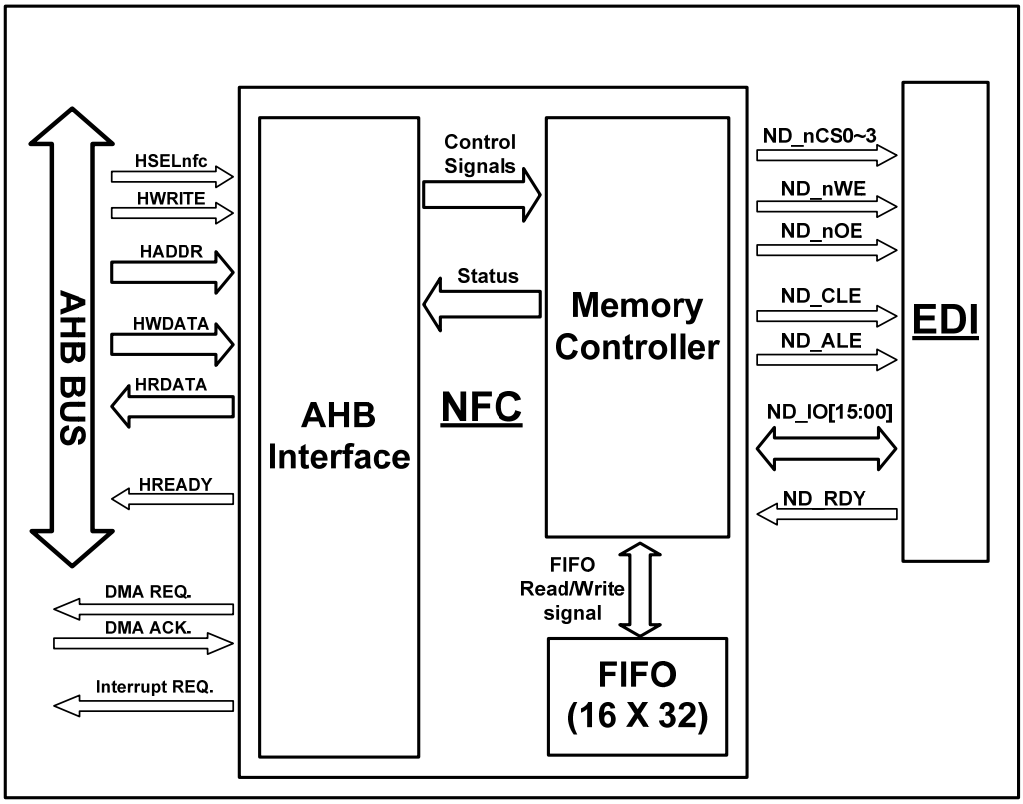


Figure 6.1 NAND Flash Controller Block Diagram

Figure 6.2 shows external NAND flash connection example. With default configuration, i.e. Each ND_nCS0 and ND_nCS1 is connected to EDI_CSN5 and EDI_CSN6. NAND programming and read operation is possible without any additional EDI configuration step. (Refer to 8 . EDI (External Device Interface)“)

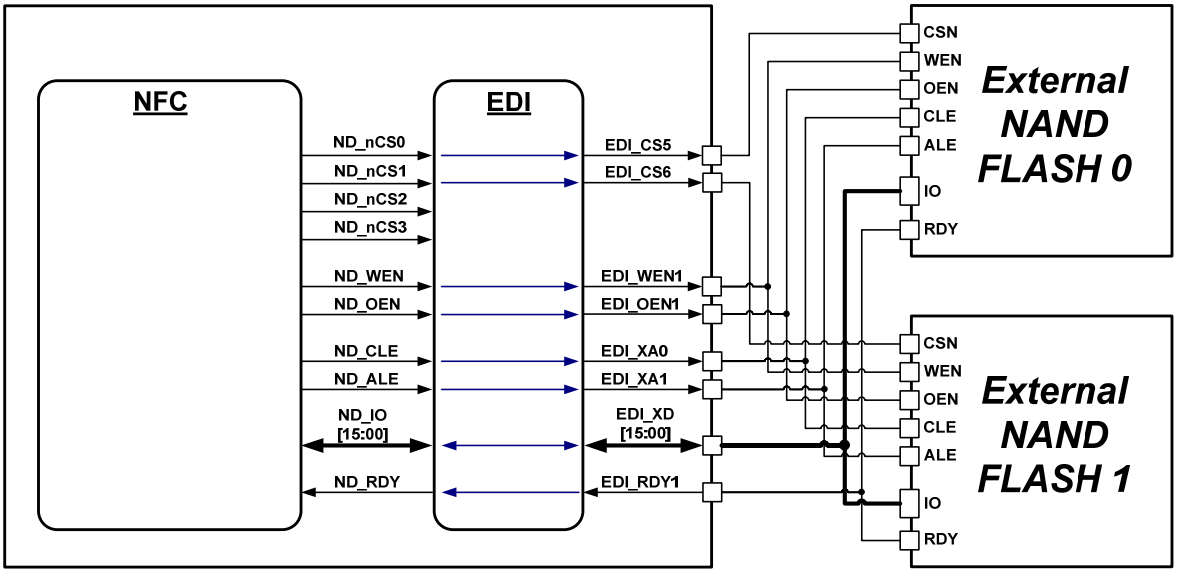


Figure 6.2 An Example of Interface for Two NAND Flash Memory

6.2 Register Description

Table 6.1 NAND Flash Controller Register Map (Base Address=0xF05B0000)

Name	Address	Type	Reset	Description
NFC_CMD	0x00	W	-	NAND Flash Command Register
NFC_LADDR	0x04	W	-	NAND Flash Linear Address Register
NFC_BADDR	0x08	W	-	NAND Flash Block Address Register
NFC_SADDR	0x0C	W	-	NAND Flash Signal Address Register
NFC_WDATA	0x1x	R/W	Unknown	NAND Flash Word Data Register
NFC_LDATA	0x2x/3x	R/W	Unknown	NAND Flash Linear Data Register
NFC_SDATA	0x40	R/W	Unknown	NAND Flash Single Data Register
NFC_CTRL	0x50	R/W	0x03e08000	NAND Flash Control Register
NFC_PSTART	0x54	W	-	NAND Flash Program Start Register
NFC_RSTART	0x58	W	-	NAND Flash Read Start Register
NFC_DSIZE	0x5C	R/W	0x0000ffff	NAND Flash Data Size Register
NFC_IREQ	0x60	R/W	0x07000000	NAND Flash Interrupt Request Register
NFC_RST	0x64	W	-	NAND Flash Controller Reset Register
NFC_CTRL1	0x68	R/W	0x00000000	NAND Flash Control Register 1
NFC_MDATA	0x7x	R/W	Unknown	NAND Flash Multiple Data Register

Command Register (NFC_CMD) 0xF05B0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD1[7:0]								CMD0[7:0]							

For 16bit bus width of NAND flash, the NFC_CMD1 may be used as command register

For 8bit bus width of NAND flash, if MSK of NFC_CTRL is Low, the NFC_CMD1 must be Zero.

- NFC_CMD = 0xFFFF and MSK(NFC_CTRL[15]) = 1 : ND_IO[15:0] = 0x00FF
- NFC_CMD = 0xFFFF and MSK(NFC_CTRL[15]) = 0 : ND_IO[15:0] = 0xFFFF

The following values are an example commands for NAND flash of SAMSUNG. Refer to corresponding datasheet of NAND flash chip for more detailed list of commands.

(For 16bit bus width)	(for 8bit parallel configuration, 16bit bus width)	
0x0000	0x0000	: Page Read Command
0x0080	0x8080	: Page Program Command
0x0060	0x6060	: Block Erase Command
0x0070	0x7070	: Status Read Command

Linear Address Register (NFC_LADDR) 0xF05B0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LADDR[15:0]															

By writing to this register, memory controller generates linear address shown in Table 6.2

For NFC_LADDR = 0x12345600, CADDR NFC_CTRL[14:12]) = 2 and PSIZE(NFC_CTRL[17:16]) = 1 (512 bytes)

- MSK(NFC_CTRL[15]) = 1 : ND_IO[15:0] = 0x0000 – 0x002B – 0x001A
- MSK(NFC_CTRL[15]) = 0 : ND_IO[15:0] = 0x0000 – 0x2B2B – 0x1A1A

Block Address Register (NFC_BADDR) 0xF05B0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BADDR[15:0]															

By writing to this register, memory controller generates Block address shown in Table 6.2.

For NFC_BADDR = 0x12345600, CADDR(NFC_CTRL[14:12]) = 2 and PSIZE(NFC_CTRL[17:16]) = 1

- MSK(NFC_CTRL[15]) = 1 : ND_IO[15:0] = 0x002B – 0x001A
- MSK(NFC_CTRL[15]) = 0 : ND_IO[15:0] = 0x2B2B – 0x1A1A

Single Address Cycle Register (NFC_SADDR) 0xF05B000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR1[7:0]								SADDR0[7:0]							

For 16bit bus width of NAND flash, the NFC_SADDR1 may be used as single address register.

For 8bit bus width of NAND flash, if MSK of NFC_CTRL is Low, the NFC_SADDR1 must be Zero.

- NFC_SADDR = 0xFFFF and MSK(NFC_CTRL[15]) = 1 : ND_IO[15:0] = 0x00FF
- NFC_SADDR = 0xFFFF and MSK(NFC_CTRL[15]) = 0 : ND_IO[15:0] = 0xFFFF

When CPU writes to this register, one cycle of address cycle is generated.

(For 16bit bus width) (for 8bit parallel configuration, 16bit bus width)
0x0012 0x1212 : for 0x12 Single Address

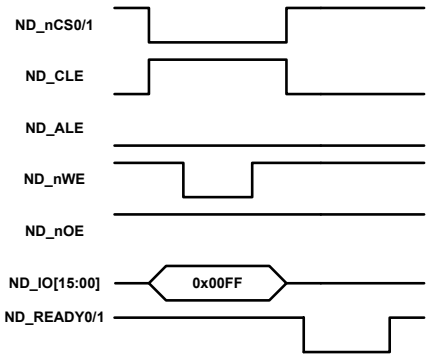
Table 6.2 Page Size of NAND Flash

# of Cycle (CADDR)	PSIZE = 0	PSIZE = 1	PSIZE = 2	PSIZE = 3	PSIZE > 3
1st	ADDR[7:0]	ADDR[7:0]	ADDR[7:0]	ADDR[7:0]	ADDR[7:0]
2nd	ADDR[16:9]	ADDR[16:9]	ADDR[10:8]	ADDR[11:8]	ADDR[12:8]
3rd	ADDR[24:17]	ADDR[24:17]	ADDR[18:11]	ADDR[19:12]	ADDR[20:13]
4th	ADDR[31:25]	ADDR[31:25]	ADDR[26:19]	ADDR[27:20]	ADDR[28:21]
5th	-	-	ADDR[31:27]	ADDR[31:28]	ADDR[31:29]

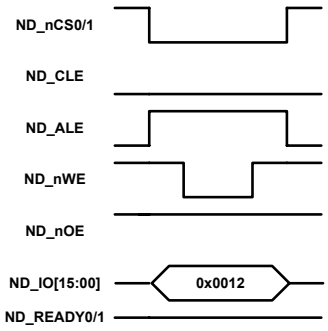
The Figure 6.3 represents the relation between each cycle and address generation.

User must set this information appropriately to PSIZE and CADDR field of NFC_CTRL register ahead of accessing NAND data.

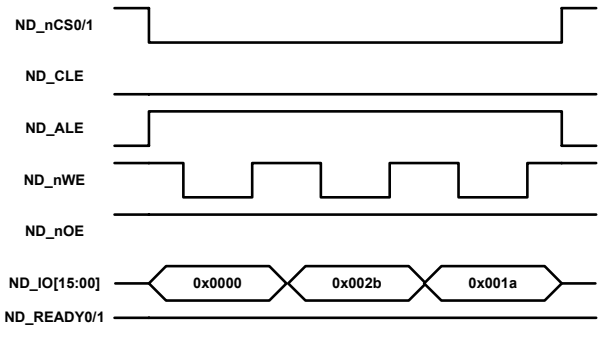
ADDR means address value that is written to NFC_LADDR or NFC_BADDR register. The shaded cycles represent Block address cycles. That is, NAND address cycles start from there when NFC_BADDR register is accessed.



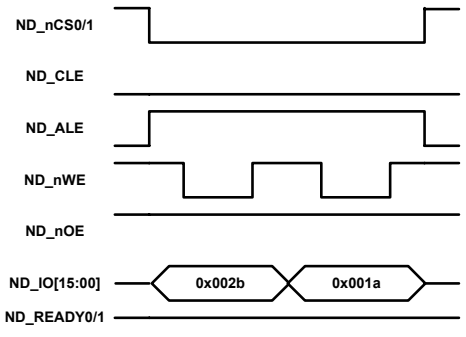
For Writing 0x00FF in NFC_CMD Register
1) COMMAND



For Writing 0x00000012 in NFC_SADDR Register
2) SINGLE ADDRESS



For Writing 0x12345600 in NFC_LADDR Register
- CADDR = 2 (3 Cycle)
- PSIZE = 512 Byte
3) LINEAR ADDRESS



For Writing 0x12345600 in NFC_BADDR Register
- CADDR = 2 (3 Cycle)
- PSIZE = 512 Byte
3) BLOCK ADDRESS

Figure 6.3 Example of Address/Command Writing Operation

Word Data Register (NFC_WDATA) 0xF05B001x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NFC_WDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NFC_WDATA[15:0]															

This register is used for single word data transfer by CPU.

This Register is useful in reading and writing NAND Flash data of spare area.

Linear Data Register (NFC_LDATA) 0xF05B002x/0xF05B003x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NFC_LDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NFC_LDATA[15:00]															

This register is used for burst data transfer by DMA/CPU.

To start burst data transfer, User must write any value to NFC_PSTART or NFC_RSTART

Single Data Register (NFC_SDATA) 0xF05B0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NFC_SDATA[7:0]								NFC_SDATA[7:0]							

This register is used for reading and writing one 8bit or 16bit data according to NAND Flash Bus Size.

For 16bit bus width of NAND flash, the NFC_SDATA0/1 may be used as single data register, otherwise only NFC_SDATA0 is used as single data register.

This Register is useful in reading NAND ID/Status data.

Multiple Data Register (NFC_MDATA) 0xF05B007x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NFC_MDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NFC_MDATA[15:0]															

This register is used for reading and writing multiple 8bit or 16bit data according to NAND Flash Bus Size.

NFC_CTRL1.DNUM[01:00] register defined the number of reading and writing.

NAND Flash Control Register (NFC_CTRL) 0xF05B0050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IEN	IEN	IEN	DEN	FS	BW	CS3	CS2	CS1	CS0	RDY	BSIZE[1:0]		PSIZE[2:0]		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MSK	CADDR[2:0]			bSTP[3:0]				bPW[3:0]				bHLD[3:0]			

RDY_IEN	[31]	NAND Flash Ready Interrupt
0	NAND Flash Ready Interrupt Disable	
1	NAND Flash Ready Interrupt Enable	

PROG_IEN	[30]	NAND Flash Program Interrupt
0	NAND Flash Program Interrupt Disable	
1	NAND Flash Program Interrupt Enable	

READ_IEN	[29]	NAND Flash Read Interrupt
0	NAND Flash Read Interrupt Disable	
1	NAND Flash Read Interrupt Enable	

DEN	[28]	NAND Flash DMA Request
0	NAND Flash DMA Request Disable	
1	NAND Flash DMA Request Enable	

Transfer type of CHCTRL register in DMA must be single transfer with level-sensitive detection.

FS	[27]	NAND Flash FIFO Status
0	FIFO status is Busy to write and read in FIFO	
1	FIFO status is Ready to write and read in FIFO	

For burst data transfer by ARM, user must check that this bit is high, ahead of access NFC_LDATA.

BW	[26]	NAND Flash Bus Width Select
0	Bus width = 8 bit	
1	Bus width = 16 bit	

CS3SEL	[25]	NAND Flash CS3 Selection
0	NAND Flash nCS3 is Enable	
1	NAND Flash nCS3 is Disable	

CS2SEL	[24]	NAND Flash CS2 Selection
0	NAND Flash nCS2 is Enable	
1	NAND Flash nCS2 is Disable	

CS1SEL	[23]	NAND Flash CS1 Selection
0	NAND Flash nCS1 is Enable	
1	NAND Flash nCS1 is Disable	

CS0SEL	[22]	NAND Flash CS0 Selection
0	NAND Flash nCS0 is Enable	
1	NAND Flash nCS0 is Disable	

RDY	[21]	NAND Flash Ready Flag
0	External NAND Flash is Busy (read-only)	
1	External NAND Flash is Ready (read-only)	

BSIZE[1:0]	[20:19]	Burst Size of NAND Controller
00	1Read/Write	

01	2Read/Write
10	4Read/Write
11	8Read/Write

This register value must be same BURST SIZE of CHCTRL in DMA.

PSIZE[2:0]	[18:16]	Page Size of NAND Flash
000	1 Page = 256 Half-Word	
001	1 Page = 512 Byte	
010	1 Page = 1024 Half-Word	
011	1 Page = 2048 Byte	
1xx	1 Page = 4096 Byte	

A Linear address and Block address for external NAND is affected by CTRL[PSIZE]. If the single address mode is only used, ignore CRTL[PSIZE].

MSK	[15]	NAND Flash IO Mask Enable Bit
0	NAND Flash High Byte(ND_IO[15:8]) is not mask	
1	NAND Flash High Byte(ND_IO[15:8]) is mask	

CADDR[2:0]	[14:12]	Number of Address Cycles
N	The number of address command cycle for NAND type flash. (N+1) cycle is used for generating address cycle command.	

bSTP[3:0]	[11:08]	Number of Base Cycle for Setup Time
N (= 0~15)	bSTP = N	

Set-Up Cycle of NAND Flash (STP)

WSTP(Write Set-Up Cycle) = bSTP(NFC_CTRL[11:8]) + wSTP(NFC_CTRL1[11:8])

RSTP(Read Set-Up Cycle) = bSTP(NFC_CTRL[11:8]) + rSTP(NFC_CTRL1[23:20])

For Command and Address Cycle

- For WSTP = 0 : Set-Up Cycle of NAND Flash (STP) = 1
- For WSTP != 0 : Set-Up Cycle of NAND Flash (STP) = WSTP

For Single Data(NFC_SDATA) Write Cycle

- Set-Up Cycle of NAND Flash (STP) = WSTP

For Single Data(NFC_SDATA) Read Cycle

- Set-Up Cycle of NAND Flash (STP) = RSTP

For Word Data(NFC_WDATA) Write Cycle

- For WSTP = 0 : Set-Up Cycle of NAND Flash (STP) = 1
- For WSTP != 0 : Set-Up Cycle of NAND Flash (STP) = WSTP

For Word Data(NFC_WDATA) Read Cycle

- For RSTP = 0 : Set-Up Cycle of NAND Flash (STP) = 1
- For RSTP != 0 : Set-Up Cycle of NAND Flash (STP) = RSTP

For Linear Data(NFC_LDATA) Write Cycle

- Set-Up Cycle of NAND Flash (STP) = WSTP

For Linear Data(NFC_LDATA) Read Cycle

- Set-Up Cycle of NAND Flash (STP) = RSTP

bPW[3:0]	[07:04]	Number of Base Cycle for Pulse Width

N (= 0~15)	bPW = N

Pule-Width Cycle of NAND Flash (PW)

WPW(Write Pulse Width Cycle) = bPW(NFC_CTRL[7:4]) + wPW(NFC_CTRL1[7:4])

RPW(Read Pulse Width Cycle) = bPW(NFC_CTRL[7:4]) + rPW(NFC_CTRL1[19:16])

For Command and Address Cycle

- Pulse Width Cycle of NAND Flash (PW) = WPW

For Single/Word/Linear Data(NFC_SDATA / NFC_WDATA / NFC_LDATA) Write Cycle

- Pulse Width Cycle of NAND Flash (PW) = WPW

For Single/Word/Linear Data(NFC_SDATA / NFC_WDATA / NFC_LDATA) Read Cycle

- Pulse Width Cycle of NAND Flash (PW) = RPW

bHLD[3:0]	[03:00]	Number of Base Cycle for Hold Time
N (= 0~15)	bHLD = N	

Hold Cycle of NAND Flash (HLD)

WHLD(Write Hold Cycle) = bHLD(NFC_CTRL[3:0]) + wHLD(NFC_CTRL1[3:0])

RHLD(Read Hold Cycle) = bHLD(NFC_CTRL[3:0]) + rHLD(NFC_CTRL1[15:12])

For Command and Address Cycle

- For WHLD = 0 : Hold Cycle of NAND Flash (HLD) = 1
- For WHLD != 0 : Hold Cycle of NAND Flash (HLD) = WHLD

For Single/Word/Linear Data(NFC_SDATA / NFC_WDATA / NFC_LDATA) Write Cycle

- For WHLD = 0 : Hold Cycle of NAND Flash (HLD) = 1
- For WHLD != 0 : Hold Cycle of NAND Flash (HLD) = WHLD

For Single/Word/Linear Data(NFC_SDATA / NFC_WDATA / NFC_LDATA) Read Cycle

- For RHLD = 0 : Hold Cycle of NAND Flash (HLD) = 1
- For RHLD != 0 : Hold Cycle of NAND Flash (HLD) = RHLD

The following figure displays the element cycle diagram for external memories.

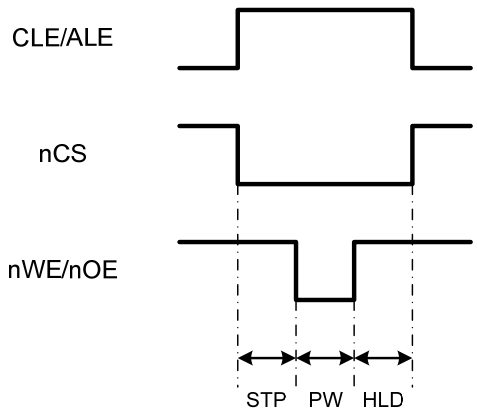


Figure 6.4 Timing Diagram of Read/Write Enanle Signal

Program Start Register (NFC_PSTART) **0xF05B0054**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Don't care															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Don't care															

When this register is written by any value, data transfer is started. That is, Programming operation to NAND flash is started.

For Burst data transfer by DMA/ARM, you must set EN flag of DMA_CHCTRL register first, and then write any value to NFC_PSTART ahead of accessing NFC_LDATA.

Read Start Register (NFC_RSTART) 0xF05B0058

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Don't care															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Don't care															

When this register is written by any value, data transfer is started. That is, Reading operation to NAND flash is started.

For Burst data transfer by DMA/ARM, you must set EN flag of DMA_CHCTRL register first, and then write any value to NFC_RSTART ahead of accessing NFC_LDATA.

Data Size Register (NFC_DSIZE) 0xF05B005C

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NFC_DSIZE[15:0]															

NFC_DSIZE	NAND Flash Data Size
N	N is transferred byte data size

For Samsung 258 Half-word/512 Byte small block, N is 512(byte).

For Samsung 1024 Half-Word/2048 Byte big block, N is 2048(byte).

NAND Flash Request Register (NFC_IREQ) 0xF05B0060

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0									NFC_FLAG[2:0]			0	NFC_IREQ[2:0]		

FLAG[2]	[6]	NAND Flash Ready Flag
1	Read	The Rising edge of Ready Signal is occurred.
1	Write	Ready Flag Clear

When this register is cleared, ready interrupt request also is cleared.

FLAG[1]	[5]	NAND Flash Program Flag
1	Read	Program data transfer is finished.
1	Write	Program Flag Clear

When this register is cleared, program interrupt request also is cleared.

FLAG[0]	[4]	NAND Flash Read Flag
1	Read	Read data transfer is finished.
1	Write	Read Flag Clear

When this register is cleared, read interrupt request also is cleared.

IRQ[2]	[2]	NAND Flash Ready Interrupt Request
1	Read	Ready Interrupt is occurred.
1	Write	Ready Interrupt Request Clear

When this register is cleared, ready flag also is cleared.

IRQ[1]	[1]	NAND Flash Program Interrupt Request
1	Read	Program Interrupt is finished.
1	Write	Program Interrupt Request Clear

When this register is cleared, program flag also is cleared.

IRQ[0]	[0]	NAND Flash Read Interrupt Request
1	Read	Read Interrupt is finished.
1	Write	Read Interrupt Request Clear

When this register is cleared, read flag also is cleared.

Controller Reset Register (NFC_RST) **0xF05B0064**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Don't care															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Don't care															

When this register is written by any value, Controller and Register is reset.

NAND Flash Control Register 1(NFC_CTRL1) 0xF05B0068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACK	ARB				IDL	DNUM[1:0]		rSTP[3:0]				rPW[3:0]			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
rHLD[3:0]				wSTP[3:0]				wPW[3:0]				wHLD[3:0]			

DACK	[31]	DMA Acknowledge Selection
0	DMA Acknowledge Disable For burst data transfer by ARM, this register is low.	
1	DMA Acknowledge Enable	

ARB	[30]	Burst Arbitration Enable
0	Burst Arbitration Disable.	
1	Burst Arbitration Enable.	

If ARB bit is set to low, NFC does not release DMA request signal until the end of all data transfer, so it becomes a deadlock condition that do not allows other DMA request to be granted. In case ARB bit is set to high, DMA request signal of NFC is asserted just before every HOP transfers occurs and deasserted other time, not to make deadlock condition.

IDL	[26]	NFC IDLE State Flag
0	NFC Active State(Read Only)	
1	NFC IDLE State (Read Only)	

DNUM[01:00]	[25:24]	NFC MDATA Number
N	N = NFC_MDATA Access Number	

Sets the number of multiple data transfer in the unit of bytes or half-word(refer to Figure 6.9 and Figure 6.10).

For 8 bit bus width of NAND flash, transferred bytes = DNUM + 1

For 16 bit bus width of NAND flash, transferred half word = DNUM + 1 (must be DNUM<2)

rSTP[3:0]	[23:20]	Number of Read Cycle for Setup Time
N (= 0~15)	rSTP = N	

rPW[3:0]	[19:16]	Number of Read Cycle for Pulse Width
N (= 0~15)	rPW = N	

rHLD[3:0]	[15:12]	Number of Read Cycle for Hold Time
N (= 0~15)	rHLD = N	

wSTP[3:0]	[11:08]	Number of Write Cycle for Setup Time
N (= 0~15)	wSTP = N	

wPW[3:0]	[07:04]	Number of Write Cycle for Pulse Width
N (= 0~15)	wPW = N	

wHLD[3:0]	[03:00]	Number of Write Cycle for Hold Time
N (= 0~15)	wHLD = N	

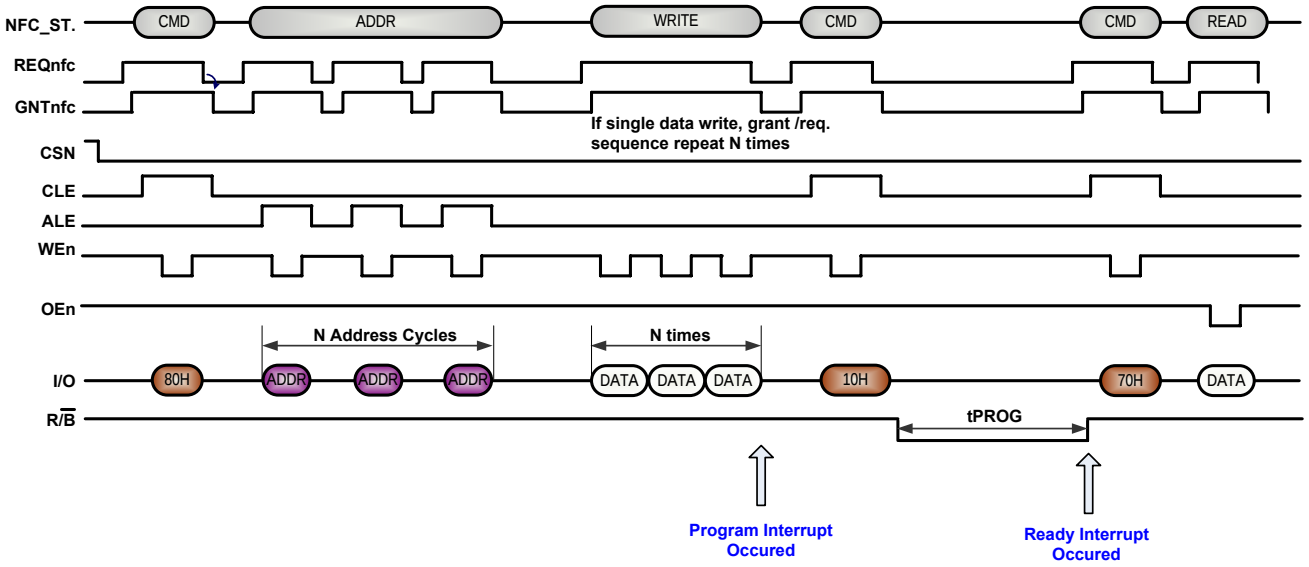


Figure 6.5 Timing Diagram of NAND Page Program

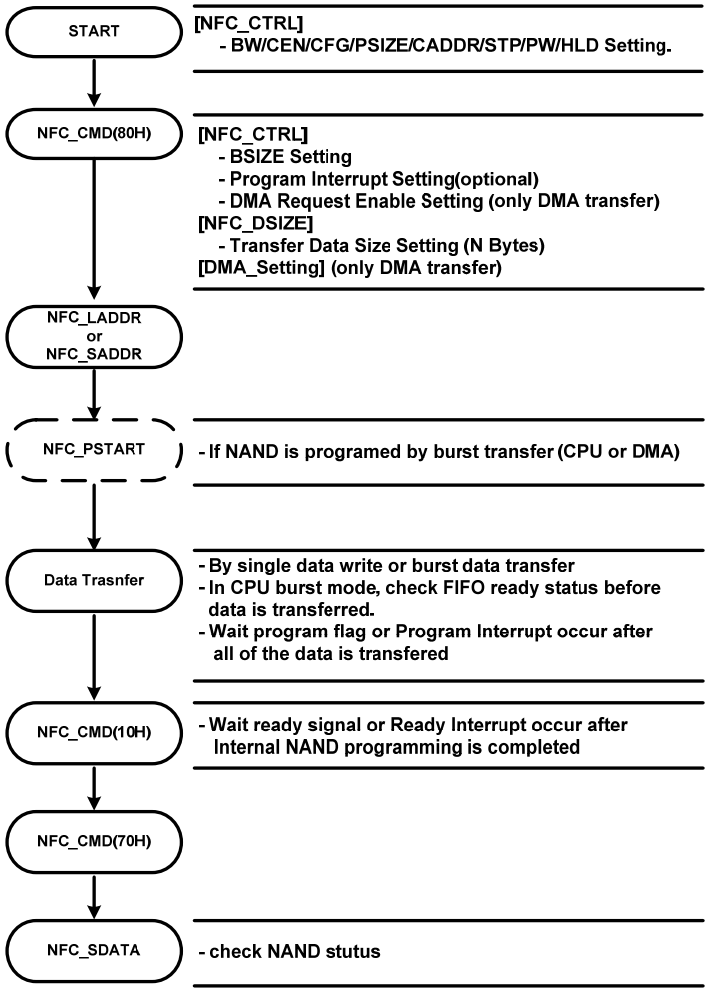


Figure 6.6 A Sequence of NAND Page Program

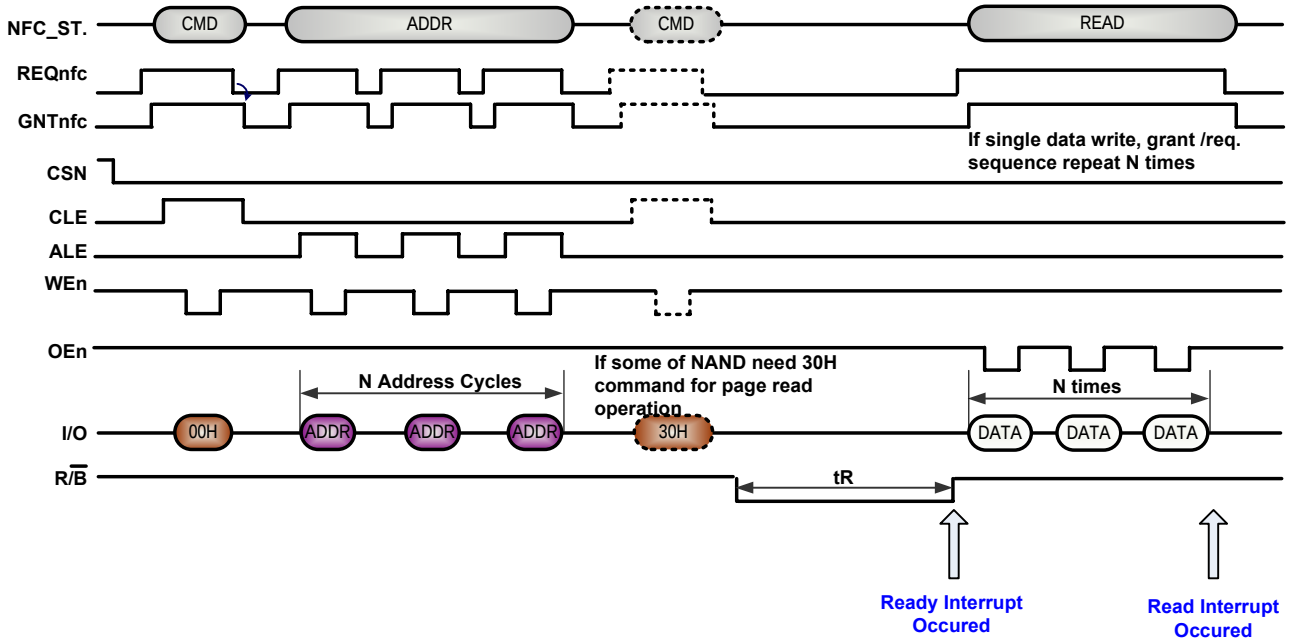


Figure 6.7 Timing Diagram of NAND Page Read

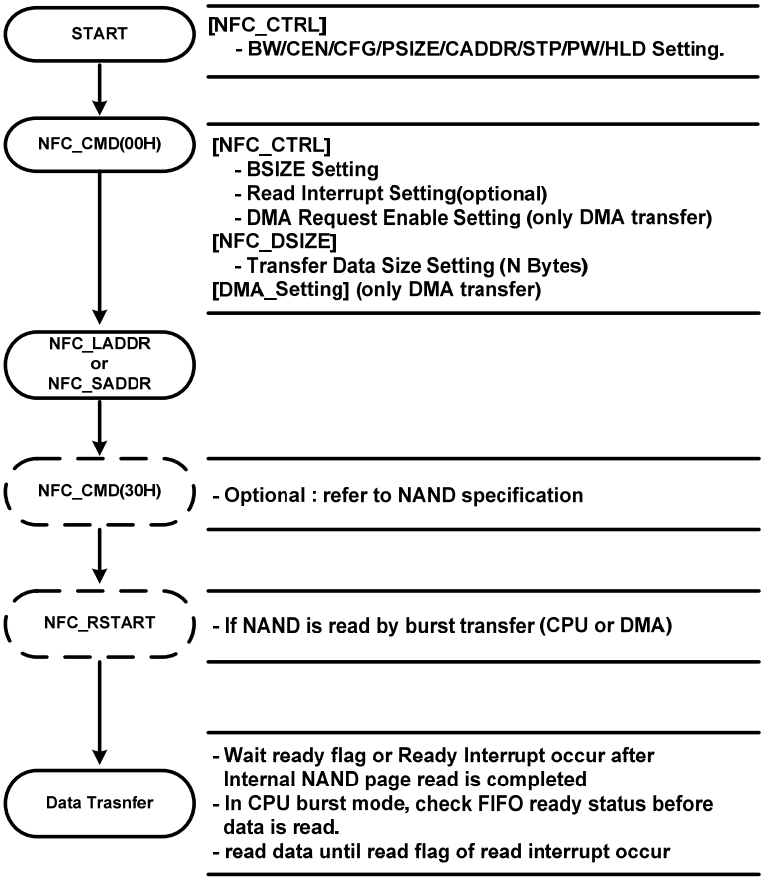


Figure 6.8 A Sequence of NAND Page Read

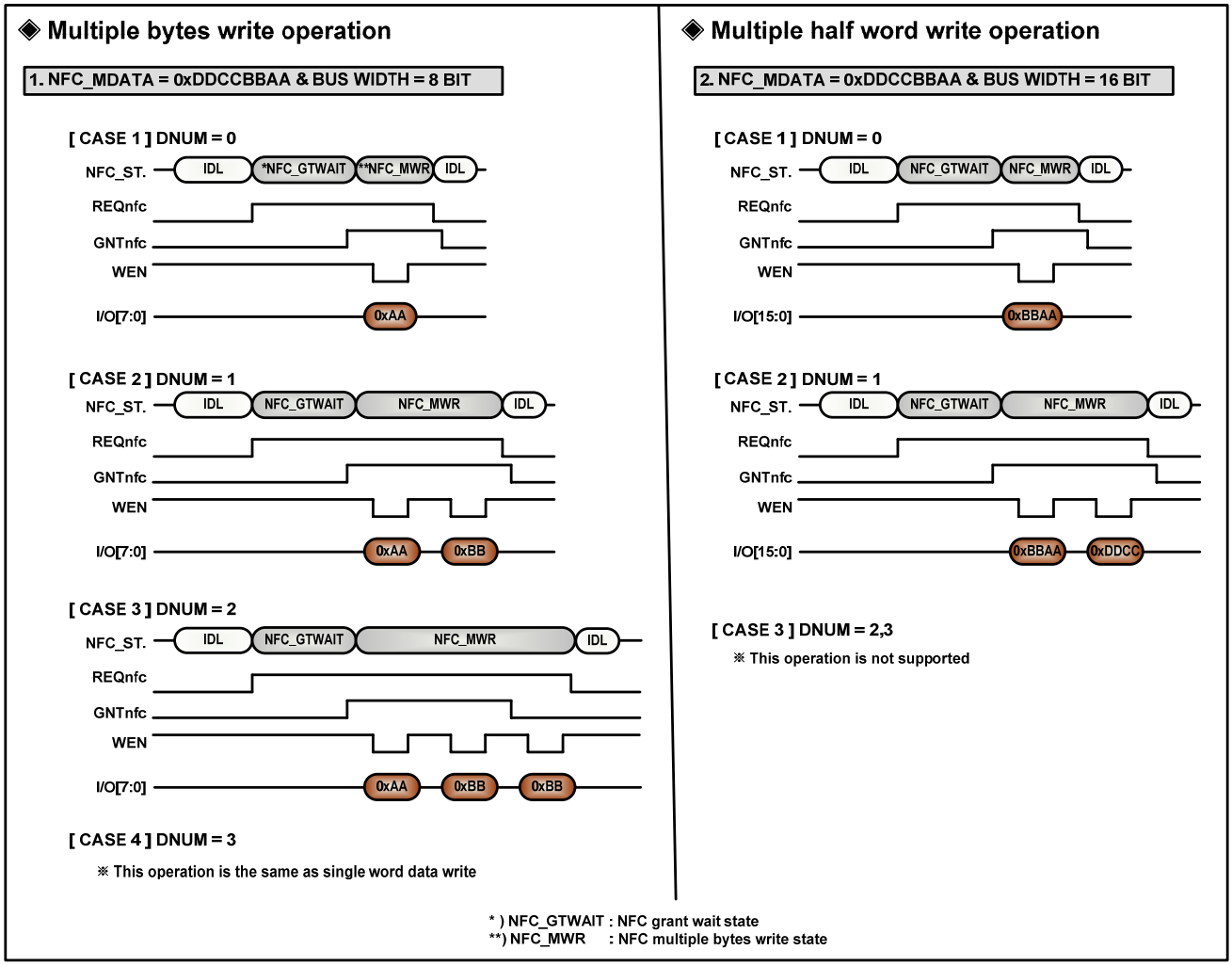


Figure 6.9 Example of Multiple Bytes/Half Word Write Operation

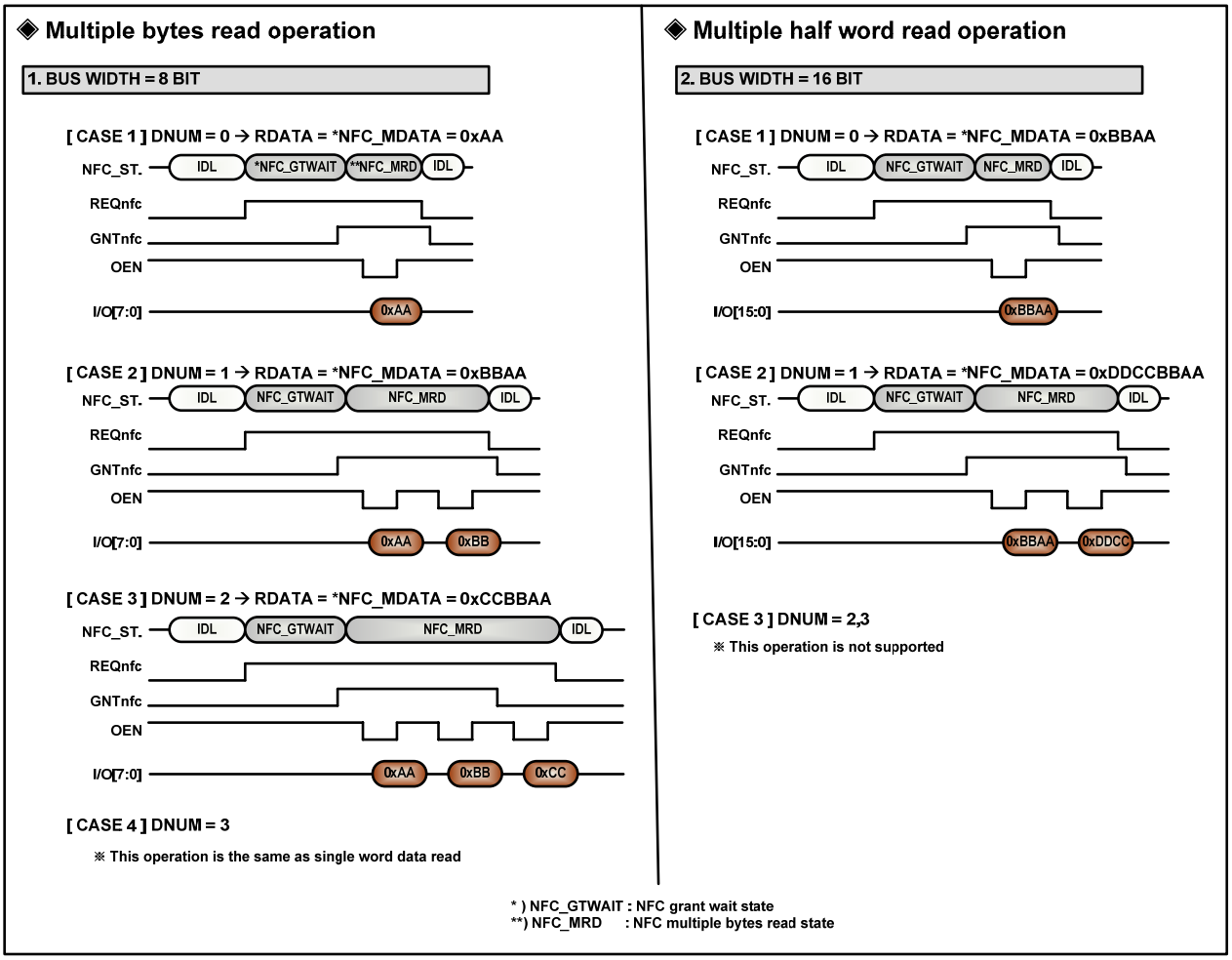


Figure 6.10 Example of Multiple Bytes/Half Word Read Operation

7 Static Memory Controller(SMC)

7.1 Overview

The TCC8900 has a memory controller for various kind of static memory for digital media en-decoding system. It can manipulate NOR type Flash, ROM, SRAM type memories. These memories are selected by nCS3 ~ nCS0 pins. The data bus width can be configured for each chip select separately. The cycle parameter for accessing external memory can be configured by internal registers.

- 4 nCS for a external static memory and 1 nCS for IDE secondary nCS(nCS_IDE)
- 8 / 16 bit configurable data bus width in each chip select
- Normal / Wrapping / Indirect address mode
- The external static memory of SRAM, ROM and IDE Type is supportable

Figure 7.1 shows a block diagram of SMC. There is 16-depth wirte buffer that helps SMC core with reading from or writing to external static memory.

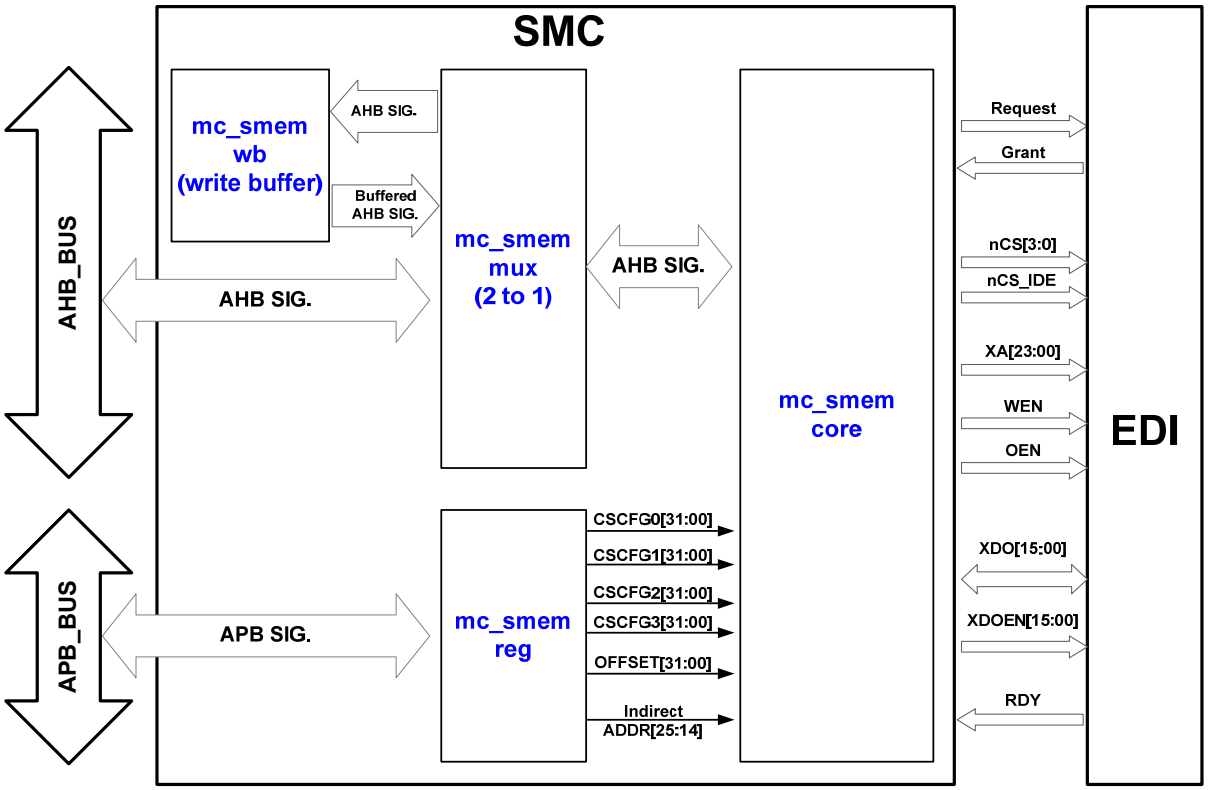


Figure 7.1 SMC Block Diagram

As shown in Figure 7.2, SMC interface signals are connected to external static memory through EDI block. For that reason, EDI configuration & GPIO(A, B and F) function selection should be done, prior to access to external devices. Figure 7.2 shows the default connection from SMC to external memory. The connection can be changed by the configuration of EDI_CSNCFG0 or EDI_CSNCFG1 register, Refer to EDI for the detail description

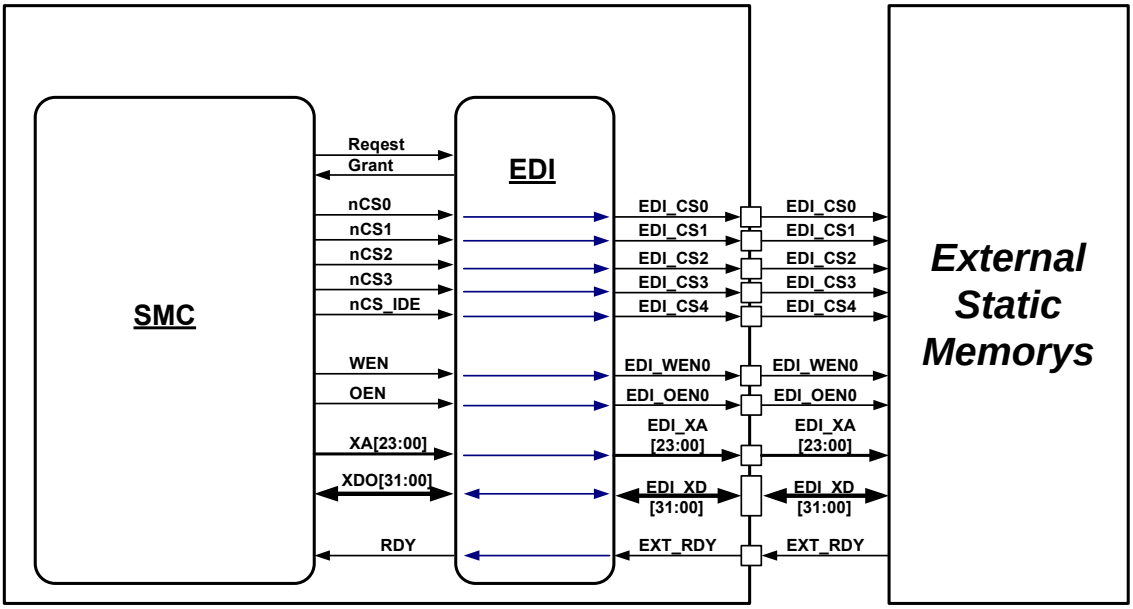


Figure 7.2 The Interface from SMC to External Static Memory in Default Configuration

The valid bit width of SMC address(XA[23:0]) varies, according to address mode.

Table 7.1 shows detailed relationships between internal AHB address and external memory address according to address mode.

Table 7.1 The Address Mapping for Each Address Mode

MODE	BW	Internal	External
NORMAL	8 BIT	HADDR[13:00]	XA[13:00]
	16 BIT	HADDR[13:01]	XA[12:00]
INDIRECT	8 BIT	{ INDIRADDR[23:14],HADDR[13:00] }	XA[23:00]
	16 BIT	{ INDIRADDR[24:14],HADDR[13:01] }	XA[23:00]
WRAPPING	8 BIT	HADDR [CSNOFFSET+2:CSNOFFSET]	XA[02:00]
	16 BIT	HADDR [CSNOFFSET+3:CSNOFFSET+1]	XA[02:00]

There are four global chip select signals and one dedicated secondary IDE chip select signal. The base address mapping to access to external static memory is shown in Table 7.2.

Table 7.2 The Base Address for Each Chip Select

MODE	BASE ADDRESS	OFFSET	ETC
nCS0	0xF05C0000	0x0000 ~ 0X3FFF	If any nCS is configured to IDE Type, nCS_IDE is used for the second CS in this type
nCS1	0xF05C4000		
nCS2	0xF05C8000		
nCS3	0xF05CC0000		

7.2 Register Description

Table 7.3 Memory Controller Register Map (Base Address = 0xF05F0000)

Name	Address	Type	Reset	Description
STATUS	0x00	R/W	Unknown	Status Register
CSNCFG0	0x20	R	0x4b40_3183	External Chip Select0 Config Register
CSNCFG1	0x24	R/W	0x4b40_1104	External Chip Select1 Config Register
CSNCFG2	0x28	W	0x4b40_4082	External Chip Select2 Config Register
CSNCFG3	0x2C	R/W	0x4b40_20C5	External Chip Select3 Config. Register
CSNOFFSET	0x30	R/W	0x0	Wapping Address Mode OFFSET Register
INDIRADDR	0x34	R/W	0x0	Indirect Address

Status Register (STATUS) **0xF05F0000 + 0x00**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										CST					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
															RDY

Field	Name	RW	Reset	Description
21 ~ 16	CST	RW	0	SMC Current Status 6'b00_0001 : ST_IDLE 6'b00_0010 : ST_WAIT … 6'b01_0000 : ST_WR0 6'b01_0001 : ST_WR1 6'b01_0010 : ST_WR2 6'b01_0100 : ST_WR3 6'b10_0000 : ST_RD0 6'b10_0001 : ST_RD1 6'b10_0010 : ST_RD2 6'b10_0100 : ST_RD3
0	RDY	RW	0	External Ready Input Status

In indirect address mode, indirect address register value should be changed when the CST is in ST_IDLE state, otherwise unexpected address could be generated.

External Chip Select Configuration Register (CSCFGn, n=0,1,2,3) 0xF05F000020 + (n*4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PTnOE	PTnWE	AM		MSIZE		MTYPE		URDY	RDY	Reserved					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STP				PW								HLD			

PTnOE[31]	nOE Pulse Type
0	STP and HLD timming parameter values would be applied as set below
1	When STP and HLD paramter values are all zeros, HLD is forced to be 1, corresponding to OE signal.

Default value: 0x0 for all of the CS.

PTnWE[30]	nWE Pulse Type
0	STP and HLD timming parameter values would be applied as set below
1	When STP and HLD paramter values are all zeros, HLD is forced to be 1, corresponding to WE signal.

Default value: 0x1 for all of the CS.

AM[29:28]	Address Mode
0	Normal address mode. HADDR[13:0] is only available memory address. others are ignored.
1	Wrapping address mode
2,3	Indirect address mode

MSIZE[27:26]	Memory Size
0, 1	Bus width = 32 bit
2	Bus width = 16 bit
3	Bus width = 8 bit

Default value : 16bit bus width (all of the CSn).

MTYPE [25:24]	Type of External Memory
0	Reserved.
1	IDE type
2	SMEM_0 type (Ex : ROM, NOR flash) Byte-write control signal (DQM) is not needed.
3	SMEM_1 type (Ex : SRAM) Byte-write control signal (DQM) is needed.

URDY [23]	Use Ready
1	Ready / Busy signal monitoring is enabled The memory controller extends access cycle until the state of READY pin indicates that the access request has accomplished.

Note : Default value : 0x0 (all of the CSn).

RDY [22]	Ready / Busy Select
0	The READY pin indicates the READY signal. The memory controller extends access cycle until this pin goes to high state.
1	The READY pin indicates the BUSY signal. The memory controller extends access cycle until this pin goes to low state.

Note : Refer to Figure 7.3 for ready/busy cycle extension.

Note : Default Value : 0x1 for all of the CS

STP [15:12]	Number of Cycle for Setup Time (tSH)
N	N cycle is issued between the falling edge of nCS[n] and nOE / nWE.

PW[11:4]	Number of Cycle for Pulse Width (tPW)

N (= 0~255)	(N+1) cycle is issued between the falling and rising edge of nOE / nWE.

HLD [3:0]	Number of Cycle for Hold Time (tHLD)
N	N cycle is issued between the rising edge of nOE / nWE and nCS[n].

The following figure displays the element cycle diagram for external memories.

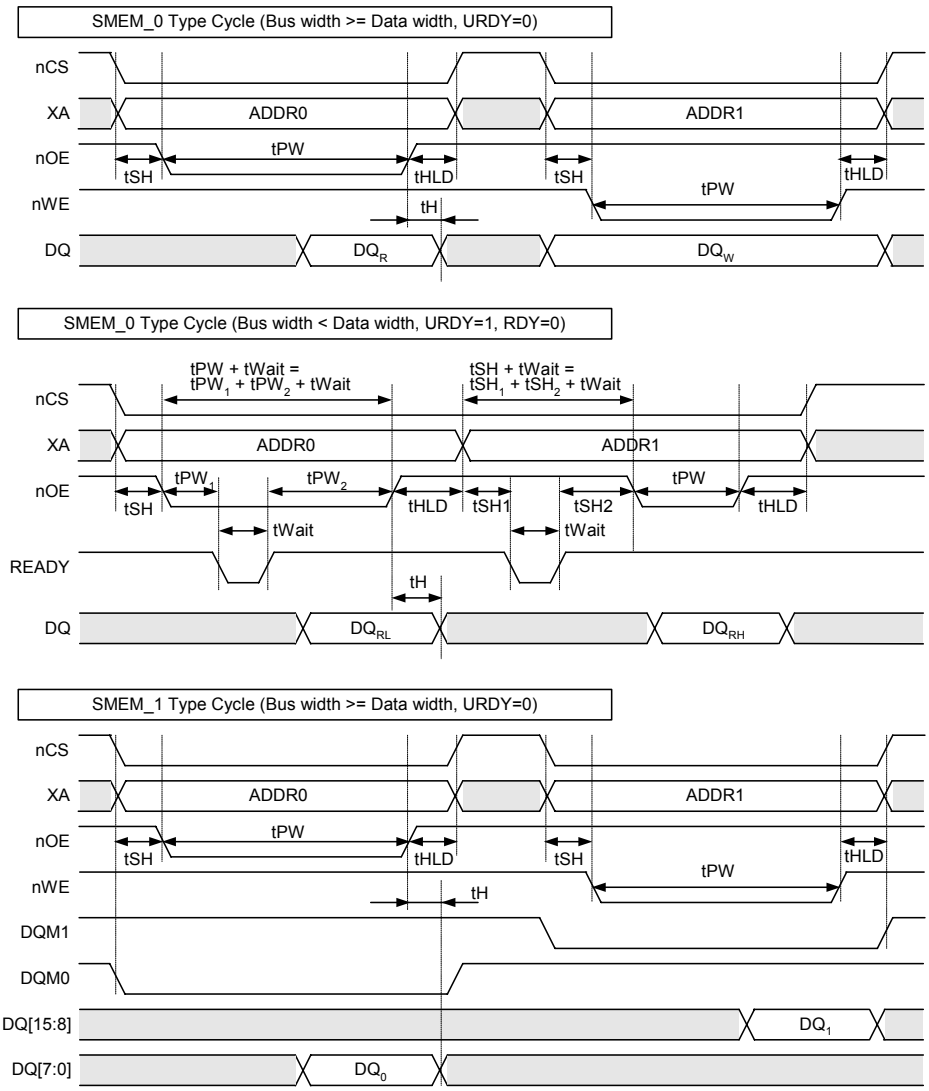


Figure 7.3 Basic Timing Diagram For External Memories

In case of IDE type memory, there are two chip-enable signals for it. In the TCC8900, each enable signal can be controlled by offset address space. 'nCS0' reflects that the offset address range of 0 ~ 0x1F is accessed, 'nCS1' reflects that 0x20 ~ 0x3F is accessed. For larger address than 0x3F, bit5 of address value means which enable signal is activated. (0 to 'nCS0', 1 to 'nCS1')

CSN OFFSET Regiser (CSNOFFSET) **0xF05F000000 + 0x30**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				CS3_OFFSET								CS2_OFFSET			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				CS1_OFFSET								CS0_OFFSET			

Note : In wrapping address mode, CSn_OFFSET determines the number of address bit to be discarded, from least significant bit.

For the detail description, refert to section Wrapping Address Mode.

Indirect Address Register **0xF05F000000 + 0x30**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							INDIRADDR[25:14]								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		0													

Note : In indirect address mode, internal AHB address[23:14] will be replaced with INDIRADDR[25:14].

For the detail description, refer to section Indirect Address Mode

7.3 Address Mode of SMC

7.3.1 Normal Address mode

If CSNCFGn[AM] is “0”, it is normal address mode.

Internal AHB address[31:14] is used for selecting external chip select signal and XA[23:14] is set to 0.

Internal AHB address[13:0] is used for selecting external address output. In 8-bit bus width, address mapping to external address output is bit by bit but in 16-bit width, internal AHB address[13:1] is linearly mapped into XA[12:0].

When the type is SMEM1, XA[23:22] is used as DQM[1:0].

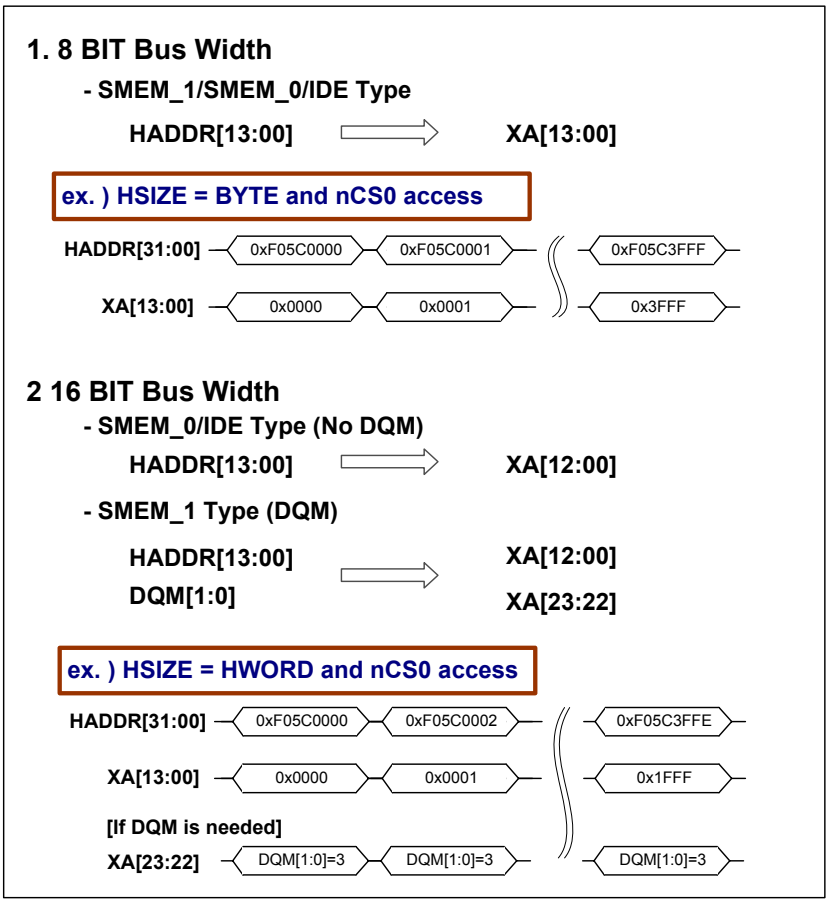


Figure 7.4 Example of Normal Address Mode

7.3.2 Indirect Address Mode

If CSNCFGn[AM] is “2 or 3”, it is indirect address mode. In this mode, address mapping method is same as in normal address mode except that internal AHB address[23:14] is replaced with INDIRADDR[25:14].

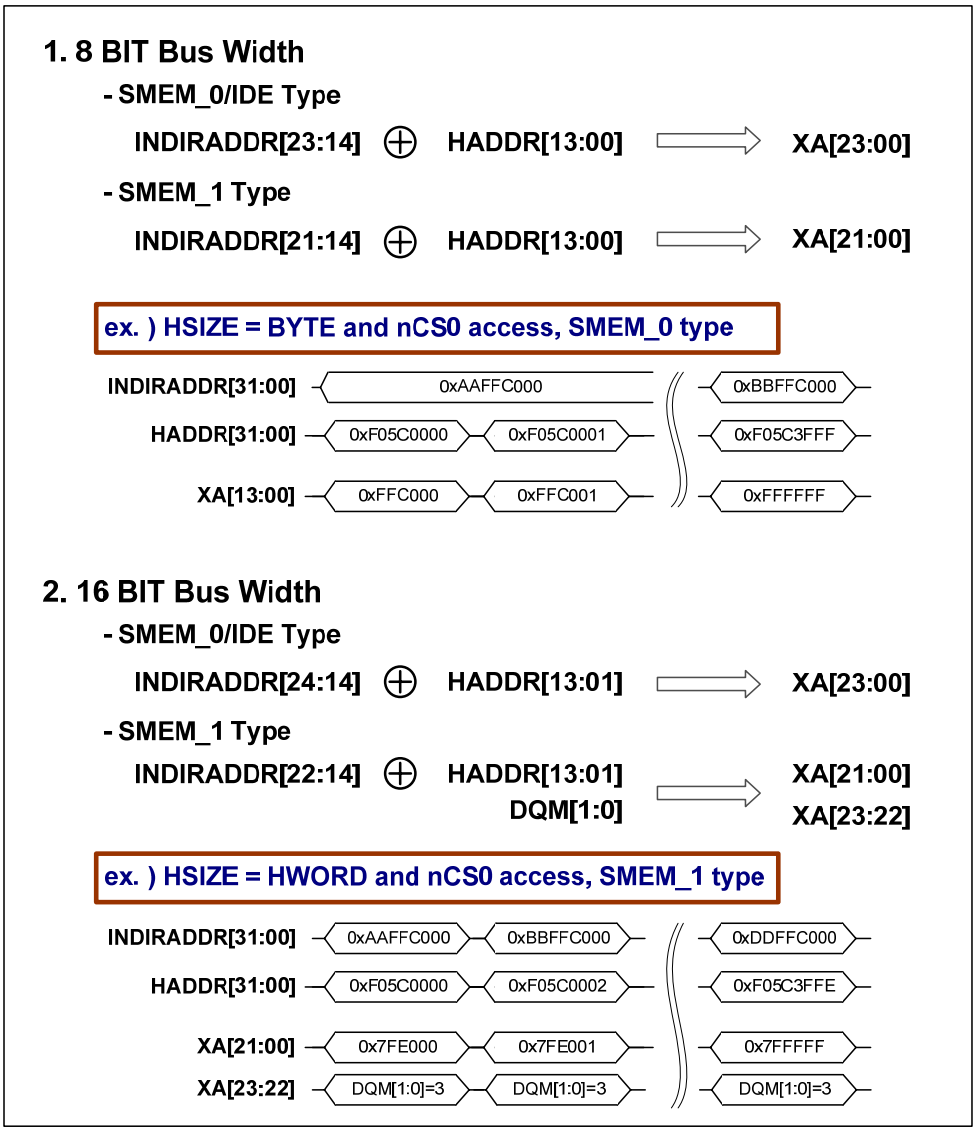


Figure 7.5 Example of Indirect Address Mode

7.3.3 Wrapping Address Mode

If CSNCFGn[AM] is “1”, it is wrapping address mode.

Only 3bits of external address output(XA[2:0]) are valid and other bits (XA[23:3]) are set to 0.

The three output address bits , to be mapped into XA[2:0], are selected from the CSnOFFSET position out of Internal AHB address.

When the type is SMEM1, XA[23:22] is used as DQM[1:0].

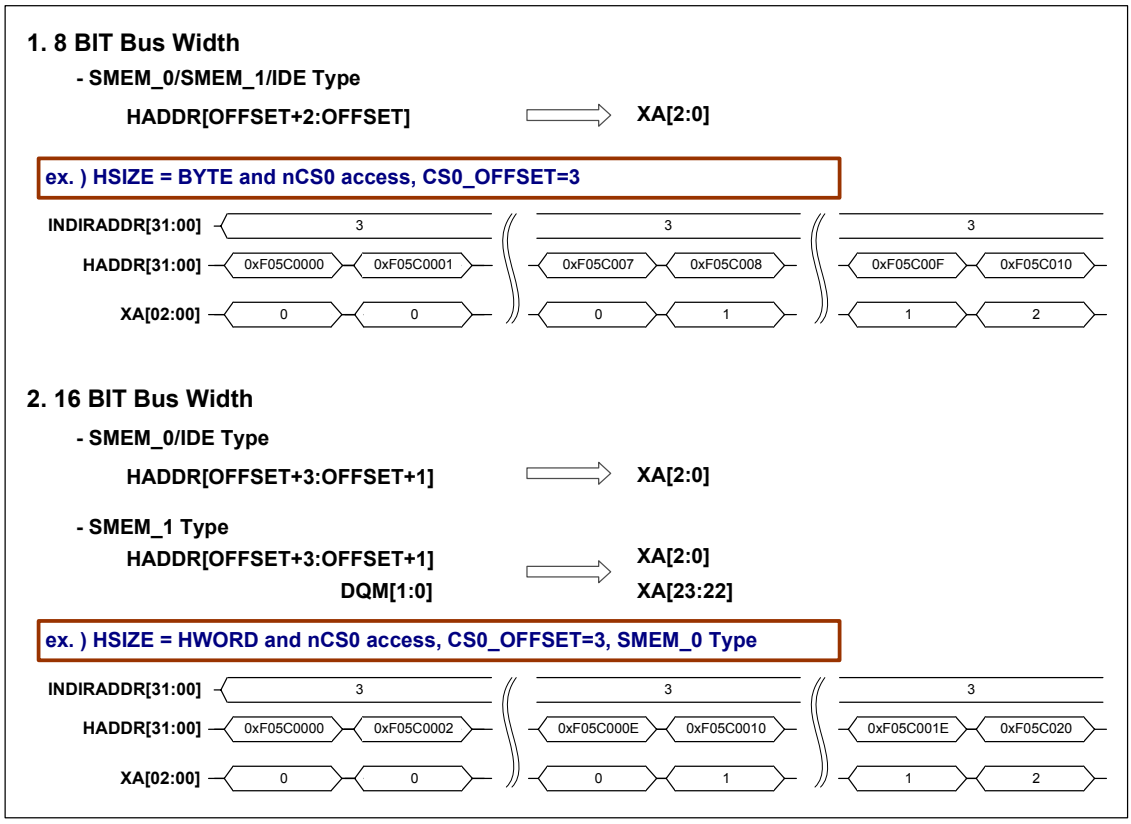


Figure 7.6 Example of Wrapping Address Mode

8 EDI (External Device Interface)

8.1 Overview

The TCC8900 EDI(External Device Interface) makes it possible to share common output port, used for control, address and data signal, and also arbitrates the port requests from SMC and NFC. The arbitration policy is, by default, Round-Robin and optionally can be change to Fixed-Priority.

The supportable main feasture is as follows

- Channel Arbitration for SMC and NFC
- 9 input CS(chip select) is configurably mapped to 7 output CS
- Round-Robin and Fixed Priotiry arbitration is supportable

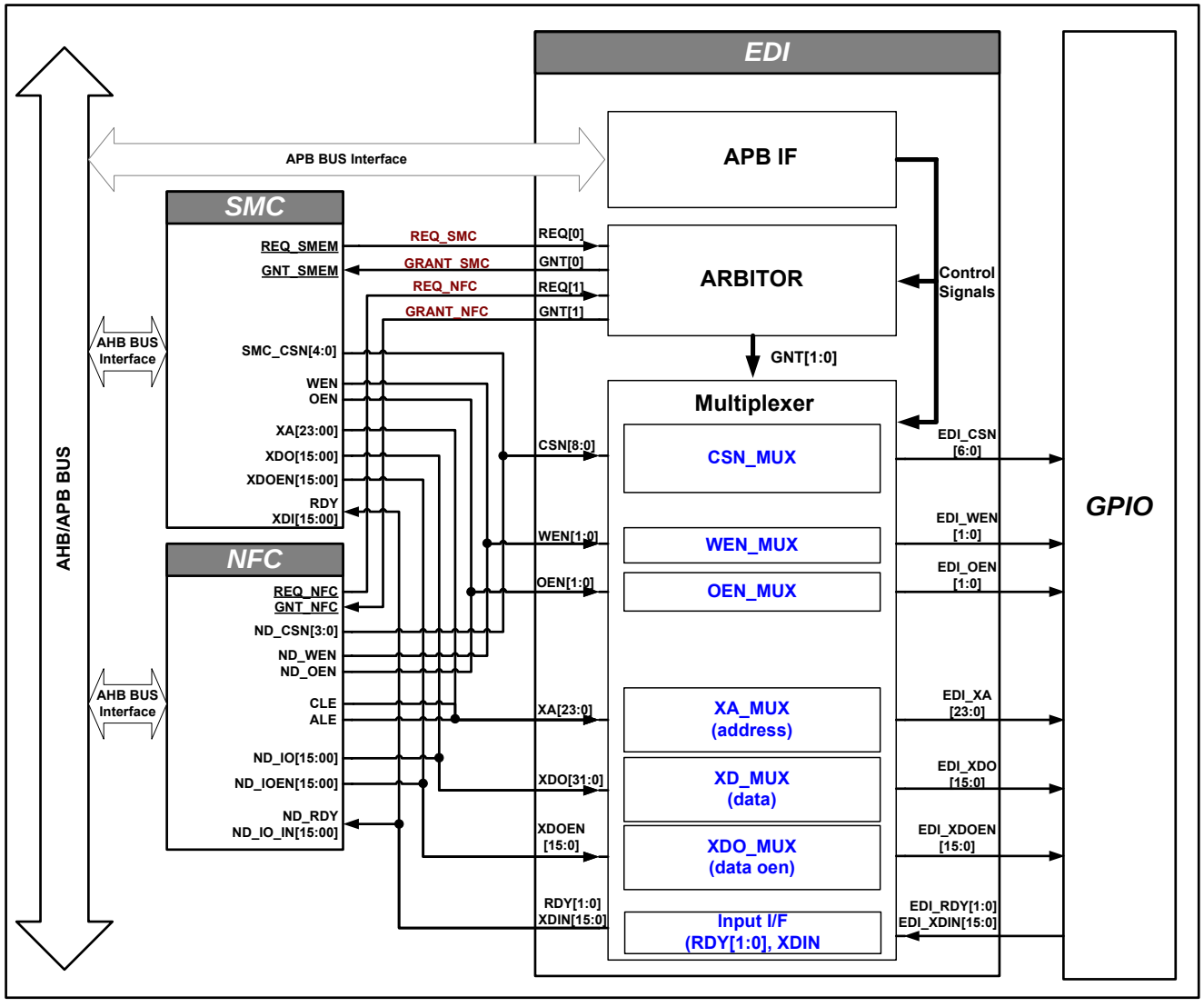


Figure 8.1 Block Diagram and Interface of EDI

The Configuration of GPIO Function for EDI shows connections between controllers and GPIO in reset default configuration by EDI. Through boot procedure, the some of the configuration can be internally changed. Refer to notes for NFC_CSN[0], NFC_RDY, CSN[3]. Where GPIO_B[27], GPIO_B[30] and GPIO_B[31] can be used as EDI_XA or other function port, according to the GPIO_B function selection result.

Table 8.1 The Configuration of GPIO Function for EDI

GPIO	I/O Function				Default connection with Internal Block		ETC.
	Function 0	CFG. Value	Function 1	CFG Value	SMC	NFC	
GPIOB[00]	EDI_XD[08]	1			XD[00]	ND_IO[00]	
GPIOB[01]	EDI_XD[09]	1			XD[01]	ND_IO[01]	
GPIOB[02]	EDI_XD[10]	1			XD[02]	ND_IO[02]	
GPIOB[03]	EDI_XD[11]	1			XD[03]	ND_IO[03]	
GPIOB[04]	EDI_XD[04]	1			XD[04]	ND_IO[04]	
GPIOB[05]	EDI_XD[05]	1			XD[05]	ND_IO[05]	
GPIOB[06]	EDI_XD[06]	1			XD[06]	ND_IO[06]	
GPIOB[07]	EDI_XD[07]	1			XD[07]	ND_IO[07]	
GPIOB[08]	EDI_XD[00]	1			XD[08]	ND_IO[08]	
GPIOB[09]	EDI_XD[01]	1			XD[09]	ND_IO[09]	
GPIOB[10]	EDI_XD[02]	1			XD[10]	ND_IO[10]	
GPIOB[11]	EDI_XD[03]	1			XD[11]	ND_IO[11]	
GPIOB[12]	EDI_XD[12]	1			XD[12]	ND_IO[12]	
GPIOB[13]	EDI_XD[13]	1			XD[13]	ND_IO[13]	
GPIOB[14]	EDI_XD[14]	1			XD[14]	ND_IO[14]	
GPIOB[15]	EDI_XD[15]	1			XD[15]	ND_IO[15]	
GPIOB[16]	EDI_WEN[0]	1			WEN		
GPIOB[17]	EDI_WEN[1]	1				ND_WEN	
GPIOB[18]	EDI_OEN[0]	1			OEN		
GPIOB[19]	EDI_OEN[1]	1				ND_OEN	
GPIOB[20]	EDI_XA[00]	1			XA[00]	ND_CLE	
GPIOB[21]	EDI_XA[01]	1			XA[01]	ND_ALE	
GPIOB[22]	EDI_XA[02]	1			XA[02]		
GPIOB[23]	EDI_CSN[0]	1			CSN[0]		refer to note (1)
GPIOB[24]	EDI_CSN[1]	1			CSN[1]		
GPIOB[25]	EDI_CSN[2]	1			CSN[2]		
GPIOB[26]	EDI_CSN[3]	1			CSN[3]		refer to note (3)
GPIOB[27]	EDI_CSN[4]	1	EDI_XA[23]	6	CSN[4] / XA[23]		
GPIOB[28]	EDI_RDY[0]	1			RDY		refer to note (2)
GPIOB[29]	EDI_RDY[1]	1				ND_RDY	refer to note (2)
GPIOB[30]	EDI_CSN[5]	1	EDI_XA[22]	6	XA[22]	ND_CSN[0]	refer to note (1)
GPIOB[31]	EDI_CSN[6]	1	EDI_XA[21]	6	XA[21]	ND_CSN[1]	
GPIOF[00]	EDI_XA[03]	6			XA[03]		
GPIOF[01]	EDI_XA[04]	6			XA[04]		
GPIOF[02]	EDI_XA[05]	6			XA[05]		
GPIOF[03]	EDI_XA[06]	6			XA[06]		
GPIOF[04]	EDI_XA[07]	6			XA[07]		
GPIOF[05]	EDI_XA[08]	6			XA[08]		
GPIOF[06]	EDI_XA[09]	6			XA[09]		
GPIOF[07]	EDI_XA[10]	6			XA[10]		
GPIOF[08]	EDI_XA[11]	6			XA[11]		
GPIOF[09]	EDI_XA[12]	6			XA[12]		
GPIOF[10]	EDI_XA[13]	6			XA[13]		
GPIOF[11]	EDI_XA[14]	6			XA[14]		
GPIOF[12]	EDI_XA[15]	6			XA[15]		
GPIOF[13]	EDI_XA[16]	6			XA[16]		
GPIOF[14]	EDI_XA[17]	6			XA[17]		
GPIOF[15]	EDI_XA[18]	6			XA[18]		
GPIOA[06]	EDI_XA[19]	6			XA[19]		
GPIOA[07]	EDI_XA[20]	6			XA[20]		

Note :
(1) In the NAND boot mode, ND_CSN[0] is remapped to EDI_CSN[0] and should be connected to external boot device.
(2) In the NAND boot mode, ND_RDY is remapped to EDI_RDY[0] and should be connected to external boot device.
(3) In the NOR boot mode, EDI_CSN[3]_should be connected to external boot device..

8.2 Registers Description

Table 8.2 EDI Register Map (BASE ADDRESS : 0xF05F6000)

Name	Offset	Type	Reset	Description
EDI_CTRL	0x00	R/W	0x00000000	EDI Control Register.
EDI_CSNCFG0	0x04	R/W	0x00543210	EDI CSN Configuration Register 0.
EDI_CSNCFG1	0x08	R/W	0x00BA9876	EDI CSN Configuration Register 1.
Reserved	0x0C	R/W	-	-
Reserved	0x10	R/W	-	-
EDI_RDYCFG	0x14	R/W	0x76543210	EDI Ready Configuration Register
Reserved	0x18	R/W	0x00000000	EDI Time-Out Configuration Register 0
Reserved	0x1C	R/W	0x00000000	EDI Time-Out Configuration Register 1
EDI_REQOFF	0x20	R/W	0x00000000	EDI Request OFF Flag Register

EDI Control Register (EDI_CTRL) 0xF05F6000 + 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0				R*			AM	0			PRIORITY[4:0]				

BIT	NAME	R/W	RESET	DESCRIPTION
8	AM	R/W	Refer to Table 8.2	EDI arbitration mode selection 0 : Round-Robine arbitration mode 1 : Fixed-Priority arbitration mode
4 : 0	Priority	R/W		EDI Fixed-Priority Configuration 5'b00000 : CH0 → CH1 5'b00110 : CH1 → CH0

EDI CSN Configuration Register 0 (EDI_CSNCFG0) 0xF05F6000 + 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				0				CFGCS5				CFGCS4			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CFGCS3				CFGCS2				CFGCS1				CFGCS0			

Refer to Table 8.3 and Figure 8.2

Table 8.3 EDI_CSNCFG0[CFGCSn] Configuration

CFGCSn (n=0,…,5)	CSN Cofiguration
0000	Input CSN[0] of EDI is connected to EDI_CSN[n]
0001	Input CSN[1] of EDI is connected to EDI_CSN[n]
0010	Input CSN[2] of EDI is connected to EDI_CSN[n]
0011	Input CSN[3] of EDI is connected to EDI_CSN[n]
0100	Input CSN[4] of EDI is connected to EDI_CSN[n]
0101	Input CSN[5] of EDI is connected to EDI_CSN[n]
0110	Input CSN[6] of EDI is connected to EDI_CSN[n]
0111	Input CSN[7] of EDI is connected to EDI_CSN[n]
1000	Input CSN[8] of EDI is connected to EDI_CSN[n]
o/w	Reserved

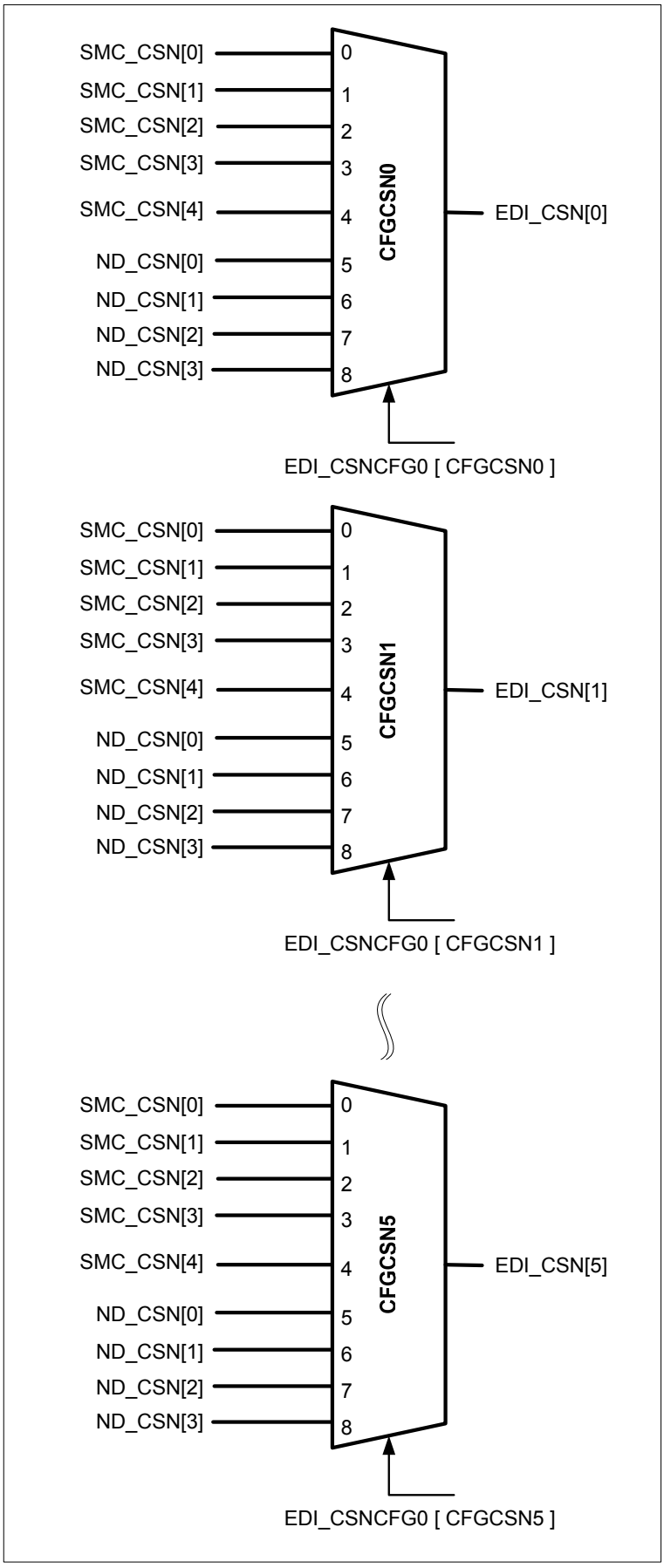


Figure 8.2 Block Diagram for EDI_CSN[5:0]

EDI CSN Configuration Register 1 (EDI_CSNCFG1) **0xF05F6000 + 0x08**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				0				Reserved				Reserved			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				Reserved				Reserved				CFGCS6			

Refer to Table 8.4 and Figure 8.3

Table 8.4 EDI_CSNCFG1[CFGCSn] Configuration

CFGCS6	CSN Cofiguration
0000	Input CSN[0] of EDI is connected to EDI_CSN[6]
0001	Input CSN[1] of EDI is connected to EDI_CSN[6]
0010	Input CSN[2] of EDI is connected to EDI_CSN[6]
0011	Input CSN[3] of EDI is connected to EDI_CSN[6]
0100	Input CSN[4] of EDI is connected to EDI_CSN[6]
0101	Input CSN[5] of EDI is connected to EDI_CSN[6]
0110	Input CSN[6] of EDI is connected to EDI_CSN[6]
0111	Input CSN[7] of EDI is connected to EDI_CSN[6]
1000	Input CSN[8] of EDI is connected to EDI_CSN[6]
o/w	Reserved

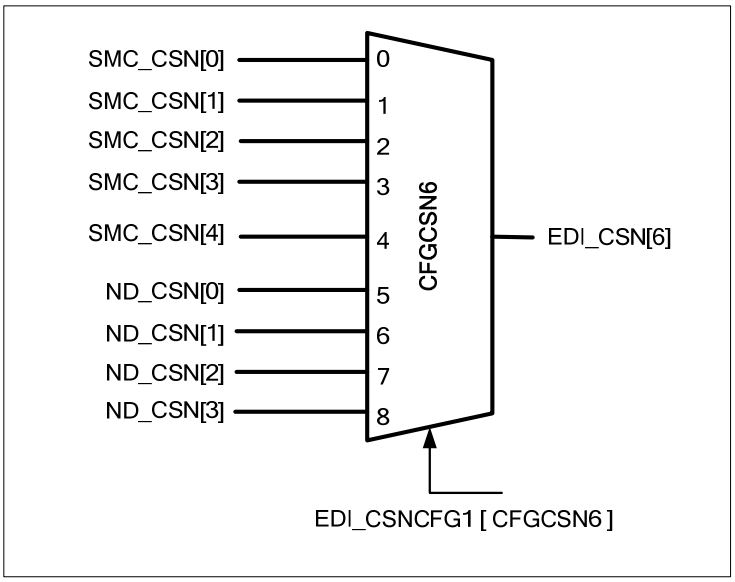


Figure 8.3 Block Diagram for EDI_CSN[6]

EDI READY Configuration Register 0 (EDI_RDYCFG) **0xF05F6000 + 0x14**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	Reserved			0	Reserved			0	Reserved			0	Reserved.		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	Reserved			0	Reserved			0	CFGRDY1			0	CFGRDY0		

Refer to Table 8.5 and Figure 8.4

Table 8.5 CFGRDYn Configuration

CFGRDY0	RESET	CSN Cofiguration
000	000	Input EDI_RDY[0] of EDI is connected to RDY[0]
001		Input EDI_RDY[1] of EDI is connected to RDY[0]
010 ~ 111	Reserved.	

CFGRDY1	RESET	CSN Cofiguration
000	000	Input EDI_RDY[0] of EDI is connected to RDY[1]
001		Input EDI_RDY[1] of EDI is connected to RDY[1]
010 ~ 111	Reserved.	

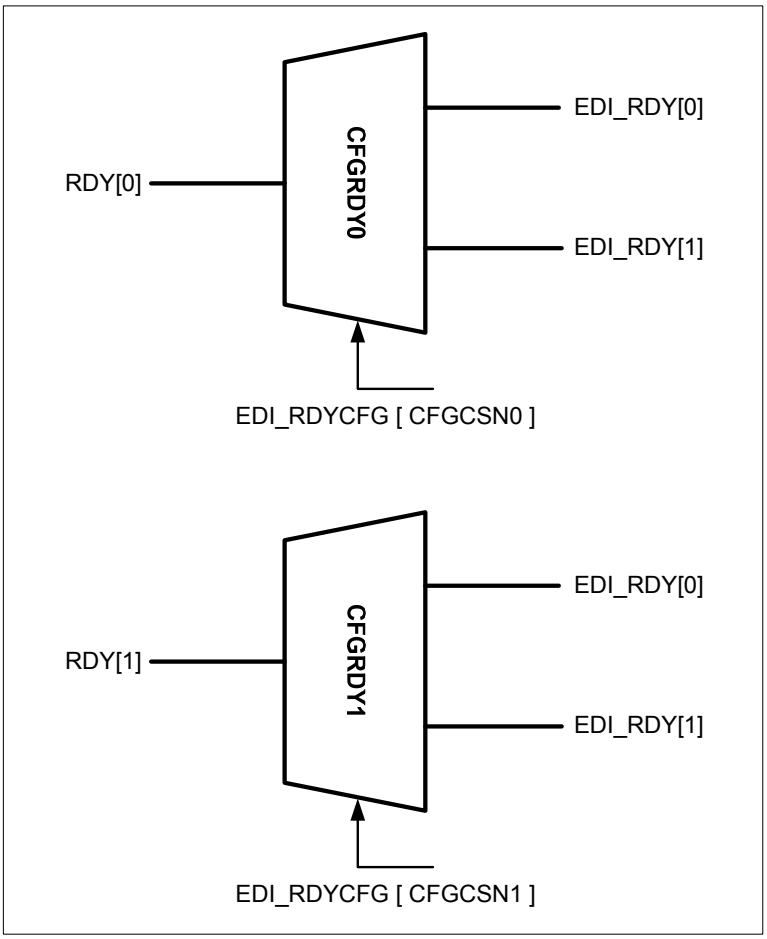


Figure 8.4 Block Diagram for EDI_RDY[1:0]

EDI Request Off Flag Register (EDI_REQOFF) 0xF05F6000 + 0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												Reserved		REQOFF	

This register shows current request pending state of EDI. Namely, when one channel(ch0:SMC, ch1:NFC) has acquired grant for access to EDI and started read/write operation, the request state of the other channel is reflected in this register.

BIT	NAME	R/W	RESET	DESCRIPTION
1:0	REQOFF	RO	Refer to Table 8.2	2'b00 : o/w 2'b01 : SMC is granted by EDI and NFC is requesting for grant to EDI. 2'b10 : NFC is granted by EDI and SMC is Requesting for grant to EDI 2'b11 : -

9 IDE

9.1 Overview

IDE controller supports PIO mode0,1,2,3,4 and UDMA mode0,1,2,3,4. IDE controller interfaces with ATA/ATAPI compliant devices. In UDMA (Ultra DMA) mode, CPU can receive or transmit data using internal trigger level interrupt but for the better performance, use Internal DMA (IDMA). Internal DMA is one of AHB master. When FIFO Pointers reaches trigger level, and IDE (Internal DMA Enable) bit is set, Internal DMA receive data from source address or transmit the data to target address. When IDE bit is set, CPU can't access data of internal FIFO. When operation is ended, check UDMAINT register. UDMAINT register includes important information (e.g., word aligned, FIFO error, valid data remained in FIFO etc).

The simple block diagram of IDE Controller is as followings.

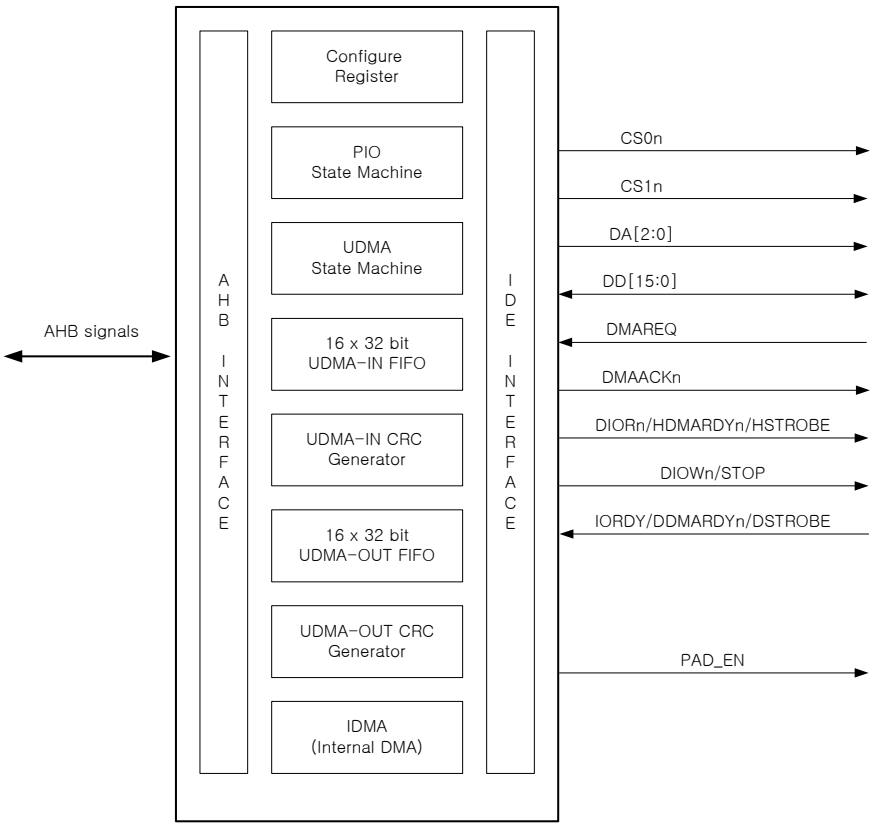


Figure 9.1 IDE Controller Block Diagram

The Maximum theoretical bandwidth of the IDE Interface is shown in Table 9.1.

Table 9.1 Speed of Data Transfer (Byte per Second)

MODE	SPEED	MODE	SPEED
PIO MODE0	3.3 MBps	UDMA MODE 0	16.67 MBps
PIO MODE 1	5.22 MBps	UDMA MODE 1	25 MBps
PIO MODE 2	8.33MBps	UDMA MODE 2	33.33 MBps
PIO MODE 3	11.11MBps	UDMA MODE 3	44.44 MBps
PIO MODE 4	16.67MBps	UDMA MODE 4	66.67 MBps

9.2 Register Description

The overall registers are shown in Table 9.2.

Table 9.2 IDE Registers (Base = 0xF0520000)

Name	Address	Type	Reset	Description
CS0n	0x00~ 0x1F	R/W	-	PIO CS0n Access Register
CS1n	0x20 ~0x3F	R/W	-	PIO CS1n Access Register
PIOCTRL	0x40	R/W	0x00600000	PIO Mode Control Register
UDMACTRL	0x44	R/W	0x00000000	UDMA Mode Control Register
IDMACTRL	0x48	R/W	0x00000000	IDMA Control Register

IDMASA	0x4C	R/W	0x00000000	IDMA Source Address Register
IDMASP	0x50	R/W	0x00000000	IDMA Source Parameter Register
IDMACSA	0x54	R	0x00000000	IDMA Current Source Address Register
IDMADA	0x58	R/W	0x00000000	IDMA Destination Address Register
IDMADP	0x5C	R/W	0x00000000	IDMA Destination Parameter Register
IDMACDA	0x60	R	0x00000000	IDMA Current Destination Address Register
IDEINT	0x64	R/W	0x0000_0000	IDE Interrupt Register
UDMATCNT	0x68	R/W	0x00FF_FFFF	UDMA Transfer Counter Register
UDMAIN	0x6C	R	-	UDMA-IN Access Register
UDMAOUT	0x70	W	-	UDMA-OUT Access register
UDMACRC	0x74	R	0x0000_4ABA	UDMA CRC Register
UDMACTCNT	0x78	R	0x00FF_FFFF	UDMA Current Transfer Counter Register

PIO CS0n Access Register (CS0n) 0xF0520000 ~ 0xF052001F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DD0[15:0]															

Bit	Name	R/W	Reset	Description
31:16			0	Reserved
15:0	DD0	R/W	0	16-bit access data for CS0n

DD0 is access data to read or to be written for CS0n in PIO MODE.

For example, to write 0x1234(data value) to PIO address 0x6 using CS0n, CPU or DMA target address(HADDR[6:0]) must be 0x18 and DD0 must be 0x1234, which means HADDR[4:2] equals DA[2:0] and DD0 equals DD[15:0]. To read data of PIO address 0x7, CPU or DMA target address (HADDR[6:0]) must be 0x1C, which means HADDR[4:2] equals DA[2:0] and DD0 is read data of PIO address 0x7.

PIO CS1n Access Register (CS1n) 0xF0520020 ~ 0xF052003F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DD1[15:0]															

DD1[15:0]	PIO Data for CS0n
DD1	16bit access data for CS1n

DD1 is access data to read or to be written for CS1n in PIO MODE.

To write 0x1234 (data value) to PIO address 0x6 using CS1n, CPU or DMA target address(HADDR[6:0]) must be (0x18 + 0x20) and DD1 must be 0x1234, which means HADDR[4:2] equals DA[2:0] and DD1 equals DD[15:0]. To read data of PIO address 0x7, CPU or DMA target address (HADDR[6:0]) must be (0x1C + 0x20), which means HADDR[4:2] equals DA[4:2] and DD1 is read data of PIO address 0x7.

PIO Mode Control Register (PIOCTRL) 0xF0520040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									SYNC		MD				STP
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STP			PW						HLD						RDY

SYNC[22:21]	Sync Bit Register for IORDY/DDMARDYn
00	Bypass
01	1 SYNC
10	2 SYNC

These bits are needed to synchronize from HDD clock domain to AHB clock domain for IORDY (and DDMARDYn) signal. The default value is 2 sync.

MD[20]	MODE SEL
0	PIO MODE
1	UDMA(IN/OUT) MODE

To Operate PIO mode, clear this bit and to operate UDMA mode, set this bit.

STP[16:13]	Number Of Cycle for Setup Time
STP	(STP value + 1) AHB cycles are issued

These bits define the time requirement from the active CSn to the active DIORn (or DIOWn) in PIO MODE. For further details, see Figure 9.2.

PW[12:7]	Number Of Cycle for Pulse Width
PW	(PW value + 1) AHB cycles are issued

These bits define the time requirement for active duration of the DIORn (or DIOWn) . For further details, See Figure 9.2.

HLD[6:1]	Number Of Cycle for HOLD Time
HLD	(HLD value + 1) AHB cycles are issued

These bits define the time requirement from the negation of the DIORn (or DIOWn) to the negation of the CSn. For further details, see Figure 9.2.

IORDY[0]	IORDY Enable
0	PW cycles is irrelative of IORDY
1	PW cycles are extended by IORDY

When this bit is set, PW Cycles are extended by IORDY. For further details, see Figure 9.2

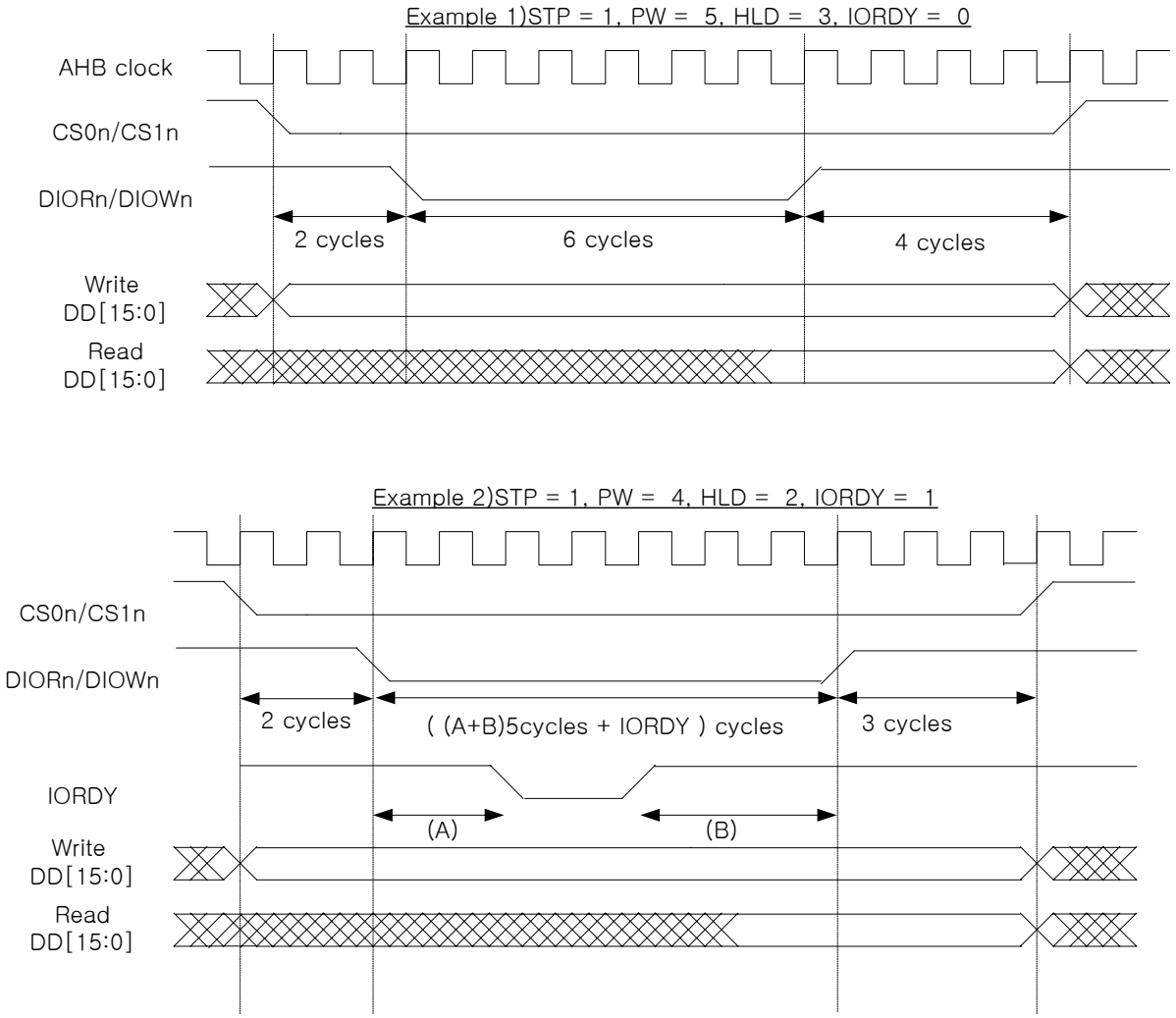


Figure 9.2 PIO Interface Timing Diagrams

UDMA Mode Control Register (UDMACTRL) 0xF0520044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		HRTL[1:0]		OTL[2:0]			ITL[1:0]		FDV[6:0]						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AHM		MAX3[1:0]		0	SNH[2:0]		0			HTB	TS	MS	UHT	UIO	UEN

HRTL[28:27]	HDMARDYn Trigger Level
00	4
01	8
10	12
11	14

In UDMA-IN mode, before UDMA-IN FIFO is full, Host must negate HDMARDYn signal. User must control these bits to prevent UDMA-IN FIFO from being full. These bits are only relative to UDMA-IN mode. If the number of valid data in UDMA-IN FIFO is more than this trigger level, HDMARDYn signal is automatically negated. This trigger level must be assigned the value which is equal to or greater than ITL (Input Trigger Level).

OTL[27:25]		Output Trigger Level
	000	Single Transfer (1read/ 1write)
	001	4 (vcnt <= 12)
	010	8 (vcnt <= 8)
	011	12 (vcnt <=4)
	110	16 (empty)

In UDMA-OUT mode, the output trigger level (OTL) is relative to Transfer Size (TS[0])bit. IDMA writes data to UDMA-OUT FIFO when it's possible to write as many data as the number of trigger level.

* For the better performance, use 32bit Transfer Mode (TS = = 1)

ITL[24:23]		Input Trigger Level
	00	Single Transfer
	01	4 (32bit data) (vcnt >= 4)
	10	8 (32bit data) (vcnt >=8)
	11	12 (32bit data) (vcnt >= 12)

Note. UDMA-IN mode, the Input Trigger Level is only relative to 32bit data.

IDMA reads from the UDMA-IN FIFO whenever the number of valid data is over than input trigger level.

* For the better performance, use 32bit Transfer Mode (TS = = 1)

FDV[22:16]	Frequency Divided Value
FDV	Frequency Divided Value

In case UDMA-IN mode, FDV value is needed to control timing parameters which is relative to MAX3 register.

In case UDMA-OUT mode, FDV value controls HSTROBE frequency. Don't set this value to zero in UDMA-OUT mode. Figure 9.3 shows the meaning of these bits.

AHM[15]	Adjust Hold margin
0	The hold margin of the data to be transferred has one AHB clock period.
1	The hold margin of the data to be transferred has two AHB clock period

When this bit is clear, the data to be transferred changes after one rising AHB clock is occurred. (Data hold margin is one clock)
In UDMA-OUT mode, when this bit is set to 1 and the values of FDV register are over than 2, the data to be transferred changes after two rising AHB clock is occurred. (Data hold margin is two clocks)

MAX3[13:12]	MAX3 Cycles (wait number of FDV cycles)
MAX3	(MAX3 + 1) FDV cycles are issued before host generate STOP signal in UDMA-IN MODE.

In UDMA-IN mode, before terminating operation, Host shall receive zero, one, two or three additional data word after negating DMARDYn. Therefore, delay is needed to operate correctly.

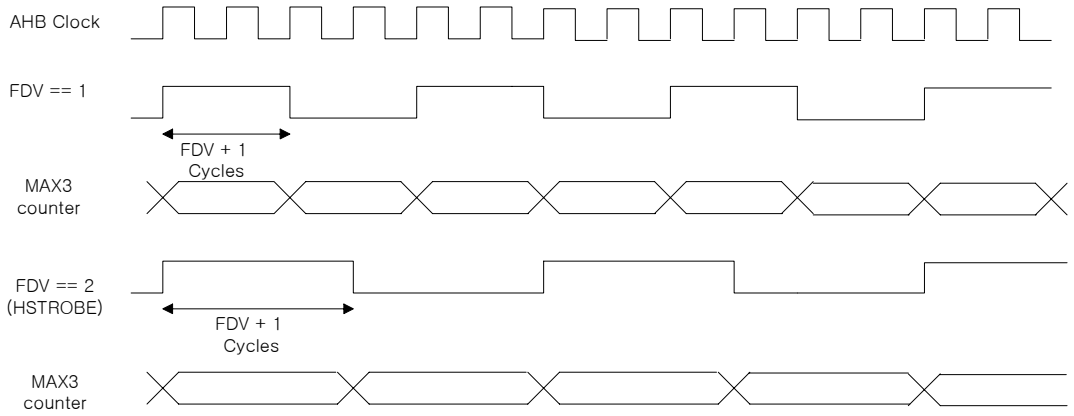


Figure 9.3 HSTROBE and Max3 Counter Timing Diagrams

SNH[10:8]	Setup and Hold time
SNH	Each TACK, TENV, TMLI is extended for (SNH+1) AHB cycles Tss is extended for (2*SNH+1) AHB cycles

There are the timing requirements in each mode. For example, in case UDMA mode4, TACK: min 20ns, TENV: min 20ns and max 55ns, TMLI : min 20ns, Tss : min 50ns is needed.

HTB[5]	Host Terminate By Transfer Counter
1	HOST terminates operation when CTCNT reaches zero.

When this bit is enabled and all data which is specified by UDMATCNT register are transferred completely, host automatically terminates all operation and then sets OPS bit to 1. If OPE is enabled, host generates interrupt to CPU.

When this bit is not enabled and CTCNT and UEN are enabled, Host wait until Device terminates operation.

TS[4]	Transfer Size
0	16 bit Transfer Mode (in Internal Bus)
1	32 bit Transfer Mode (in Internal Bus)

MS [3]	Multiple Section Enable Bit
1	To Support Multiple Section Toggle ‘1’ is recommended

This bit should be set to one for normal operation. Don’t clear this bit.

HT [2]	Host Termination
HT	Host Termination

Host can terminate UDMA operation normally by force using this bit. When this bit is set, IDE interface signals are negated or active by standard interface protocol.

UIO [1]	UDMA IO Configure
0	UDMA-IN MODE
1	UDMA-OUT MODE

When this bit is cleared, IDE Controller will receive the data from Device and then will send to CPU or General DMA. When this bit is set. IDE Controller will receive from CPU or general DMA and then will send the data to Devices.

UEN [0]	UDMA Enable Bit
1	UDMA Enable Bit

This bit is UDMA enable bit. When Operation is over, this bit is automatically cleared.

When this bit is clear by CPU, internal state goes to idle state and IDE interface signals are negated abnormally.

IDMA Control Register (IDMACTRL) 0xF0520048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
											-	LOC	CON	REP	IDE

LOC [3]	Locked Transfer
1	DMA transfer executed with lock transfer

Lock field controls the LOCK signal (refer to AHB specification). When the LOCK is set to 1, the DMA transfer doesn't be bothered by other AHB masters like LCD controller, ARM etc. This field is only meaningful in case of non-burst type transfers.

CON [2]	Continuous Transfer
0	DMA transfer begins from CSAR / CDAR address.
1	DMA transfer begins from CSAR / CDAR address. It must be used after the former transfer has been executed, so that CSAR and CDAR contain a meaningful value.

REP [1]	Internal DMA Repeat
0	When UDMA operation ends, IDE bit is automatically disabled.
1	When UDMA operation ends, IDE bit is not disabled.

IDE [0]	Internal DMA Enable
1	Internal DMA (IDMA) enable bit.

IDMA Source Address Register (IDMASA) 0xF052004C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SAR[15:0]															

This register contains the start address of source memory block for IDMA transfer. The transfer begins reading data from this address. This register is only relative of UDMA-OUT mode.

IDMA Source Parameter Register (IDMASP) 0xF0520050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMASK[23:8]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMASK[7:0]								SINC[7:0]							

SMASK [31:8]	Source Address Mask Register
0	non-masked
1	Masked so that source address bit doesn't be changed during DMA transfer

Each bit field controls the dedicated bit of source address field. That is, if SMASK[23] is set to 1, the 28th bit of source address is masked, and if SMASK[22] is set to 1, the 27th bit of source address is masked, and so on. If a bit is masked, a corresponding bit of address bus is not changed during DMA transfer. This function can be used to generate circular buffer address.

SINC [7:0]	Source Address Increment Register
sinc	Source address is added by amount of sinc at every write cycles. sinc is represented as 2's complement, so if SINC[7] is 1, the source address is decremented.

The addresses of DMA transfer are 32bit wide, but the upper 4bit of them are not affected during DMA transfer. If the source or destination address reaches its maximum address space like 0x7FFFFFFF or 0x2FFFFFFF, the next transfer is starting from 0x70000000 or 0x20000000 not from 0x80000000 or 0x30000000.

IDMA Current Source Address Register (IDMACSA) 0xF0520054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSAR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSAR[15:0]															

This register contains the current source address of DMA transfer. It represents that the current transfer read data from this address. This is read only register.

IDMA Destination Address Register (IDMADA) 0xF0520058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR[15:0]															

This register contains the start address of destination memory block for DMA transfer.

This register is only relative of UDMA-IN mode.

IDMA Destination Parameter Register (IDMADP) 0xF052005C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMASK[23:8]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMASK[7:0]								DINC[7:0]							

DMASK [23:8]	Destination Address Mask Register
0	non-masked
1	Masked so that destination address bit doesn't be changed during DMA transfer

Each bit field controls the corresponding bit of source address field. That is, if DMASK[23] is set to 1, the 28th bit of source address is masked, and if DMASK[22] is set to 1, the 27th bit of source address is masked, and so on. If a bit is masked, a corresponding bit of address bus is not changed during DMA transfer. This function can be used to generate circular buffer address.

DINC [7:0]	Destination Address Increment Register
dinc	Destination address is added by amount of dinc at every write cycles. dinc is represented as 2's complement, so if DINC[7] is 1, the destination address is decremented.

The addresses of DMA transfer are 32bit wide, but the upper 4bit of them are not affected during DMA transfer. If the source or destination address reaches its maximum address space like 0x7FFFFFFF or 0x2FFFFFFF, the next transfer is starting from 0x70000000 or 0x20000000 not from 0x80000000 or 0x30000000.

IDMA Current Destination Address Register (IDMACDA) 0xF0520060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CDAR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDAR[15:0]															

This register contains current destination address of DMA transfer. It represents that the current transfer write data to this address. This is read only register.

IDEINT Register 0xF0520064

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UB	NWA	DT	0		VEE[4:0]					0	VOE[4:0]				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				URS	FES	TRS	OPS	0				URE	FEE	TRE	OPE

UB[31]	Unit Busy
0	State of UDMA is IDLE.
1	UDMA is BUSY.

This bit is set while UDMA is running operation.

NWA[30]	Not Word Aligned
0	Word (32 bits) aligned.
1	Not Word (32 bits) aligned.

This bit is for 32bit transfer mode(TS = = 1).

This bit is set when UDMA –IN/OUT Operation is ended, remain FIFO data is not word aligned. When TS= = 1(32bit transfer mode) and this bit is set, CPU or DMA have to read last 16bit data.

DT[28]	Device Terminate
0	Device did not terminate operation.
1	Device terminated operation.

This bit is set when Device terminates operation before host terminates operation.

VEE[26:22]	Valid EVEN Entry
VEE	- Number of Valid data in Even FIFO when UDMA-IN mode - Number of Valid data in UDMA-OUT FIFO when UDMA-OUT mode

These bits specify for UDMA-IN mode. When operation is ended, these bits represent how many valid data is remain in even FIFO . Data needs to be read after internal DMA transfer is ended, because data may be remained in the receive FIFO when the receive FIFO trigger condition is not satisfied. Width of even FIFO is 16bit.

In case UDMA-OUT mode, these bits represent Number of data that are not transferred in UDMA-OUT FIFO.

VOE[26:22]	Valid ODD Entry
VOE	Number of Valid data in Odd FIFO When UDMA-IN mode

These bits specify for UDMA-IN mode. When operation is ended, these bits represent how many valid data is remain in odd FIFO. Data needs to be read after internal DMA transfer is ended, because data may be remained in the receive FIFO when the receive FIFO trigger condition is not satisfied .Width of odd FIFO is 16bit.

Cf. 32bit data of UDMA-IN FIFO is composed of one odd FIFO 16bit data and one even FIFO 16bit data.

For example, If VEE is equal to 0x6 and VOE is equal to 0x5 when operation ends, CPU or DMA has to read (5 x 32 bit + 1 x 16 bit) data.

URS[11]	UDMA Request Status
0	UDMA Request from the Device is not active or automatically cleared.
1	UDMA Request from the Device is active

This bit is set when DMARQ from the Device is active high. This bit is automatically cleared when UEN bit is set.

FES[10]	FIFO Error Status
0	Not FIFO Error
1	FIFO Error

This bit is set when FIFO Error occurs. In this case, Host have to terminate operation using HT bit, and check Frequency and timing parameter, and then try transfer again. This bit is cleared when this bit is written to 1.

TRS[9]	Trigger Level Request Status
0	Trigger Level Request is not active or automatically cleared.
1	Trigger Level Request is active

When this bit is set, CPU can read or write data. This bit is never set when IDE (Internal DMA Enable bit) is set. This bit is cleared when the number of valid data in In(Out)- FIFO is less(more) than Trigger level.

OPS[8]	Operation End Status
0	Host or Device does not terminate operation yet.
1	Host or Device terminates operation.

This bit is set when the operation is ended and cleared when this bit is written to 1.

URE[3]	UDMA Request Interrupt Enable
0	UDMA Request Interrupt is disabled.
1	UDMA Request Interrupt is enabled.

FEE[2]	FIFO Error Interrupt Enable
0	FIFO Error Interrupt is disabled.
1	FIFO Error Interrupt is enabled.

TRE[1]	Trigger Level Interrupt Enable
0	Trigger Level Interrupt is disabled.
1	Trigger Level Request Interrupt is enabled.

OPE[0]	Operation End Interrupt Enable
0	Operation End Interrupt is disabled in UDMA Mode.
1	Operation End Interrupt is enabled is enabled in UDMA Mode.

UDMATCNT Register 0xF0520068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								UDMATCNT[23:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UDMATCNT[15:0]															

These bits define how many data need to be transfer By host. The host will transfer 2*UDMATCNT bytes in UDMA-OUT mode and 4*UDMATCNT bytes in UDMA-IN mode.

UDMAIN Register (Read Only) 0xF052006C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDMAIN[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UDMAIN[15:0]															

This register is Read Access port to read data (received from the Device) in IDE FIFO. When TS is cleared (Transfer size: 16bits), UDMAIN[31:16] will be zero and UDMAIN[15:0] is valid data.

UDMAOUT Register (Write Only) 0xF0520070

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDMAOUT[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UDMAOUT[15:0]															

This register is Write Access port to write data(to be send to Device) in IDE FIFO. When TS is cleared (Transfer size: 16bits), only UDMAOUT[15:0] is valid data.

UDMACRC Register (Read Only) 0xF0520074

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UDMACRC[15:0]															

When operation is ended, it’s possible to check difference between Device CRC value and Host CRC value. If each CRC value is different, the operation was ended incorrectly.

UDMACTCNT Register **0xF0520078**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CTCNT[23:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CTCNT[15:0]															

CTCNT[23:0]	Current TCNT
CTCNT	Current Transfer Counter.

At the beginning of Transfer, The CTCNT is updated by TCNT value.

When UDMA is Enabled, CTCNT will be decremented according to TS (transfer Size). CPU can check how many data was transferred by checking these bits. When TCE bit is set and all number of data specified by UDMATCNT was transferred (CTCNT reaches zero), Host will terminate operation.

10 SATA Host Interface

10.1 Overview

The SATA Host implements the Serial Advanced Technology Attachment (ATA) storage interface for physical storage devices.

SATA is a half-duplex system. Either a receive or transmit operation is performed between the two agents (host and device) at any given time, but not both. This is true only for data frames transfer. Control traffic (primitives) is full duplex to maintain receiver synchronization. For example, SYNC primitives are sent continuously while both host and device sides are in their idle states.

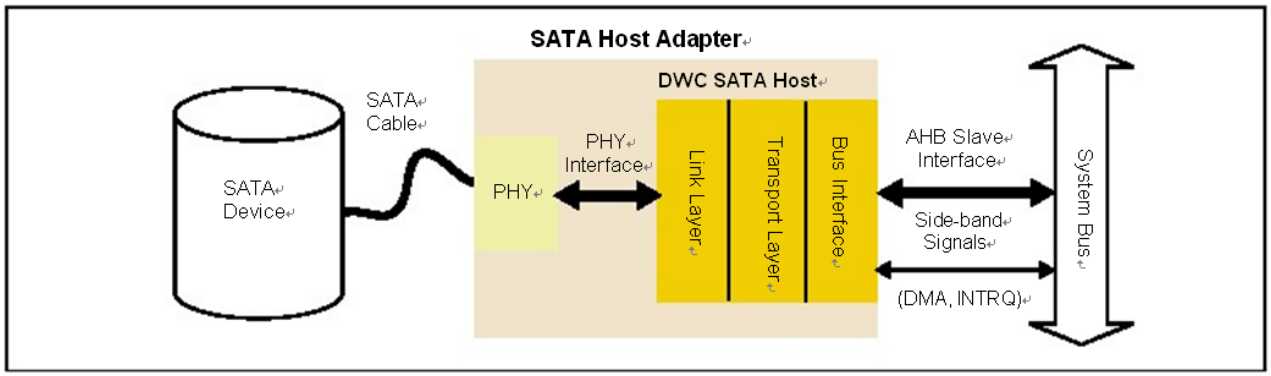


Figure 10.1 SATA Host Adaptor Block Diagram

The SATA Host link consists of three main functional blocks: Bus interface, Transport Layer, and Link Layer. Together with the physical layer (PHY) it forms a complete Serial ATA (SATA) host adapter interface.

It operates primarily in two or three clock domains: receive (Rx), transmit (Tx), and application. Rx (when present) and Tx clocks are generated in the PHY and are 75 for Gen1 (150 for Gen2).

Most of the Link Layer (both receive and transmit data paths) and part of the Transport Layer always operate in the Tx clock domain. Rx clock is a clock recovered from the PHY and is used for clocking data for performing optional 8b/10b decoding, dropping ALIGNs and aligning data.

The Bus Interface block provides a configurable AHB slave interface, which is used to connect to the system bus. Host application software (driver) accesses SATA Host link using a set of ATA, SATA and sata host link-specific registers. DMA flow control is implemented with several handshaking signals compatible with the internal DMA controller.

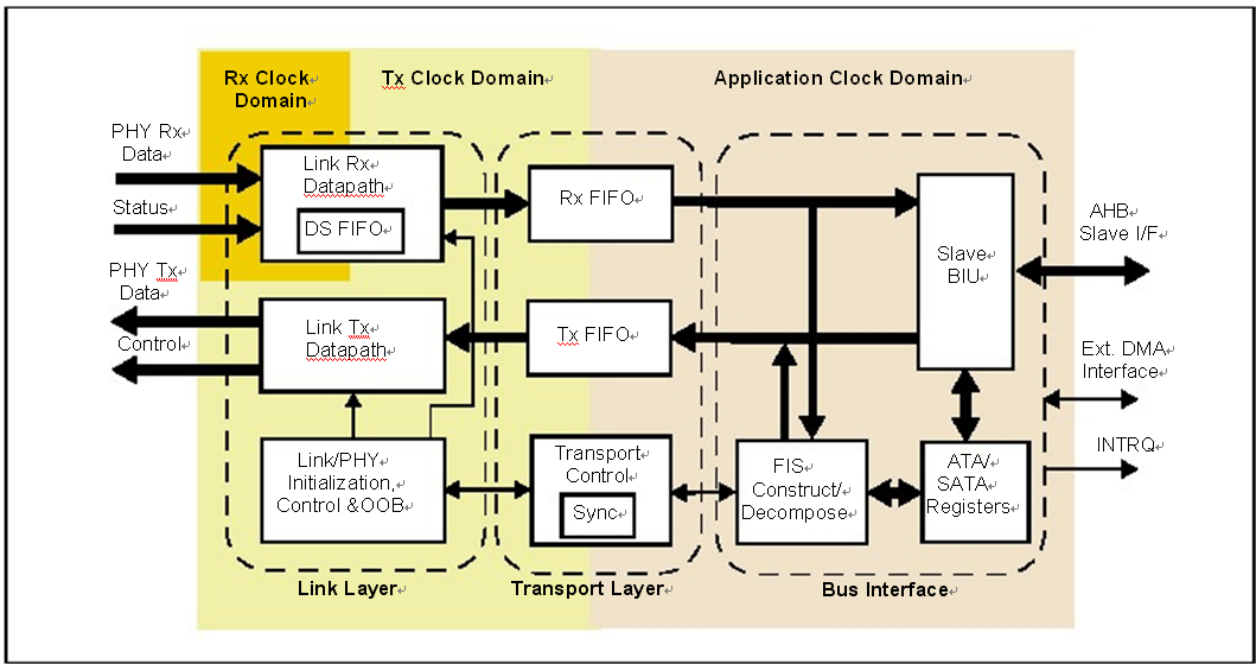


Figure 10.2 SATA Host Link Block Diagram

10.2 SATA Host Bus Interface

The Host Bus section discusses the following topics

- Bus Interface Overview
- Bus Interface FIS Decomposition
- Bus Interface FIS Construction
- Programmed I/O Operation
- Direct Memory Access Operation
- Interrupt Control
- Register Access
- AHB Error Conditions
- Bus Interface Power Management
- Hot-Plug
- PHY and Link Control
- Reset Conditions
- BIST operation

10.2.1 Bus Interface Overview

The Bus Interface contains all SATA Host registers and consists of the following modules:

- Receive Data Path (RxDP)
- Transmit Data Path (TxDP)
- Shadow/ Serial ATA Registers (SSR)
- DMA Flow Control (DMAC)
- Slave Bus Interface (SBIU)

Next figure shows how the Bus Interface Block Bus connects the SATA Host Link Layer and the The SATA Host Transport Layer to a system bus by way of the slave AHB interface.

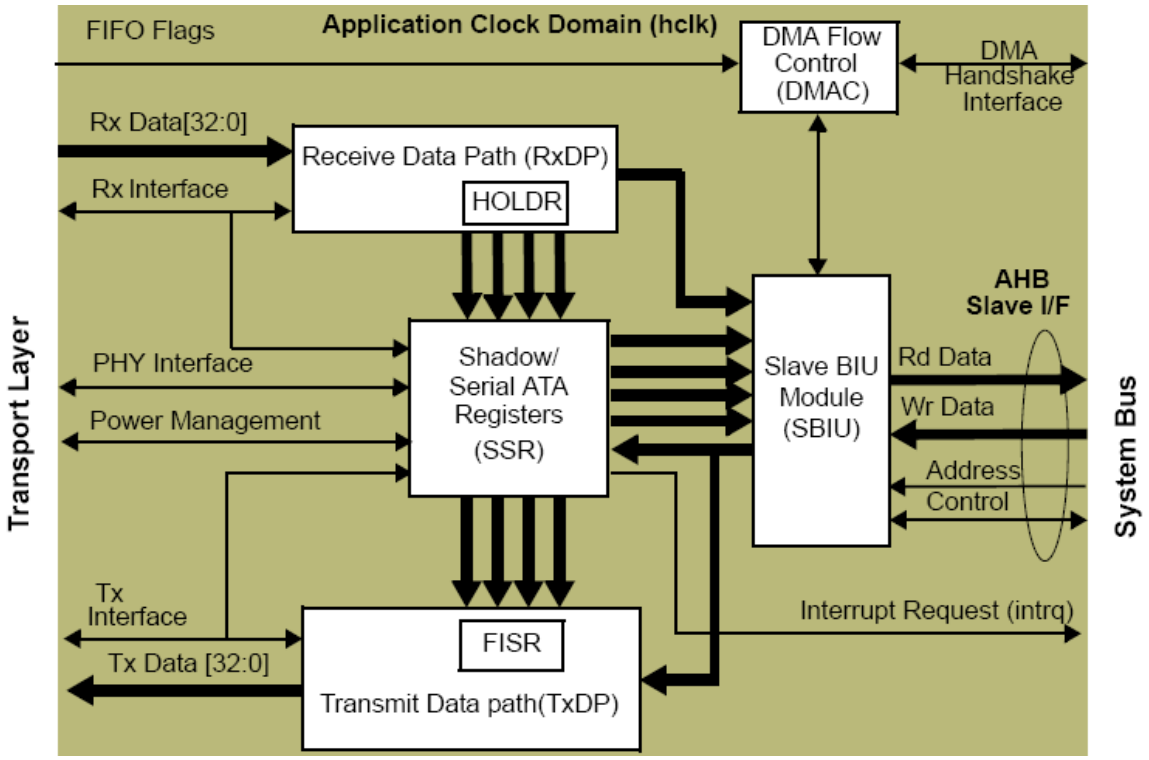


Figure 10.3 SATA Host Bus Interface Block Diagram

The Bus Interface operates in the system bus (application) clock domain and implements the following functions:

- "Bus Interface FIS Decomposition" and "Bus Interface FIS Construction": these are basic Transport Layer functions.

- "Programmed I/O Operation": monitors interface state
- "Direct Memory Access Operation" : generates DMA controller handshake signals and supports First Party DMA capability.
- "Interrupt Control": implements ATA master (Device 0) emulation and other SATA Host specific interrupts.
- "Register Access": monitors register reads/writes and generates requests to the Transport Layer to send corresponding FIS to the device.
- "AHB Error Conditions" : receives errors from the PHY, Link Layer, and Transport Layer and updates corresponding SError register bits.
- "Bus Interface Power Management": Power management via SATA registers.
- "Hot-Plug": provides details of hot plug support.

10.2.2 Bus Interface FIS Decomposition

FIS decomposition is a Transport Layer function and is controlled by the Transport State Machine (TSM) module (see "The SATA Host Transport Layer" for more details). The TSM detects FIS in the RxFIFO by its type (least-significant byte of the first double-word (DWORD)). If the FIS type is one of the following non-data FIS types, the FIS is read from the RxFIFO into the RxDP Holding registers (HOLDR):

- Register FIS (Device to Host)
- Set Device Bits FIS
- PIO Setup FIS
- DMA Setup FIS
- BIST Activate FIS

The FIS end-status and errors are then checked by the TSM. If no errors were detected, the HOLDR content is loaded into the corresponding registers according to the FIS type. Bit 32 of the RxFIFO is used to indicate the FIS end-status/error DWORD. If any error is detected, the SError register bits are updated accordingly, the HOLDR content is discarded, and the registers are not updated. If the FIS is determined to be a data type, the data is read from the RxFIFO either by the application software read access to the CDR0 location (PIO mode) or by the DMA controller read access to the DMADR location (DMA mode).

Register FIS (Device to Host) is used by the device to indicate command completion status and update Shadow ATA registers. When the TSM indicates the Register FIS type in the RxFIFO, the FIS is read into the HOLDR and the FIS status is checked. If no errors are detected, the HOLDR content is transferred to the corresponding ATA Shadow registers. If the interrupt bit is set, the IPF (Interrupt Pending Flag) register is set and interrupt (intrq) is asserted if the nIEN bit of the Device Control register is cleared.

The Set Device Bits FIS is used to update ATA Error, Status shadow registers (except bits BSY (7) and DRQ (3)) and SATA SActive register. When the Transport Layer indicates the Set Device Bits FIS type in the RxFIFO, the FIS is read into the HOLDR and the status is checked. If no errors are detected, the HOLDR content is transferred to the corresponding ATA and SATA registers.

PIO Setup FIS is used for PIO operation: DMA Setup FIS for First Party DMA operation, operation, BIST Activate FIS to put sata host in one of the SATA BIST modes.

10.2.3 Bus Interface FIS Construction

Any of the following conditions initiate FIS construction:

- ATA Command or Device Control register write (Register FIS)
- ATA Data register write and previous FIS was PIO Setup FIS (PIO Data FIS)
- DMA Activate FIS or DMA Setup FIS with A = 1 bit is received (DMA Data FIS)
- BIST Control register is written (BIST Activate FIS)

The corresponding FIS is constructed in the TxDP module and written (pushed) into the TxFIFO for the transmission by the Link Layer. Bit 32 of the TxFIFO is used to indicate the last DWORD of the FIS to the Link Layer, so it closes the frame with CRC and EOFp. The transmission end status is received by the Bus Interface from the RxFIFO the same way as during FIS reception process. The Bus Interface maintains Data FIS byte count to ensure that the FIS length does not exceed the required 8192 bytes (2048 DWORDs) for both PIO and DMA modes.

Register FIS (Host to Device) is used by the host to send a command to the device or initiate a control function (for example, Device Reset). Register FIS is constructed in the TxDP FISR registers by transferring ATA Shadow register content to FISR upon application software write to either ATA Command or Control register.

10.2.4 Programmed I/O Operation

The following subsections describe the programmed I/O (PIO) operation:

- Read Transfer (Device to Host)
- Write Transfer (Host to Device)
- Slow Device Access

10.2.4.1 Read Transfer (Device to Host)

PIO read transfer starts with the host issuing a “PIO Read” command to the device by sending a corresponding Register FIS. After the command has been processed, the device sends the PIO Setup FIS to the host with D bit = 1. The content of the FIS is loaded into the HOLDR registers. The device sends the PIO Data FIS. When the header of the PIO Data FIS has been received, intrq is asserted.

The TSM detects Data FIS in the RxFIFO and signals the Bus Interface to transfer “Initial” Error and Status register content from the HOLDR to the ATA registers when the Data FIS is being received.

RxFIFO is logically mapped to the 16-bit ATA Data register (CDR0). The data DWORD is popped out of the RxFIFO every two read accesses to this register.

The Bus Interface keeps track of the read accesses and signals a “PIO count reached” condition, indicating the end of the current PIO transfer to the Transport Layer. The transfer “Ending” Status register content is then transferred from the HOLDR registers to the ATA Status register.

If the device has more PIO data to send, the above process is repeated from the device sending PIO Setup FIS followed by the Data FIS until all data has been transferred.

10.2.4.2 Write Transfer (Host to Device)

A PIO write transfer is similar to the PIO read transfer, except D = 0 in the PIO Setup FIS and the “Initial” Error and Status register content is transferred to the Shadow registers immediately upon successful reception of the PIO Setup FIS.

TxFIFO is logically mapped to the 16-bit ATA Shadow Data register (CDR0). The data DWORD is pushed into the TxFIFO every two application software writes to this register. The Bus Interface keeps track of the write accesses and signals a “PIO count reached” condition indicating the end of the current PIO transfer to the Transport Layer.

The transfer “Ending” Status register content is transferred from the HOLDR registers to the ATA Status register.

If the transfer is not finished, the above process is repeated from the device sending the PIO Setup FIS, followed by the host sending the Data FIS until all data has been transferred. The device then sends Register FIS with the ending command status.

10.2.4.3 Slow Device Access

In some cases a device might transfer PIO data at a very slow rate (for example, ATA to SATA bridge sending one DWORD at a time at a 4 MB/sec rate). The SATA Host AHB slave interface inserts wait states (hready = 0) into the PIO access until either the RxFIFO contains data, or the TxFIFO contains free space.

If the device does not supply data during a PIO read operation or accept data during a PIO write operation within 256 hclk cycles, the SATA Host generates an ERROR response on the hresp output (assuming RETURN_ERR_RESP = 1) to prevent a possible bus lock-up condition from occurring.

This delay corresponds to about a 2.3 MB/sec transfer rate with a 300 MHz hclk, assuming back-to-back software reads (300 MWORD/sec divided by 256 = 1.17 MWORD/sec, or 2.34 MB/sec).

10.2.5 Direct Memory Access Operation

SATA Host requires an external direct memory access (DMA) controller to support DMA data transfers and provides several handshaking signals for flow control. This controller resides on the system bus and acts as a bus master during DMA transfers.

This section discusses the following topics:

- DMA Initialization
- DMA Read Transfer (Device to Host)

- DMA Write Transfer (Host to Device)
- DMA Requirements
- DMA Termination
- First-Party DMA

10.2.5.1 DMA Initialization

The DMA initialization process is described as follows:

1. The host software programs (initializes and enables) the DMA controller with all the parameters necessary for a given transfer (for example, transfer direction, address pointers, burst size.).

2. The host software initializes corresponding MRD/MWR fields of the DBTSR register, then sets RXCHEN/TXCHEN bits of the DMACR register depending on the transfer direction. The DMACR.TXMODE bit selects whether Tx Data FIS is closed after dma_finish_tx is asserted (TXMODE = 0) or after DMACR.TXCHEN is cleared (TXMODE = 1).

3. The host software issues a DMA command to the device by writing to the Command register resulting in the Register FIS transmission to the device.

4. The device executes the command and notifies the host that it is ready to send/receive data. During a read operation the device sends the Data FIS to the host; during a write, a DMA Activate or a DMA Setup FIS with A = 1 requesting Data FIS from the host is sent.

5. The data is then transferred between the two DMA agents: system memory and peripheral (both system bus slaves). DMA controller acts as a flow controller.

10.2.5.2 DMA Read Transfer (Device to Host)

During a read operation the data is transferred from the RxFIFO to the system memory through the DMA channel. RxFIFO acts as a source of data (source peripheral) and system memory, as a destination of data (destination memory). Data is received in the RxFIFO from the device in the form of one or more Data FISs.

The Transport Layer decodes the FIS type and initiates a DMA transfer. The Bus Interface generates DMA requests to the DMA controller (dma_req_rx or dma_single_rx) based on the RxFIFO “empty” and “almost empty” flags.

The following sequence of events occurs during a DMA read transfer:

1. The Transport Layer detects a Data FIS in the RxFIFO and notifies the Bus Interface to start DMA transfer;

2. The Bus Interface asserts the dma_req_rx signal to initiate a burst read transaction if RxFIFO has enough data for at least one burst or dma_single_rx to initiate a single transaction if RxFIFO has at least one data DWORD. MRD field of the DBTSR register should be set to the burst transaction size expected on the bus in DWORDs prior to setting RXCHEN bit of the DMACR register.

3. The DMA controller initiates single or burst read cycles on the system bus to the DMADR location to transfer data from the RxFIFO to its internal FIFO and then to destination memory.

4. The DMA controller asserts dma_ack_rx to indicate the DMA transaction completion. The Bus Interface detects dma_ack_rx assertion and negates dma_req_rx and dma_single_rx.

5. The process is repeated until all the data in the Data FIS has been transferred. The Bus Interface suspends the transfer by negating dma_req_rx or dma_single_rx. The device either sends another Data FIS or Register FIS with command end-status information if all data was transferred. Software clears the RXCHEN bit of the DMACR register.

10.2.5.3 DMA Write Transfer (Host to Device)

During a write operation the data is transferred from the system memory to the TxFIFO through the DMA channel. TxFIFO acts as a data destination (destination peripheral) and system memory acts as a source of data (source memory). The Bus Interface generates DMA requests to the DMA controller (dma_req_tx or dma_single_tx) based on the TxFIFO “full” and “almost full” flags.

The following sequence of events occurs during a DMA write transfer:

1. The device signals its readiness to receive data by sending a DMA activate or DMA Setup FIS with A = 1 to the host. The Transport Layer detects this FIS and notifies the Bus Interface to start a DMA transfer.

2. The Bus Interface asserts dma_req_tx to initiate a burst write transaction if TxFIFO has enough space for at least one

burst or dma_single_tx if TxFIFO has space for at least one DWORD. MWR field of the DBTSR register should be set to the burst transaction size expected on the bus in DWORDs prior to setting TXCHEN bit of the DMACR register.

3. The DMA controller generates single or burst write cycles on the system bus to the DMADR location to transfer data from the system memory to the TxFIFO.

4. The DMA controller asserts dma_ack_tx to indicate the DMA transaction completion. The Bus Interface detects dma_ack_tx assertion and negates dma_req_tx and dma_single_tx.

5. The process is repeated until the number of bytes transferred reaches 8192 bytes (2048 DWORDs) or the DMA controller asserts dma_finish_tx (same timing as dma_ack_tx) indicating the completion of the current block (if DMACR.TXMODE = 0). The Bus Interface notifies the Link Layer to close the FIS (generate CRC and EOFp) by setting bit 32 of the last DWORD. The DMA controller should be suspended at this time. A device sends either a DMA Activate FIS to request another Data FIS or a Register FIS with command endstatus information. When all data has been transferred, the software clears the TXCHEN bit of the DMACR register.

10.2.5.4 DMA Requirements

DMA block and burst transaction sizes are critical for DMA operation. (Also see the definition of "DMA transfer".) These sizes should be selected properly to ensure error-free bus transfers. It is required that the DMA write burst transfer does not cross the 8192-byte Data FIS boundary, because the Transport Layer maintains the DMA state for the duration of the Data FIS transmission.

Note: Violation of these requirements results in transfer abort and bus ERROR response (if RETURN_ERR_RESP = 1).

DMA controller block/burst sizes should be set as follows:

- Receive (read) operation: The block can be any size up to the DMA transfer size. Burst can be any size up to the maximum limited by the DBTSR.MRD value.
- Transmit (write) operation: Block and burst sizes must be selected so that the burst does not cross a 8192-byte FIS boundary. If the block size is set to be exactly 8192 bytes (except for the last one), then the burst size can be any value limited by the DBTSR.MWR. If the block size is larger than 8192 bytes (for example, 16 KB, 32 KB), it must be an integer multiple of 8192 and the burst size must be a 2^n number of DWORDs (n = 0 to 10, limited by the DBTSR.MWR). If the block size is smaller than 8192 bytes, then the several consecutive block sizes must add up to 8192 bytes and the burst size can be any value limited by the DBTSR.MWR. For example: 1 KB - 4 KB - 2 KB - 1 KB, 1 KB - 2 KB - 2 KB - 3 KB.

Note: Using DMACR.TXMODE = 0 in this case results in transmission of the Data FIS smaller than 8192 bytes, which is violation of the SATA spec. It is recommended to use DMACR.TXMODE = 1 for any block/burst size.

- Transmit (write)/Receive (read) operation: The block size must be 32-bit aligned.
- The DBTSR register should be programmed with the burst transaction size values (in DWORDs): MRD field for read transfer, MWR field for write transfer prior to enabling the channel, and the same values should be used to program DMA controller.
- SATA Host should be notified that the DMA channel is enabled by setting the corresponding TX/RXCHEN bit of the DMACR register prior to issuing a DMA command to the device. The TX/RXCHEN bits must be cleared by software after the DMA controller has transferred all data for the command. These bits are cleared automatically on completion of the first-party DMA data phase (see "First-Party DMA" on page 43).
- DMA transaction transfer size should be 32-bits and the address should be 32-bit-aligned, and be within the DMADR range.

10.2.5.5 DMA Termination

A DMA transfer can be terminated (aborted) before its normal completion either by a host or a device in the following cases:

- DMA Read Transfer Abort
- DMA Write Transfer Abort

[DMA Read Transfer Abort]

A DMA read transfer is aborted:

- By the device when it stops sending Data FIS and sends Register FIS with command end status. Host software detects this condition with ATA Status/Error registers and disables the DMA controller.

Note : The SATA specification recommends ignoring reception of the DMATp and completing the current transfer for better compatibility.

- By the host software when it disables the DMA channel by clearing the RXCHEN bit of the DMACR register. This is usually done prior to resetting the device with SRST/Device Reset command (see next case).
- By the host software when it sets SRST bit of the Control register or issues Device Reset command to the device. The Transport Layer notifies the Link Layer of the condition. The current Data FIS is “flushed” from the RxFIFO and the corresponding Register FIS is sent to the device. Host software disables the channel and clears RXCHEN/TXCHEN bits of the DMACR register.

[DMA Write Transfer Abort]

A DMA write transfer is aborted:

- By the device when it sends DMATp to the host requesting to abort current transfer. The Transport Layer detects this condition and sets DMAT bit of the INTPR register. Interrupt output intrq is asserted if the DMATM bit of the INTMR register is set.
- By the host software when it disables (deactivates) the channel by clearing the TXCHEN bit of the DMACR register. The current Data FIS is closed. Host software issues a SRST/Device Reset command to the device (see next case).
- By the host software when it sets SRST bit of the Control register or issues Device Reset command to the device. The Transport Layer notifies the Link Layer of the condition. The current transfer is interrupted, TxFIFO is “flushed” and the corresponding Register FIS is sent to the device. Host software is responsible for disabling the DMA channel in this case.

10.2.5.6 First-Party DMA

First-Party DMA is a method of implementing the native SATA command queuing mechanism. The details can be found in the SATA specification. SATA host provides several registers to implement this functionality, which serve as hooks for software. The functionality is the same as described in the “Direct Memory Access Operation”, except it is preceded by the following:

1. The host software sets SActive register bit corresponding to the command TAG value and issues “First-Party DMA” command. Register FIS is sent to the device and BSY is set in the Status register.

2. Device adds the command to it’s internal queue and sends Register FIS to clear BSY bit.

3. Device executes a command from the queue and sends DMA Setup FIS to the host to initiate data transfer phase.

Note: Host software is not allowed to issue a new command to the device during the current First-Party DMA data phase, however, since the BSY bit is cleared, potentially it can. In this case, SATA Host sets the BSY bit and delays the corresponding Register FIS transmission until after the current First-Party DMA data phase is complete, that is, all the data for the current DMA Setup FIS has been transferred.

4. Upon reception of the DMA Setup FIS, the Bus Interface updates corresponding the FPTAGR, FPBOR and FPTCR registers from the values in the FIS and sets NEWFP bit of the INTPR register. Interrupt output intrq is asserted if NEWFPM bit of the INTMR register is set.

5. The host software updates the DMA controller context accordingly and enables the channel for transfer. The host software sets the corresponding TXCHEN or RXCHEN bit in the DMACR register.

6. The SATA Host generates dma request signals to the DMA controller to initiate the transfer. The operation proceeds as described in the “DMA Operation” chapter.

7. When the device completes the command data phase, it sends Set Device Bits FIS to the host with the bit corresponding to the TAG value set in the SActive field, so this bit is cleared in the host SActive register.

The RXCHEN/TXCHEN bits of the DMACR register are cleared automatically by the SATA Host when the current data phase is complete, that is, when the transfer count for the DMA Setup FIS is reached.

10.2.6 Interrupt Control

The SATA Host provides an interrupt request output intrq to implement system interrupt. It is asserted if any of the interrupt events are set in the INTPR register and the corresponding mask bit is set in the INTMR or the IPF flag is set and the nIEN bit is cleared in the Device Control register (CLR0).

The ATA interrupt pending flag (IPF) is set when the device signals an interrupt condition to the host by sending a Register, Set Device Bits, or PIO Setup FIS with Interrupt bit set (I = 1). The IPF is cleared when the ATA Status register (CDR7) is read. If IPF is set and nIEN in the Device Control register is cleared (nIEN = 0), SATA Host asserts the external interrupt output intrq.

The intrq output is also asserted if any of the bits in the INTPR is set and the corresponding bit in the INTMR is set (interrupt event is detected and enabled/unmasked). The following events cause INTPR bits to be set:

- DMAT: The DMATp is received from the device during DMA Data FIS transmission.
- NEWFP: A new DMA Setup FIS is received from the device.
- PMABORT: A power mode abort condition is detected by the Link Layer.
- ERR: Any of the SError register bits is set.
- NEWBIST: The new BIST Activate FIS is received from the device
- PRIMERR: The Link Layer detects an error in the primitive DWORD.
- CMD_ABORT: A Register or Set Device Bits FIS is received with Status.ERR==1 (“Command abort” status).
- CMD_GOOD: A Register or Set Device Bits FIS is received with Status.ERR==0 (“Command good” status).

ERRMR register is used to mask corresponding bits of the SError register before they cause setting the INTPR register ERR bit.

10.2.7 Register Access

The SATA Host occupies 2048 bytes of the system memory space and uses the lower 256 bytes for Shadow, SATA, and SATA Host-specific registers. The rest of the locations (addresses from 0x100 to 0x3FF) are not implemented and return an ERROR response if accessed (if RETURN_ERR_RESP = 1). All SATA Host register locations are 32-bit aligned.

Address CDR0 is used for data transfer to/from the Tx/RxFIFO in PIO mode and DMADR in DMA mode. In PIO mode, SATA Host supports only 16-bit single bus transfers. In DMA mode, SATA Host supports 16- or 32-bit single/burst bus transfers.

The ATA/ATAPI Shadow and SATA registers’ access is implemented according to the SATA specification. SATA Host-specific registers provide various functions such as: DMA support, interrupt control, testing, PHY/Link control, and so on.

10.2.8 AHB Error Conditions

The SATA Host AHB slave interface detects a number of illegal conditions during bus transfer, sets SError register bit 11 (ERR_E), and generates ERROR response on the hresp output if RETURN_ERR_RESP = 1.

Following is the list of these conditions:

- Address is not 32-bit aligned when accessing CDR0-CDR7, CLR0, DMADR locations, or 16-bit-aligned when accessing DMADR location and AHB_DATA_WIDTH = 16.
- Write access to the read-only locations (for example, SStatus, FPTR, FPBOR, FPTCR.)
- Address selects non-implemented area of the SATA Host address space.
- Shadow ATA registers write access when either BSY or DRQ bit is set in Status register.
- Transfer size exceeds 32 bits.
- Access to the FIFO (CDR0/DMADR) and SATA host not in PIO or DMA mode.
- Burst access to the FIFO (CDR0) in PIO mode or to registers if the burst size exceeds one (burst size of one is acceptable).
- Access to the FIFO (CDR0/DMADR) in PIO or DMA mode when the RxFIFO is empty, or the TxFIFO is full. his error might result from the AHB burst size exceeding the value programmed in the DBTSR register that sets the Rx/TxFIFO threshold values for DMA flow control. DMA bus transfer starts only when there is enough data in the RxFIFO or enough space in the TxFIFO for the requested DMA transaction. In PIO mode, the error is

generated if the device supplies read data (or accepts write data) at a rate slower than 2.3 MB/sec (hclk = 300 MHz). Refer to “Slow Device Access” for more details.

Note: In some cases, SError bit 11 (SCR1.ERR_E) can be set while the AHB ERROR response is not generated (assuming RETURN_ERR_RESP = 1). For example, when a TxFIFO overflow condition is detected at the end of the burst transfer as a result of the last DWORD being written to the TxFIFO, when the FIFO is already full.

All of the cases above are indications of a system design or operation error and are not expected to happen during normal operation.

10.2.9 Bus Interface Power Management

The host software can request either PARTIAL or SLUMBER power management states by writing to the SPM field of the SControl register (bits [15:12]). The device requests power management state by transmitting PMREQ_Pp or PMREQ_Sp primitives to the host. The power state machine is implemented in the Link Layer power management module (refer to “Link Layer Power Management Details” for more details). It asserts corresponding signal (phy_partial or phy_slumber) to the PHY to enter the power management state.

The host can disable transition to power management states when the device requests it using the IPM field of the SControl register (bits [11:8]).

The power management state is terminated when either one of the following conditions becomes true:

- Host software issues a new command by writing to the Command register
- Host software requests device reset by toggling SRST bit of the Device Control register
- Host software requests BIST mode by writing to the TXBISTPD register
- Host software requests transition to active mode by writing to the SControl SPM field (bits [15:12] = 4’b0100)
- Device requests interface wakeup by transmitting COMWAKE OOB sequence

The state of the interface (active, PARTIAL or SLUMBER power management) is reflected in the IPM field of the SStatus register (bits [11:8]).

10.2.10 Hot-Plug

The SATA Host supports hot-plug through the use of the SError bit 26 (SCR1.DIAG_X). It is set every time the Link detects a COMINIT sequence, indicating a new device insertion event or system power-up.

10.2.11 PHY and Link Control

The SATA PHY control is provided via SMU_I2C only.

The Link Layer features (scrambler, descrambler, repeat drop) are controlled by the LLCR register and can be disabled for testing by clearing corresponding bits.

10.2.12 Reset Conditions

The different reset conditions are identified as follows:

- Power-on reset. This reset is initiated by the system bus immediately after power-on or when it crashes. It is provided by the system bus as an asynchronous, active-low signal. All components of the SATA Host are initialized, including Link Layer, Transport Layer, FIFOs, ATA/SATA registers.
- Hard interface reset (COMRESET). This reset is initiated by the host software by setting bit 0 of the SControl register (DET field). The actions are equivalent to power-on reset. The Link Layer LFSR registers are initialized to the required values.
- Soft interface reset. This reset is initiated by the host software either by setting the SRST bit in the Shadow Device Control register or by issuing an ATAPI DEVICE RESET command (08h). In both cases, the Register FIS is sent to the device. It is used by the host to reset the device. Any on-going interface activity is interrupted to allow for the Register FIS to be sent to the device.

10.2.13 BIST Operation

Note : Scrambler/Descrambler is bypassed (disabled) in SATA Host Link Layer in all BIST modes except when in loopback device far-end retimed mode.

10.2.13.1 Loopback Device

The sata host enters one of the BIST loopback device modes when a corresponding BIST Activate FIS is successfully received from the device.Upon reception of the valid BIST Activate FIS, sata host sets NEWBIST bit of the INTPR register and asserts interrupt output intrq if the NEWBISTM = 1 in the INTMR register. SStatus register DET field (bits [3:0]) returns 4'b0100 value when read.

The following loopback device modes are supported:

- Far-end retimed

Sata host receives BIST Activate FIS with Pattern Definition field (bits [23:16] of the first DWORD) = 0x10 from the RxFIFO and stores it in the RXBISTPD register.

All the data received from the device in the form of a SATA-compliant pattern is retimed in the Link Layer and transmitted back to the device.

- Far-end analog (not support this mode)
- Far-end transmit only

Sata host receives BIST Activate FIS with Pattern Definition field (bits [23:16] of the first DWORD) = 0x80 or 0xA0 (scrambling is bypassed) from the RxFIFO and stores it in the RXBISTPD register. The following two data DWORDs are stored in the RXBISTD1 and RXBISTD2 registers.

Sata host transmits a corresponding SATA non-compliant test pattern to the device based on the value in the RXBISTD1 register (bits [31:0]):

- 0xEE27FEF1: Low transition density pattern
- 0xB5B5B5B5: High transition density pattern
- 0xABABABAB: Low frequency spectral component pattern
- 0x7F7F7F7F: Simultaneous switching outputs pattern

Loopback device BIST modes can be exited either when the device signals COMINIT OOB condition, or when the host software sets SControl[0] = 1 (COMRESET).

Note : If the device sends a BIST Activate FIS with a request to enter a non-supported loopback mode, sata host responds with R_ERRp response upon reception of the FIS.

10.2.13.2 Loopback Initiator

The host software should reset the interface by toggling SControl[0] (COMRESET) and wait for the BSY bit to be cleared prior to entering one of the initiator loopback modes (or changing from one BIST mode to another) to ensure the device is in the known state and the DW_hsata BIST logic is properly initialized.

Sata host enters one of the BIST loopback initiator modes after the host software writes to the TXBISTPD and BISTCR registers. Any write to the TXBISTPD register initiates transmission of the BIST Activate FIS to the device. After the host successfully transmits this FIS, it enters this mode and generates/receives a compliant test pattern. SStatus register DET field (bits [3:0]) returns 4'b0100 value when read. BISTSR and BISTFCTR are updated with error/FIS count information for each received BIST FIS.

The following BIST initiator modes can be requested by the host software:

- Far-end retimed

The host software writes the BISTCR register PATTERN field to select one of the SATA-defined compliant patterns:

- 3'b000: Simultaneous switching bit pattern
- 3'b001: High transition density bit pattern
- 3'b010: Low transition density bit pattern
- 3'b011: Low frequency spectral component bit pattern
- 3'b100: Composite pattern

The host software writes 10h to bits [7:0] of the TXBISTPD register. Corresponding BIST Activate FIS is sent to the device with the Pattern Definition field (bits [23:16] of the first DWORD) containing this value.

After successful transmission of the BIST Activate FIS (the device acknowledges the FIS with R_OKp). The SATA Host continuously generates the requested compliant pattern in the form of BIST frames, and checks for errors on the receive side.

BISTFCTR register is updated with the received BIST frame count and BISTSR is updated with a frame/burst error count. BISTDECR is updated with DWORD error count. SError register is updated with CRC, disparity and 10B8B errors for each frame. BISTFCTR, BISTSR, and BISTDECR registers can be cleared by writing to the BISTCR with CNTCLR bit set.
To change the pattern, the software writes to the BISTCR PATTERN field to select a new pattern.

- Far-end analog

The host software requests this mode by writing 08h to bits [7:0] of the TXBISTPD register. Corresponding BIST Activate FIS is sent to the device with the Pattern Definition field (bits [23:16]) of the first DWORD containing this value.

The operation proceeds as described in the far-end retimed test above.

- Far-end transmit only

The host software writes the required pattern DWORDs to the TXBISTD1 and TXBISTD2 registers. The host software initiates transmission of the BIST Activate FIS to the device by writing bits [7:0] of the TXBISTPD register with the value corresponding to the required mode: Bit 7 is set, bits 4, 3, 2, and 1 cleared, and bits 6, 5, and 2 are used to enable the following options:

- Bit 6 is set: Bypass ALIGN
- Bit 5 is set: Bypass scrambling
- Bit 2 is set: Primitive bit (refer to the SATA specification for more details)

The BIST Activate FIS is sent to the device with the Pattern Definition field (bits [23:16]) of the first DWORD containing this value.

After the device acknowledges the reception of this FIS with R_OKp, sata host disables the PHY receiver and transmitter (any received data is ignored by the Link Layer, transmitter is idle and maintains common mode bias per SATA 1.0a specification).

Loopback initiator BIST modes can be terminated either by the device when it signals COMINIT OOB condition (except the near-end analog mode), or by the host software when it sets SControl[0] = 1 (COMRESET).

Note : The device must support either PARTIAL or SLUMBER power modes for near-end analog loopback mode, otherwise it should be initiated with the device disconnected from the host PHY. It is not clear how the device responds if it does not support the requested BIST mode—with R_OKp and ignoring the request or with R_ERRp. In the former case, the host assumes the device has entered BIST mode and starts the test that fails.

Note : When BIST patterns are generated by sata host as an initiator, BIST patterns with different character sequences depending on starting running disparity, have a 50% chance of being the desired bit pattern on the high-speed differential pair. This is due to the fact that only one set of data values is used to generate the BIST pattern, regardless of current running disparity at the 8b/10b encoder

10.3 SATA Host Transport Layer

The Transport Layer shown in figure below consists of the following five main modules:

- Receive FIFO (RxFIFO)
- Transmit FIFO (TxFIFO)
- Transport Check module (TCHK)
- Transport State Machine module (TSM)
- Synchronization module (SYN)

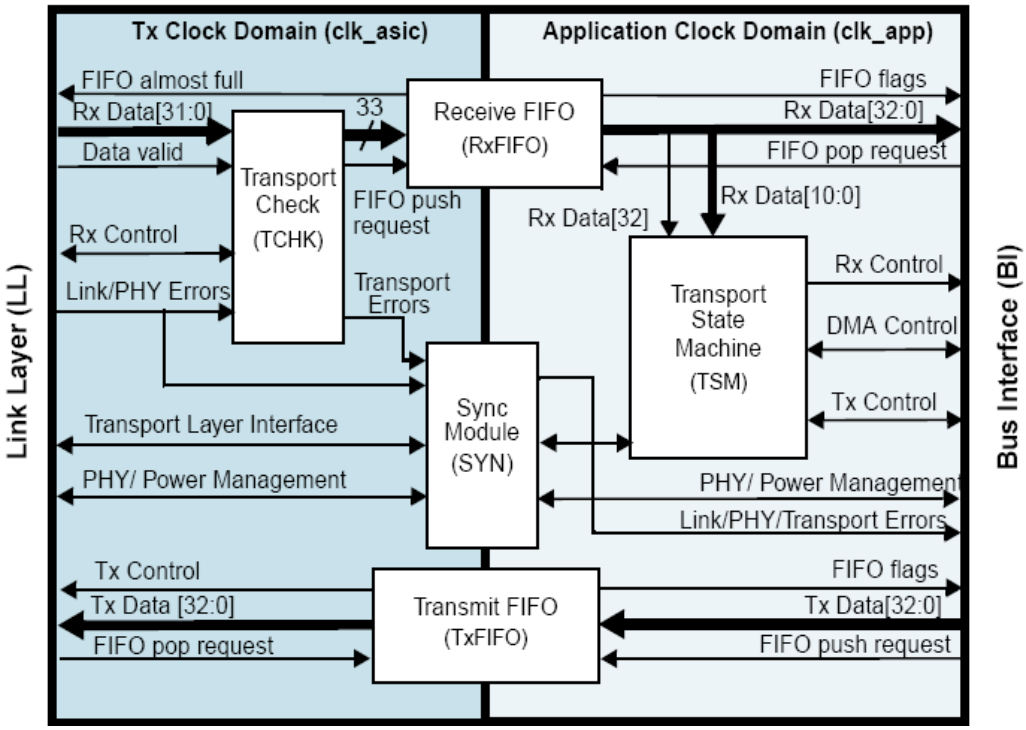


Figure 10.4 Transport Layer Block Diagram

The Transport Layer operates in two clock domains: transmit and application. Transmit clock is generated in the PHY and depends on the Link Layer data path width (valid frequency values are: 75 MHz, 150 MHz). The application clock is sourced from the system bus and depends on the application. Both transmit and receive data paths are 32-bits wide.

The Transport Layer operates in half-duplex mode; it either receives data in the form of FIS from the Link Layer or transmits data in the form of FIS to the Link Layer. The Transport Layer block provides FIS reception and transmission functions of the SATA Transport Layer. The actual FIS decomposition and FIS construction functions are implemented in the Bus Interface block, since all the ATA/SATA registers are located there. During reception the Transport Layer receives a new FIS from the Link Layer through the RxFIFO, decodes the FIS type, and instructs the Bus Interface to route the FIS payload data to the appropriate location, either ATA/SATA registers or the DMA engine. During transmission the Transport Layer instructs the Bus Interface to construct the appropriate FIS, and then passes it to the Link Layer through the TxFIFO for transmission. The Transport Layer block detects all the PHY, Link Layer and Transport Layer errors and passes them to the Bus Interface for setting the corresponding error register bits.

The Transport Layer processes one FIS at time on the transmit side, meaning only one FIS is allowed in the TxFIFO at a time. On the receive side, RxFIFO can potentially contain more than one FIS at a time. For example, when the device transmits several DMA Data FISs back-to-back with minimal delay, RxFIFO might still have the previous Data FIS while the next FIS is being received.

10.3.1 Transport Layer FIS Reception

The Link Layer starts frame reception to the Transport Layer by asserting the “first DWORD” signal and pushing incoming data DWORDs into the RxFIFO, provided it is not full. The RxFIFO “almost full” flag notifies the Link Layer to send HOLDp to the device to prevent RxFIFO overflow. Upon detecting EOFp, the Link Layer asserts an “End status” signal to indicate the end of the FIS. All Link Layer/PHY errors are valid at this time.

The FIS check and all the Transport Layer errors are handled by the Transport Check (TCHK) module, while the Transport Layer main control is done in the Transport State Machine (TSM) module.

The Transport Check module provides the following functions:

- Detects new FIS reception by the Link Layer based on the received control signals.
- Decodes the FIS type located in the least-significant byte of the first DWORD and checks its validity. If the type is invalid, it asserts an “Unrecognized FIS” error to the Link Layer and waits for the FIS end status. The following FIS types are supported by the host:
  - Register FIS
  - Set Device Bits FIS
  - PIO Setup FIS
  - DMA Activate FIS
  - First Party DMA Setup FIS
  - Data FIS
- Checks the FIS length according to the FIS type. If the length is invalid, it asserts a “Bad FIS” status signal to the Link Layer and TSM. This check includes all other Transport Layer errors (unrecognized FIS, protocol, transition, internal interface). The TCHK provides a Transport Layer error check status to the Link Layer at the end of the received FIS in the form of the “Good FIS/Bad FIS” acknowledgement.
- Detects an “End Status” signal assertion indicating the end of the current FIS from the Link Layer and passes all Link Layer/PHY/Transport Layer errors to the RxFIFO. Once this signal is asserted, it means either the Link Layer detected EOFp or any of the Link Layer/PHY errors. Bit 32 of the RxFIFO differentiates between the FIS data DWORDs and error/status DWORD.

The Transport State Machine module provides the following functions:

- Implements the host Transport Layer state machine according to the SATA spec with the exception of the FIS checking and error handling functions.
- Decodes the FIS type by reading the least-significant-byte of the first DWORD of the FIS.
- Detects the “End status/error” DWORD and checks for any Link Layer/PHY/Transport Layer errors. If any of the errors is detected, the FIS is discarded (if this is non-data FIS), and the corresponding SError register bit is set. If this is a data FIS, it must be passed to the system bus before the final status is reflected in the SError register. Bit 32 of the RxFIFO is used to indicate the “End status/error” DWORD.
- Generates the appropriate receive, transmit, and DMA control signals to the Bus Interface based on the received FIS and its state.
- Handles transfer termination requests originated from the Link Layer or Bus Interface.

10.3.2 Transport Layer FIS Transmission

The FIS transmission process is described as follows:

- The Bus Interface detects a request from the system bus (access to ATA registers) and notifies the TSM to enter a transmit state. The DMA data transmission is activated by the TSM after it receives DMA Activate FIS from the device. Only TSM and TxFIFO are involved in this process. TCHK is used only in the FIS reception.
- The Bus Interface forms the appropriate FIS and pushes it into the TxFIFO. The following FIS types are supported:
  - Register FIS: Control or Command type.
  - Data FIS: PIO or DMA type.
- The Link Layer uses negation of the TxFIFO “empty” flag to generate SOFp and begin frame transmission. Bit 32 of the TxFIFO is used to indicate the FIS “last DWORD” to the Link Layer. When the Link Layer sees this bit valid, it closes the frame with CRC and EOFp.
- The TSM waits for either positive or negative frame transmission acknowledgement from the Link Layer (Link Layer “handshake” error). Negative acknowledgement is generated when the device detects an error during the frame reception and signals it to the host Link Layer. TSM re-sends any non-data FIS to the device if transmit error is detected by asserting corresponding “Retry” signal to the Bus Interface.

10.3.3 Error Handling

All SATA Host errors are summarized in “SCR1”. The Link Layer accumulates all Link Layer/PHY errors during frame reception and presents them to the Transport Layer TCHK module with End Status signal asserted when it detects EOFp. PHY Not READY and Link Illegal Transition/Sequence errors terminate the current frame reception/transmission in progress and cause PHY/Link Layer reset and End Status assertion. During frame transmission, Link Layer detects R_ERRp/R_OKp from the device and passes this condition in the End Status DWORD.

Note: SCR1 Serror bits DIAG_B/DIAG_D are set only if the Link Layer detects errors in the data DWORD. If errors are detected in the primitive DWORD, INTPR register PRIMERR bit is set instead.

The TCHK module checks for Transport Layer errors in the received FIS if no PHY/Link Layer errors were detected. All errors from the PHY, Link Layer, and Transport Layer TCHK module are passed to the Transport Layer TSM module by way of RxFIFO in the “End status” DWORD (with bit 32 set).

All PHY, Link, and Transport errors are also passed to the SSR module through the APP_ASIC module directly for setting the appropriate bits in the SError register. This is done to ensure that error bits are reliably set in various error conditions. For example, if Data FIS is corrupted, this may result in potential DMA lock-up unless error that caused this condition is passed to the SSR module and software can act on it.

The TCHK module also detects the command status when a good Register or Set Device Bits FIS is received from the Device, and sets either the INTPR.CMD_ABORT bit if the Status.ERR bit (bit 16 of the first DWORD) is set, or the INTPR.CMD_GOOD bit if the Status.ERR bit is cleared. An interrupt is generated if the corresponding INTMR mask bit, CMD_ABORTM or CMD_GOODM, is set.

Caution: The Command Status interrupt with INTPR.CMD_ABORT and INTPR.CMD_GOOD is intended to aid software error recovery and is not intended for use in normal operation. If this interrupt is enabled in INTMR CMD_ABORTM and CMD_GOODM, the software must be able to handle its assertion before a normal Command Completion interrupt is triggered (when the Register or Set Device Bits FIS is processed by the TSM, the CDR7 Status or Sactive register is updated with new values, and the Internal IPF bit is set).

The TSM module detects Non-Recovered/Recovered Data Integrity errors when the FIS transfer completes, that is, when the receive FIS is completely read out of the RxFIFO and the transmit FIS status is received from the Link. An Internal Host Adapter error is detected in the Bus Interface.

PHY Internal errors from the Link layer (as indicated by the rx_phy_err assertion) are filtered out outside the FIS. Only errors related to the current receive FIS cause SError bits to be set. To enable errors inside the FIS or outside the FIS to be reflected in the SError register, software must set LLCR.ERREN bit.

The following is an example of error processing when an interrupt is asserted due to some error condition (SCR1/SError bit is set) during command execution:

- The command completes with a Command Abort status (CDR7 Status register == 0x51 and BSY/DRQ cleared). This applies to both Rx/read and Tx/write commands. In this case, the software simply retries the failed command.
- The command does not complete (CDR7 Status register BSY/DRQ are not cleared). This applies to Rx/read operation only, because some errors might result in a Data FIS with more or less data than expected causing a DMA lock-up. In this case, the software can initiate an RxFIFO flush by clearing DMACR.RXCHEN bit (see DMACR register description for details):
  - If BSY/DRQ is cleared, the command can be retried.
  - If BSY/DRQ is not cleared, a hard reset (COMRESET) is required.

10.3.4 Transfer Termination

Normal FIS transfer (reception or transmission) can be interrupted either by the device or the host. All the transfer termination cases are described as follows:

- Soft reset. Both receive and transmit operation in-progress can be terminated by the host software by setting the SRST bit in the Device Control register (CLR0) or issuing the ATAPI Device Reset command (08h). Both Link Layer and Transport Layer are returned to their Idle state. The Transport Layer sends the appropriate Register FIS to reset the device.
- Device terminates DMA transfer. When the host is sending DMA Data FIS to the device and the device decides to abort the current transfer, it may send the DMATp primitive to the host on the backchannel. The Link Layer notifies the Transport Layer of this condition.

This sets DMAT bit of the INTPR register and intrq is asserted if DMATM bit of the INTMR register is set. The host software may choose to ignore this request and complete the current transfer.

- Host terminates DMA transfer. When the device is sending DMA data FIS to the host and the host software decides to terminate this transfer, it would deactivate the DMA channel by clearing RXCHEN bit of the

DMACR register. The Transport Layer notifies the Link Layer of this condition similar to the soft reset (using SYNC escape mechanism) except no Register FIS is sent to the device.

10.3.5 Module Description

The following subsections describe the Transport Layer modules:

10.3.5.1 Receive/Transmit FIFO (Rx/TxFIFO)

Both receive and transmit FIFOs are used as temporary FIS buffers and for clock domain crossing.

Both FIFOs are reset on power-up either by the system bus reset signal, by the software setting SControl register bit 0 (COMRESET), or by the COMINIT condition.

The FIFO depth is 256 The RxFIFO “almost full” flag is set to 43 DWORDs to comply with the SATA HOLDp latency requirement: the Link Layer sends HOLDp on the backchannel when this flag is asserted to prevent RxFIFO overflow. Since the maximum Data FIS length does not exceed 2048 data DWORDs, the depth is limited to 2048.

Both FIFOs are memory-mapped in the SATA host address space. Data is read from the RxFIFO or written into the TxFIFO from the system bus master (for example, DMA controller) either in a single Read/Write cycle or burst, if there is enough data in the RxFIFO or room in the TxFIFO for a given burst size.

The RxFIFO “almost empty” and the TxFIFO “almost full” thresholds are set by the host software to the maximum burst size in DWORDs expected on the bus through the DBTSR register. The same value should be programmed in the DMA engine, otherwise an error bus response can be generated on the system bus in the situations when RxFIFO is empty or TxFIFO is full and the DMA burst size exceeds the values in the DBTSR.

10.3.5.2 Transport Check (TCHK)

The TCHK module receives data from the Link Layer in the form of 32-bit FIS DWORDs. With bit 32 cleared, the data is written to the RxFIFO. With bit 32 set, the final Link/PHY/Transport end status is written. If any error is detected in the Link/PHY layers, then the FIS is not checked for Transport Layer errors and the errors are passed to the RxFIFO in the end status DWORD. If no errors were detected in the Link/PHY, the following checks are done in the TCHK:

1. FIS length: Non-data FIS according to the FIS type. Data FIS should not exceed 2049 DWORDs, and DMA Data FIS should be at least 2 DWORDs long (including FIS type DWORD).

2. PIO Setup FIS transfer count: Should be non-zero, with an even byte count, and should not exceed 8192 bytes.

3. PIO Data FIS following the PIO Setup FIS with D = 1 (PIO read) DWORD count: Should match the transfer count.

4. PIO read protocol FIS sequence: Only Data FIS or end status if errors are expected after the PIO Setup FIS with D = 1. Any other FIS would be negatively acknowledged to the Link Layer.

5. DMA Setup FIS buffer offset: Bits 0 and 1 should be cleared and transfer count should be an even (not zero) number.

6. First Party DMA read protocol: DMA Setup FIS with D = 1 is followed either by Data FIS or Set Device Bits FIS or end status if error.

7. First Party DMA write protocol: DMA Setup FIS with D = 0 is followed by DMA Activate FIS (if A = 0) or Set Device Bits FIS or end status (if A = 1).

8. BIST Activate FIS is supported type only (See the “RXBISTPD” for more information).

9. FIS type is valid type, per the SATA specification.

10. RxFIFO push error for Data FIS: Detected when Link has valid data and RxFIFO is full (for example, device violates HOLD latency requirement).

The Transport State Transition error SError.DIAG_T bit is set when errors 1-8 are detected. The Unrecognized FIS Type

error SError.DIAG_F bit is set when error 9 is detected. The Protocol error SError.ERR_P bit is set when error 10 is detected. Note, that SError.ERR_P bit is also set if Serror register DIAG_T, DIAG_F, or DIAG_S bits are set. The TCHK asserts the appropriate positive or negative FIS acknowledgement signals to the Link Layer only if no Link/PHY errors are detected.

After a PIO or a DMA Data FIS end status is received, the RxFIFO “full” flag is checked before end status DWORD is written into the RxFIFO. If it is asserted, TCHK waits for the RxFIFO to become not full. This is done to prevent the possibility of writing the end status DWORD into the full RxFIFO during Data FIS reception.

10.3.5.3 Transport State Machine (TSM)

This module implements the Transport Layer state machine according to the SATA specification. It monitors RxFIFO for received FISs and generates appropriate control signals to the Bus Interface to decompose them on the received side. On the transmit side, the TSM monitors the state of the ATA registers to generate control signals to the Bus Interface to construct non-data FIS, or initiate Data FIS transmission. Refer to “Figure 10.3 shows how the Bus Interface Block Bus connects the SATA host The SATA Host Link Layer and the SATA Host Transport Layer to a system bus by way of the slave AHB interface.” for more details.

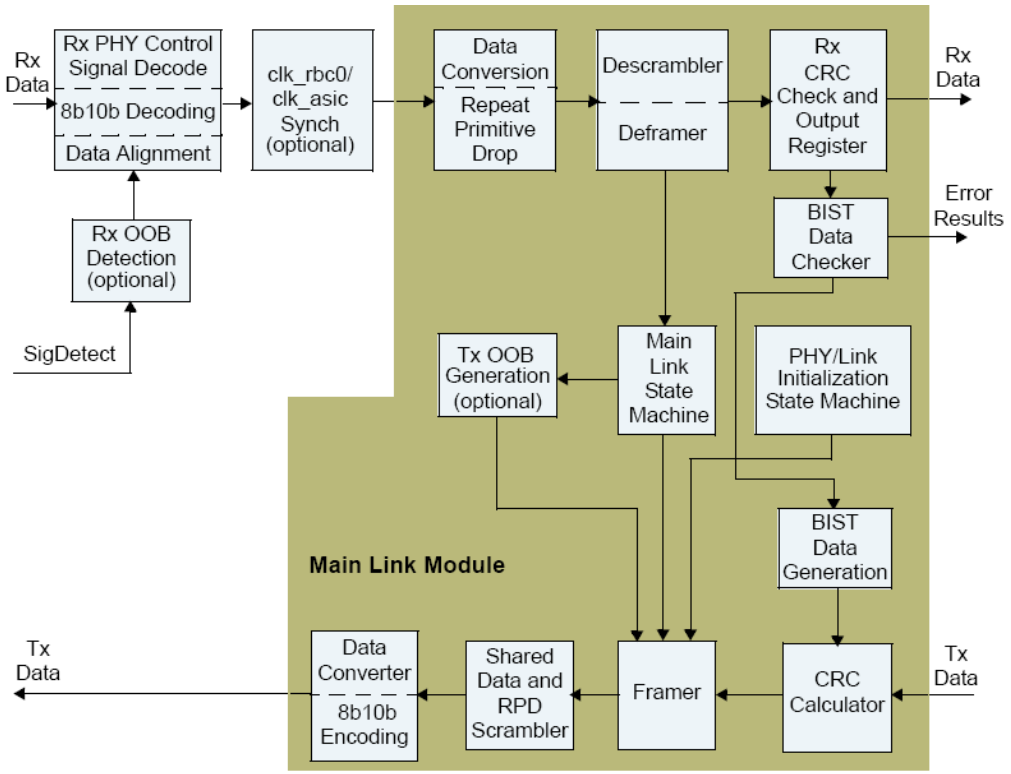
10.4 SATA Host Link Layer

This section discusses the following topics:

- Link Layer Overview
- Configurable PHY Interface
- PHY Initialization Details
- Link Layer Power Management Details

10.4.1 Link Layer Overview

Next figure shows a block diagram of Link Layer.



10.4.1.1 Features

The Link Layer features are as follows:

- Rx Data Buffer for recovered clock systems
- OOB signaling and system Initialization
- Gen2 speed negotiation when Tx OOB signaling included

- Frame negotiation and arbitration
- Envelope framing/deframing
- CRC calculating, insertion and checking
- 8b/10b encoding/decoding
- Flow control
- Frame acknowledgement and status reporting
- Data width conversions
- Data scrambling/descrambling for EMI reduction
- Repeat Primitive data transmission and reception handling
- ALIGN Primitive detection, dropping and data alignment
- Power management support

10.4.1.2 Operational Overview

On power-up, system reset or device hot-plug, the Link Layer transmits sequences of control data and ALIGN Primitives to the PHY. In that case, they are then forwarded to a device PHY as OOB signaling. In addition, the Link Layer detects OOB sequences.

These OOB sequences bring the PHY and device to an initialized condition. Once this occurs, the Link Layer passes a PHY Ready status to the Transport Layer and normal communication begins. The Link Layer receives requests from the Transport Layer to transmit data, in the form of a Frame Information Structure (FIS) comprised of DWORDs, to a device via the local PHY. The Link Layer in turn transmits the FIS by inserting Primitives, scrambling and optionally encoding the data, sending it to the PHY and waiting for status. When a status FIS is received, the Link Layer decodes, aligns and descrambles the data, removes Primitives and forwards the data to the Transport Layer. The Link Layer then notifies the Transport Layer of the ending transfer status. The Link Layer has no notion of the FIS content, other than its beginning and end points and CRC. Data alignment is performed on received FIS data via ALIGN Primitives. Flow control is also achieved on FIS going in either direction via HOLD Primitives.

In addition, the Link Layer receives requests from the Transport and PHY Layers to go into and out of power management modes. Power management is achieved by notifying the PHY of a partial or slumber condition and then disabling normal data transmission on PHY Rx and Tx interfaces until a wake-up request from Transport Layer or remote device via the PHY is seen. Power management is controlled via Partial and Slumber requests as described in the SATA specifications.

The Initialization State Machine controls the Link Layer, PHY and device system initialization. The main Link Layer State Machine controls FIS traffic, flow control and error detection and status reporting. FIS traffic is generated and disassembled via Framer and Deframer modules. The Link Layer also performs CRC calculations on FIS, as well as scrambling and optionally encoding the data. Decoding of received FIS is performed in the clk_rbc0 clock domain due to the fact that the incoming FIS is on an asynchronous, but frequency locked clock of the same rate as the clk_asic clock domain. Note that 8b/10b encoding and decoding are only performed in the Link Layer.

The Link Layer receives data on either clk_rbc0, recovered from the incoming data stream by the PHY, or on clk_asic. This single receive clock is then used in this module to decode data and control signals from the PHY and pass it to the rest of the Link Layer. Data is passed through a synchronizing Datastream FIFO. ALIGN Primitives are also detected and dropped in the front end of the receiver as a means of guaranteeing no Datastream FIFO overruns, when a Datastream FIFO is included. ALIGN Primitives are also used to synchronize to the data stream in the PHY by triggering data realignment where necessary.

Finally, ALIGNs are required by the Tx OOB initialization state machine to complete initialization, per the SATA specifications. For this reason, the PHY must indicate the presence of at least two ALIGNs after the Link Layer detects the release of COMWAKE. Otherwise the Link Layer is not able to complete initialization and begin normal operation. This is required regardless of whether or not the PHY drops ALIGNs at any other time.
This functional specification attempts to avoid redundancy with the ‘High Speed Serialized AT Attachment’ document. Please refer to it for standard specifications and descriptions. This document pertains to specific implementation related operation and details.

10.4.2 PHY Interface

Many of the features of the SATA Host are detailed in the SATA specifications and are not repeated here.

10.4.2.1 Rx and Tx Data

The Link Layer can receive data on a clock recovered from the incoming data stream (clk_rbc0), or the data can already be synchronized into the clk_asic clock domain. When data is presented to the Link Layer on a recovered clock, the Link Layer synchronizes data into the clk_asic clock domain via a Datastream FIFO. In addition, the Link Layer has the capability of dealing with data that is misaligned and/or has “bubbles” in the Rx data. Each byte of incoming data has an associated phy_rx_data_vld bit associated with it, regardless of the total Rx data width. This, in combination with the ALIGN_MODE

being set to 'Misaligned', provides complete data alignment in any situation.

When data is guaranteed to be aligned and does not contain bubbles, setting ALIGN_MODE to 'Aligned' eliminates the logic for this extra circuit in the design. In this case, phy_rx_data_vld can either be connected 'high', to phy_sig_det, or to phy_ready (depending on the particular PHY), as long as data is valid at all times the signal is asserted.
Finally, all status or control signals associated with Rx or Tx data have information slots for each byte. In designs where information is only valid for the Least Significant Byte slot, the other byte slot information can be permanently tied 'high' or 'low', depending on the information. One example of this would be to connect phy_comma_det[1] 'low' in two-byte Rx data systems where the comma character is guaranteed to only ever fall into the phy_comma_det[0] byte slot. Note that phy_tx_data_vld is comprised of only one bit because all Tx data bytes are valid and aligned when available from the Link Layer to the PHY.

10.4.2.2 8b/10b Encoding and Decoding

Note that the Tx data and K-characters are always valid and aligned with the K-character in the Least Significant Byte slot, so only one bit is required for phy_tx_kch.

10.4.3 PHY Initialization Details

Please refer to the 'Out of band signaling' section of the SATA specifications for the following descriptions. The PHY/device initialization signaling is dependent on whether or not the Link Layer generates and detects the proper OOB sequences or whether the PHY generates and detects them.

10.4.3.1 Link Layer Tx OOB Initialization Sequence Details

Note : In order for the Link Layer to generate the Tx OOB initialization sequences, at least two ALIGN Primitives must be indicated and flagged to the Link Layer by the PHY via the phy_comma_det signal. This must occur after the release of COMWAKE, per the SATA initialization specifications. Otherwise the Link Layer is not able to complete initialization and begin normal operation. This is required regardless of whether or not the PHY drops ALIGNs at any other time. In addition, if data can ever become misaligned from the PHY, ALIGN Primitives are required in order for the Link Layer to realign the data. Finally, ALIGNs must be returned to the SATA host for all BIST modes to function properly.

The Link Layer Tx OOB initialization sequence is depicted in figure below. Not all details are shown, but the following summarizes the flow:

1. PHY Reset state is entered during a system asynchronous-reset or a host hard-reset (COMRESET). This causes the phy_reset output signal to become active and can be used to reset the PHY. The PHY reset is active for 12 Gen1 DWORDs of time, or 320ns.

2. Upon exiting PHY Reset state, the PHY speed is always set to Gen1 speed, regardless of speed negotiating for Gen1 only.

3. After a PHY reset, and anytime an unsolicited COMINIT has been detected, the PHY Wait state is entered in order to check and wait for PHY calibration.

4. After the PHY Wait state, communication with a device is attempted by transmitting a COMRESET sequence, followed by waiting for a COMINIT sequence from the device. This takes place in a polling routine. During the Wait Cominit state, if a COMINIT has not been detected in 14ms, the state machine loops back and restarts from the PHY Wait state indefinitely.

5. Once a COMINIT has been detected, normal OOB sequences then take place, interrupted only by an unsolicited COMINIT; a COMINIT that was not expected.

6. After completion of the OOB sequence, if Gen2 is enabled and has not already been attempted and failed, phy_spd_sel is asserted to the PHY to change the clk_asic rate to Gen2. The host then begins waiting for ALIGNs.

7. At the beginning of awaiting ALIGNs, the speed is changed to Gen2 if Gen2 is to be negotiated for, otherwise the speed stays at Gen1. During the Await ALIGN state, if an ALIGN has not been detected by the time 32K Gen1 DWORDs of D10.2 characters have been transmitted, the Link Layer will move to the PHY Wait state. In addition, if Gen2 speed was attempted, an internal flag is set to force the host to stay at Gen1 speeds after the next OOB sequence has been complete, unless one of the following occurs:

- Asynchronous system reset
- Port reset
- An unsolicited COMINIT has been detected from the device
- Gen2 speed disabled and re-enabled (programmed to Gen1, then Gen2, followed by a host COMRESET)

7. If an ALIGN has been detected before 32K Gen1 DWORDs of D10.2 have been transmitted, the Link Layer moves to the Send ALIGN state, followed by the Init Complete state, once three non-ALIGN Primitives have been detected. Note that while the Link Layer is generating D10.2 characters when Gen2 speed has been selected, Gen1 D10.2 characters are emulated by doubling the transmitted data, i.e. “1100110011”, as opposed to “1010101010”.

8. Once initialization is complete, the Ready state is entered and the Main Link state machine takes control.

9. Finally, the Partial and Slumber states are only entered after a Power Mode has been set. This disables the Tx interface until either the host or device requests a wake-up. For further information see the “Link Layer Power Management Details”.

10.4.3.2 Link Layer Tx OOB Sequence Generation

When the Link Layer generates the Tx OOB sequences, the Tx interface connections are different than when the PHY generates them. Next figure depicts a small portion of an initialization phase, which includes the first part of a COMRESET condition, followed by a portion of a COMWAKE.The actual OOB sequences are created by sending ALIGN Primitives in conjunction with deassertion of the phy_tx_enable signal. This causes the Tx OOB data and NULL sequences required by the SATA PHY specifications. Note that this diagram is not the complete initialization sequence. In a complete initialization sequence, there are multiple bursts of COMRESTs and COMWAKES, followed by a d10.2 stream, followed by an ALIGN stream and then finally normal data. The phy_tx_data_vld is only used for qualifying data in systems that require it, but is unrelated to the OOB sequencing itself.

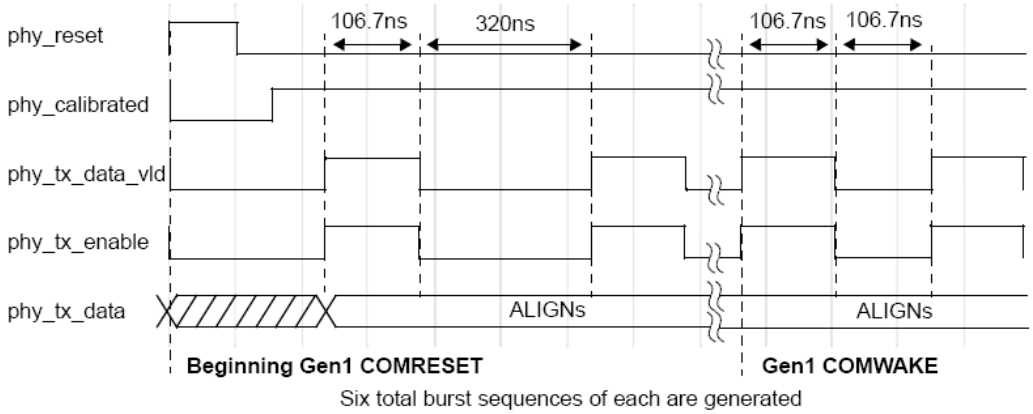


Figure 10.5 Link Layer Generated Tx OOB Signaling

10.4.3.3 Link Layer Rx OOB Sequence Detection

When the Link Layer detects the Rx OOB sequences, the Rx interface connections are different than when the PHY detects them. Next figure depicts a small portion of an initialization phase, which includes the first part of a COMINIT condition, followed by a portion of a COMWAKE.

The actual OOB sequences are comprised of specifically timed ALIGN Primitives in combination with negation of the phy_sig_det signal (indicating NULL on the differential pair). However, only the phy_sig_det signal is used for qualifying Rx OOB sequences by the Link Layer.

The next figure does not show the complete initialization sequence. In a complete initialization sequence, the host receives multiple bursts of COMINITs and COMWAKEs, followed by an ALIGN stream, and then finally normal data.

The Link Layer qualifies valid COMINIT and COMWAKE sequences according to the SATA specifications, with the exception that the clk_rxoob frequency can introduce variances from the required ranges of OOB sequence spacing. The actual COMINIT and COMWAKE sequences are calculated and qualified by the Link Layer as described in the following subsections. The upper ranges of COMINIT and COMWAKE are forced to be always less than the upper limit to guarantee removal of the appropriate condition. Please see the application note and example Rx OOB Detection Versus Clock Frequency table in section “Example Calculated COMWAKE, COMINIT, and Data Burst Lengths” for further important information.

Obviously, the faster the clk_rxoob, the less the sampling error is (30 MHz is the absolute minimum Rx OOB clock frequency supported). Finally, the phy_rx_data_vld is only used for qualifying data in systems that require it, but is unrelated to the OOB sequencing itself.

Caution: If the Rx OOB clock frequency is less than 60 MHz, there are stringent requirements.

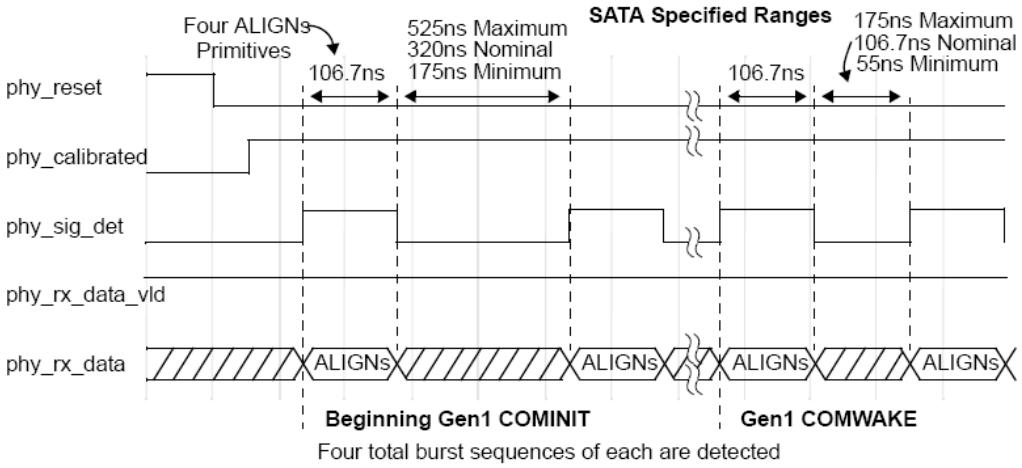


Figure 10.6 Link Layer Rx OOB Detection

10.4.4 Link Layer Power Management Details

Power management OOB detection and generation sequences occur in the same signaling format as described in “PHY Initialization Details”. However, they follow the required sequencing specified in the SATA Power management specifications. All data on the Tx channel is disabled until the host or device requests a wake-up condition. In addition, Rx data is not allowed to propagate in the SATA host Rx datapath until one of these conditions has been detected and the PHY and device have been fully initialized per requirements.

A power down and wake up is depicted in next figures, excluding actual OOB signaling, which are dependent on OOB Modes. Note that ‘OOB’ on Tx and Rx Data indicates that OOB sequences are present. Also note that if clk_asic is slowed down during power mode, phy_calibrated should be pulled low to indicate this. The link does not proceed until phy_calibrated goes high again.

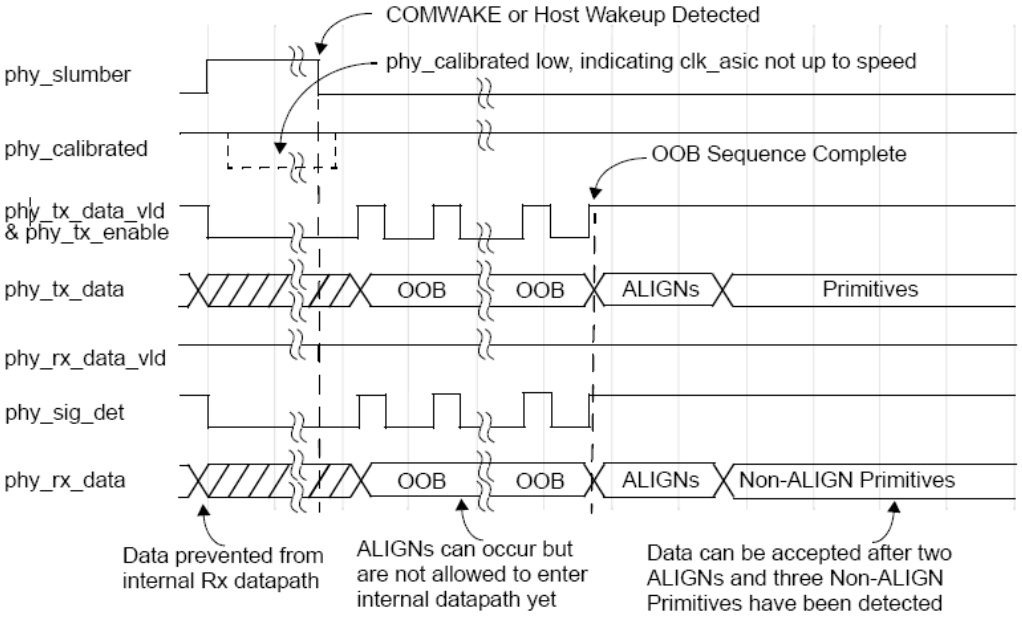


Figure 10.7 Power Mode Example: Rx & Tx OOB In Link

10.5 Register Descriptions

Table 10.1 SATA Host Register Map (Base Address = 0xF0560000)

Shadow ATA/ATAPI Registers: CDR—Command Block, CLR—Control Block					
Name	**Address**	**BITS**	**Type**	**Reset**	**Description**
CDR0	0x00	16	RO/WO	-	Data register in PIO mode Dependencies: Read-only for PIO read/receive operation, write-only for PIO write/transmit operation
CDR1	0x04	8 8 8	RO WO WO	0xFF 0x00 0x00	Error Register Feature Register(current value) Feature Expanded Register(previous value)
CDR2	0x08	8 8	R/W	0xFF	Sector count register (current value) Sector count expanded register (previous value)
CDR3	0x0C	8 8	R/W	0xFF	Sector number register (current value) Sector number expanded register (previous value)
CDR4	0x10	8 8	R/W	0xFF	Cylinder low register (current value) Cylinder low expanded register (previous value)
CDR5	0x14	8 8	R/W	0xFF	Cylinder high register (current value) Cylinder high expanded register (previous value)
CDR6	0x18	8	R/W	0xEF	Device/ Head register
CDR7	0x1C	8 8	RO WO	0x7F 0x00	Status register Dependencies: Value is 0x7F on power-up, then 0x80 when device presence is detected via PHY READY condition. Command register
CLR0	0x20	8 8	RO WO	0x7F 0x00	Alternative status register Dependencies: Value is 0x7F on power-up, then 0x80 when device presence is detected via PHY READY condition. Device control register
Serial ATA Registers					
Name	**Address**	**BITS**	**Type**	**Reset**	**Description**
SCR0	0x24	32	RO	0x0	SStatus Register
SCR1	0x28	32	R/W	0x0	SError Register
SCR2	0x2C	32	R/W	0x0	SControl Register
SCR3	0x30	32	R/W	0x0	SActive Register
SCR4	0x34	32	R/W	0x0	Snotification Register
SCR5-SCR15	0x38-0x60	32	See description	0x0	Reserved for SATA Dependencies: Reads to these locations return zeros; writes have no effect
SATA Host-Specific: DMA Registers					
Name	**Address**	**BITS**	**Type**	**Reset**	**Description**
FPTAGR	0x64	32	RO	0x0	First Party DMA tag Register
FPBOR	0x68	32	RO	0x0	First Party DMA buffer offset Register
FPTCR	0x6C	32	RO	0x0	First Party DMA transfer count Register
DMACR	0x70	32	R/W	0x0	DMA Control Register
DBTSR	0x74	32	R/W	0x0014_0010	DMA Burst Transaction Size Register
SATA Host-Specific: Interrupt Registers					
Name	**Address**	**BITS**	**Type**	**Reset**	**Description**
INTPR	0x78	32	R/W	0x0	Interrupt Pending Register
INTMR	0x7C	32	RO	0x0	Interrupt Mask Register
ERRMR	0x80	32	RO	0x0	Error Mask Register
SATA Host-Specific: Link/PHY Registers					
Name	**Address**	**BITS**	**Type**	**Reset**	**Description**
LLCR	0x84	32	R/W	0x0000_0007	Link Layer Control Register
SATA Host-Specific: BIST Registers					
Name	**Address**	**BITS**	**Type**	**Reset**	**Description**
RXBISTPD	0x90	32	RO	0	Received BIST Pattern Definition Register
RXBISTD1	0x94	32	RO	0	Received BIST Data DWORD 1 Register
RXBISTD2	0x98	32	RO	0	Received BIST Data DWORD 2 Register
TXBISTPD	0x9C	32	R/W	0	Transmit BIST Pattern Definition Register
TXBISTD1	0xA0	32	R/W	0	Transmit BIST Data DWORD 1 Register
TXBISTD2	0xA4	32	R/W	0	Transmit BIST Data DWORD 2 Register
BISTCR	0xA8	32	R/W	0	BIST Control Register
BISTFCTR	0xAC	32	RO	0	BIST FIS Count Register

BISTSR	0xB0	32	RO	0	BIST Status Register
BISTDECR	0xB4	32	RO	0	BIST DWORD Error Count Register
SATA Host-Specific: Miscellaneous Registers					
Name	**Address**	**BITS**	**Type**	**Reset**	**Description**
TESTR	0xF4	32	R/W	0	Test register
VERSIONR	0xF8	32	RO	0x3139_302A	Version register
IDR	0xFC	32	RO	0	ID register
-	0x100-0x3FF	-	-	-	Unimplemented locations. Access to these locations is illegal and results in bus ERROR response if RETURN_ERR_RESP = 1.
DMAR	0x400-0x7FC	16/32	RO/WO	-	FIFO locations in DMA mode Dependencies: Read-only for DMA read/receive operation. Write-only for DMA write/transmit operation.

Shadow ATA/ATAPI registers (locations CDR1 to CDR7, CLR0) support 8, 16, or 32-bit transfer sizes with 32bit aligned addresses. Non-aligned address accesses to these locations is illegal and return ERROR bus response if RETURN_ERR_RESP = 1. Access to these registers should be done according to the ATA/ATAPI specification protocol. For example: writing to the CDR1-CDR6 registers is prohibited when either the BSY or DRQ bit is set in the Status (CDR7) register. Any write to the Command (CDR7) register is ignored (bus access is acknowledged normally, but no action is taken) in this case except Device Reset command.

CDR1–CDR5 registers are used for 48-bit addressing and are implemented as two-byte FIFOs. For example, first write to CDR1 results in the Feature register being written, second write to CDR1 results in the Feature register being written with a new value, while its previous value is transferred to the Feature expanded register.

During 16 or 32-bit read accesses to CDR2-CDR5 locations, bits [15:8] return corresponding register value depending on the Device Control register HOB bit (see register description). Bits [31:16] are treated as reserved. Write accesses have no effect, read accesses return zeros.

Other register locations—SATA and SATA host-specific registers (except DBTSR) support 8, 16, or 32-bit transfers with addresses naturally-aligned according to the transfer size.

DBTSR registers support only 16 or 32-bit transfer sizes for write accesses, 8-bit writes are ignored.

Note
Only the DMADR location in DMA mode supports burst transfers, all other locations should be accessed using single transfers Unless otherwise noted, all registers are reset on power-up or a COMRESET condition (SControl[0] = 1).

10.5.1 SATA Host Controller Registers

Data Register (CDR0) **0xF0560000**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDR0															

This register is used to transfer data from host-to-device and from device-to-host in PIO.

Bit	Name	Initial	Type	Description
15-0	CDR0	-	RO/WO	This register can only be accessed by the host software when the SATA Host is in the corresponding PIO mode as follows: • Read, when the PIO Setup FIS with D = 1 is followed by the Data FIS • Write, when the PIO Setup FIS with D = 0 is received. During PIO read, the software performs a series of reads from this location; during PIO write, the software performs a series of writes to this location. Only single 16-bit bus transfers are supported in this mode.

Error/Feature/Feature Expanded Register (CDR1) **0xF0560004**

7	6	5	4	3	2	1	0
Error/Feature/Feature_exp							

This location is used as one of the following:

- Error register when read
- Feature/ Feature Expanded registers when written. Implemented as two-byte FIFO..

Bit	Name	Initial	Type	Description

7-0	Error	0xFF	RO	Error register contains error/diagnostic information from the device.
7-0	Feature	0	WO	Current value of the Feature register. Determines the specific function of the SET FEATURES commands.
7-0	Feature_exp	0	WO	Previous value of the Feature register (used for 48-bit addressing). Dependencies: The value is pushed from the Feature register every time CDR1 is written.

Sector Count/ Sector Count Expanded Register (CDR2) 0xF0560008

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Seccnt/ Seccnt_exp								Seccnt/ Seccnt_exp							

These two 8-bit registers are implemented as a two-byte FIFO and contain the number of sectors for ATA/ATAPI data commands or command-specific parameters on some non-data commands.

Bit	Name	Initial	Type	Description
7-0	Seccnt	0xFF	R/W	Current value of the CDR2 register when written. Dependencies: Can be read when Device Control register HOB bit is cleared (CLR0.HOB = 0).
7-0	Seccnt_exp	0xFF	R/W	Previous value of the CDR2 register (used for 48-bit addressing) pushed from the Seccnt register every time CDR2 is written. Dependencies: Can be read when Device Control register HOB bit is set (CLR0.HOB = 1).
15-8	Seccnt	-	RO	Current value of the CDR2 when read Dependencies: Device Control register HOB bit is set (CLR0.HOB = 1)
15-8	Seccnt_exp	-	RO	Previous value of the CDR2 register when read Dependencies: Device Control register HOB bit is cleared (CLR0.HOB = 0).

Sector Number/Sector Number Expanded Register (CDR3) 0xF056000C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Secnum/ Secnum_exp								Secnum/ Secnum_exp							

These two 8-bit registers are implemented as a two-byte FIFO and contain the start sector number (CHS mode) or LBA low value (LBA mode bits [7:0], [31:24]).

Bit	Name	Initial	Type	Description
7-0	Secnum	0xFF	R/W	Current value of the CDR3 register when written. Dependencies: Can be read when Device Control register HOB bit is cleared (CLR0.HOB = 0).
7-0	Secnum_exp	0xFF	R/W	Previous value of the CDR3 pushed from the Secnum every time CDR3 is written. Contains LBA [31:24] bits. Used for 48-bit addressing.. Dependencies: Can be read when Device Control register HOB bit is set (CLR0.HOB = 1).
15-8	Secnum	-	RO	Current value of the CDR3 when read Dependencies: Device Control register HOB bit is set (CLR0.HOB = 1)
15-8	Secnum_exp	-	RO	Previous value of the CDR3 register when read Dependencies: Device Control register HOB bit is cleared (CLR0.HOB = 0).

Cylinder Low/Cylinder Low Expanded Register (CDR4) 0xF0560010

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cyllow / Cyllow _exp								Cyllow / Cyllow _exp							

These two 8-bit registers are implemented as a two-byte FIFO and contain cylinder number low byte (CHS mode) or LBA mid value (LBA mode bits [15:8], [39:32]).

Bit	Name	Initial	Type	Description
7-0	Cyllow	0xFF	R/W	Current value of the CDR4 register when written. Dependencies: Can be read when Device Control register HOB bit is cleared (CLR0.HOB = 0).
7-0	Cyllow _exp	0xFF	R/W	Previous value of the CDR4 pushed from the Cyllow every time CDR4 is written. Contains LBA [39:32] bits. Used for 48-bit addressing.. Dependencies: Can be read when Device Control register HOB bit is set (CLR0.HOB = 1).
15-8	Cyllow	-	RO	Current value of the CDR4 when read Dependencies: Device Control register HOB bit is set (CLR0.HOB = 1)
15-8	Cyllow _exp	-	RO	Previous value of the CDR4 register when read Dependencies: Device Control register HOB bit is cleared (CLR0.HOB = 0).

Cylinder High/Cylinder High Expanded Register (CDR5) 0xF0560014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cylhigh / Cylhigh _exp								Cylhigh / Cylhigh _exp							

These two 8-bit registers are implemented as a two-byte FIFO and contain cylinder number high byte (CHS mode) or LBA high value (LBA mode bits [23:16], [47:40]).

Bit	Name	Initial	Type	Description
7-0	Cylhigh	0xFF	R/W	Current value of the CDR5 register when written. Contains LBA[23:16] bits. Dependencies: Can be read when Device Control register HOB bit is cleared (CLR0.HOB = 0).
7-0	Cylhigh _exp	0xFF	R/W	Previous value of the CDR5 pushed from the Cylhigh every time CDR5 is written. Contains LBA [47:40] bits. Used for 48-bit addressing.. Dependencies: Can be read when Device Control register HOB bit is set (CLR0.HOB = 1).
15-8	Cylhigh	-	RO	Current value of the CDR5 when read Dependencies: Device Control register HOB bit is set (CLR0.HOB = 1)
15-8	Cylhigh _exp	-	RO	Previous value of the CDR5 register when read Dependencies: Device Control register HOB bit is cleared (CLR0.HOB = 0).

Device/Head Register (CDR6) 0xF0560018

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cylhigh / Cylhigh _exp								Cylhigh / Cylhigh _exp							

This register selects the device and contains command-dependent information.

Bit	Name	Initial	Type	Description
7	-	1	R/W	Obsolete bit
6	LBA	1	R/W	Logical Block Addressing • 1'b0: CHS Mode • 1'b1: LBA Mode
5	-	1	R/W	Obsolete bit
4	Dev	0	R/W	Device select bit with the following values: • 1'b0: Device 0 (Master) • 1'b1: Device 1 (Slave) This bit should always be cleared (Dev = 0) since SATA Host implements Master-only emulation. If this bit is set, any write to the Command register is ignored, except for a EXECUTE DEVICE DIAGNOSTIC command. A read from the Status/Alternative Status register returns 0x00. (See the ATA/ATAPI specification.) Note: this bit is not updated when Register FIS is received from the device. No device can change the state of this bit. This bit is cleared by one of the following conditions: • power-up • SControl[0] = 1 (COMRESET) • device signals COMINIT • SRST bit set in the Device Control register • EXECUTE DEVICE DIAGNOSTIC command written to the Command register
3-0	Head	4'b1111	R/W	Head number or other command-dependent information

Command/Status Register (CDR7) 0xF056001C

7	6	5	4	3	2	1	0
Command							
BSY	DRDY	DWF	SERV	DRQ	CORR	IDX	ERR

This location is used as one of the following:

- Command register when written (Reset: 0x00)
- Status register when read (Reset: 0x7F)

Bit	Name	Initial	Type	Description
7-0	Command	0	WO	Contains command code for device to execute. A write to this register sets the BSY bit in the Status register (BSY = 1). This register is written last to initiate command execution. Command Register FIS is sent to the device every time this register is written and if both BSY and DRQ bits of the Status register are cleared.

Status is a read-only 8-bit register and can be written only by the device with either the Register or the Set Device Bits FIS. Read access to the Status register clears the IPF, which negates the ATA interrupt request signal (intrq) to the system bus. Reset occurs on power-on.

Bit	Name	Initial	Type	Description
7	BSY	0	RO	Busy bit Indicates the device is busy when set. All other Command Block registers are invalid. When BSY and DRQ bits are cleared by the device upon reception of the Register FIS, the host software can access any of the Command Block registers. This bit is cleared on power-on or a COMINIT condition from the device (resulting in Status = 0x7F) and set in the following cases: • Device presence is detected via PHY READY signal assertion (this condition results in Status = 0x80) • SControl[0] = 1 (COMRESET condition) • A new command is written to the Command register when BSY = 0 and DEV = 0, or EXECUTE DEVICE DIAGNOSTIC command is written when BSY = 0 and DEV = 1

				• DEVICE RESET command is written to the Command register • SRST bit is set in the Device Control register
6	DRDY	1	RO	Device Ready (device-specific) It is not required that DRDY bit is set.
5	DWF	1	RO	Drive Write fault (device-specific)
4	SERV	1	RO	Service—used in overlap and queued commands (device specific)
3	DRQ	1	RO	Data Request—asserted when device is ready to transfer data
2	CORR	1	RO	Obsolete bit
1	IDX	1	RO	Obsolete bit
0	ERR	1	RO	Error when high (Error register contains further information)

Alternative Status/Device Control Register (CLR0) 0xF0560020

7	6	5	4	3	2	1	0
HOB					SRST	nIEN	

This location is used as one of the following:

- Device Control register when written
- Alternative Status register when read. Contains the same value as the Status register (CDR7), except the IPF is not cleared when the CLR0 is read.

Bit	Name	Initial	Type	Description
7	HOB	0	WO	This bit is used to read expanded registers (CDR2–CDR5) when set • 1’b1: CDR2 Seccnt_exp register is read • 1’b0: CDR2 Seccnt register is read The HOB bit is cleared after the write access to any Command Block register (CDR1–CDR7)
6-3	-	0	WO	Reserved
2	SRST	0	WO	Soft reset bit • 1’b1: Device is reset • 1’b0: Device is not reset Control register FIS is transmitted to the device every time the state of this bit is changed
1	nIEN	0	WO	Interrupt enable bit • 1’b0: Enable (intrq is asserted if IPF = 1) • 1’b1: Disable (intrq is negated) The nIEN bit is always cleared in all the Host-to-Device Register FISes for better compatibility
0	-	0	WO	Reserved

SStatus Register (SCR0) **0xF0560024**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				IPM				SPD				DET			

This register contains the current state of the interface and host adapter. It is updated continuously and asynchronously by the host adapter. Writes to this register result in bus error response. Resets on power-on.

Bit	Name	Initial	Type	Description
31-12	-	0		Reserved
11-8	IPM	0	RO	This value indicates the current interface owner management state: • 4'b0000: Device not present or communication not established • 4'b0001: Interface in active state • 4'b0010: Interface in PARTIAL power management state • 4'b0110: Interface in SLUMBER power management state All other values reserved.
7-4	SPD	0	RO	This value indicates the negotiated interface communication speed established: • 4'b0000: No negotiated speed (device not present or communication not established) • 4'b0001: Generation 1 communication rate negotiated • 4'b0010: Generation 2 communication rate negotiated All other values reserved.
3-0	DET	0	RO	This value indicates the interface device detection and PHY state: • 4'b0000: No device detected and PHY communication is not established • 4'b0001: Device presence detected but PHY communication not established (PHY COMWAKE signal is detected). • 4'b0011: Device presence detected and PHY communication established (PHY READY signal is detected). • 4'b0100: PHY in offline mode as a result of the interface being disabled or running in a BIST loopback mode All other values reserved

SError Register (SCR1) **0xF0560028**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				DIAG_A	DIAG_X	DIAG_F	DIAG_T	DIAG_S	DIAG_H	DIAG_C	DIAG_D	DIAG_B	DIAG_W	DIAG_I	DIAG_N
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				ERR_E	ERR_P	ERR_C	ERR_T							ERR_M	ERR_I

This register contains supplemental SATA interface error information to complement the error information available in the Shadow Error register. The register represents all the detected errors accumulated since the last time the SError register was cleared. The set bits in the SError register indicate that the corresponding error condition became true one or more times since the last time this bit was cleared. The set bits in the SError register are explicitly cleared by a write operation to the register, or a reset operation (power-on or COMRESET). The value written to clear the set error bits should have ones encoded in the bit positions corresponding to the bits that are to be cleared. The reset value of each bit described in the following table is 0.

Bit	Name	Initial	Type	Description
31-28	-	0		Reserved
27	DIAG_A	0	R/W	Port Selector Presence detected. Note: Port selector detection is not supported and this bit always reads zero.
26	DIAG_X	0	R/W	Exchanged error. This bit is set when Phy COMINIT signal is detected.
25	DIAG_F	0	R/W	Unrecognized FIS type. This bit is set when the Transport Layer receives a FIS with good CRC, but unrecognized FIS type.
24	DIAG_T	0	R/W	Transport state transition error. This bit is set when the Transport Layer detects one of the following conditions: (See “Transport Check (TCHK)” for more details) • Wrong sequence of received FISes • PIO count mismatch between the PIO Setup FIS and the following Data FIS • Odd PIO/DMA byte count or DMA buffer offset • Wrong FIS length
23	DIAG_S	0	R/W	Link sequence (illegal transition) error. This bit is set when the Link Layer detects an erroneous Link state machine transition.
22	DIAG_H	0	R/W	Handshake error. This bit is set when the Link Layer receives one or more R_ERRp handshake responses form the device after frame transmission.
21	DIAG_C	0	R/W	CRC error. This bit is set when the Link Layer detects CRC error in the received frame.
20	DIAG_D	0	R/W	Disparity error. This bit is set when the Link Layer detects incorrect disparity in the received data.
19	DIAG_B	0	R/W	10b to 8b decoder error. This bit is set when the Link Layer detects 10b to 8b decoder error in the received data.
18	DIAG_W	0	R/W	Comm Wake. This bit is set when Phy COMWAKE signal is detected.
17	DIAG_I	0	R/W	PHY internal error. This bit is set when the PHY detects an internal error, as indicated by the assertion of the phy_rx_err input. Note: Setting this bit is controlled by the LLCR.ERREN bit: • If ERREN==0 (default), only errors occurring inside the received FIS set the DIAG_I bit. • If ERREN==1, any error inside or outside the FIS sets the DIAG_I bit.
16	DIAG_N	0	R/W	PHYRdy change. This bit is set when Phy READY signal state is changed.
15-12	-	0		Reserved
11	ERR_E	0	R/W	Internal host adapter error. This bit is set when any one of the illegal accesses is attempted on the AHB bus (see “AHB Error Conditions”), or dma_finish_tx is asserted, but no data had been written during DMA write operation.
10	ERR_P	0	R/W	Protocol error. This bit is set when any of the following error conditions is detected: • Transport state transition error (DIAG_T) • Unrecognized FIS type (DIAG_F) • Link sequence error (DIAG_S) • RxFIFO overflow condition
9	ERR_C	0	R/W	Non-recovered persistent communication or data integrity error. This bit is set when PHY READY signal is negated (PHY Not Ready condition) due to the loss of communication with the device or problems with interface, but not after the transition from active to PARTIAL or SLUMBER power management state.
8	ERR_T	0	R/W	Non-recovered transient data integrity error. This bit is set if any of the 10b to 8b decoder error (DIAG_B), CRC error (DIAG_C), Disparity error (DIAG_D), Handshake error (DIAG_H), Transport transition error (DIAG_T), or RxFIFO overflow is detected in the Data FIS, or non-data FIS transmission was not

				recovered after 3 retries.
7-2	-	0		Reserved
1	ERR_M	0	R/W	Recovered communication error. This bit is set when PHY READY condition is detected after interface initialization, but not after transition from PARTIAL or SLUMBER power management state to active state.
0	ERR_I	0	R/W	Recovered data integrity error. This bit is set when non-data FIS transmission was unsuccessful (such as when a DIAG_H error is detected), but was recovered after 1 to 3 retries.

SControl Register (SCR2) 0xF056002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												PMP			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPM				IPM				SPD				DET			

This register provides control for the SATA interface. Write operations to the SControl register result in an action being taken by the host adapter or interface. Read operations from the register return the last value written to it.

Note: These bits are static and should not be changed frequently due to the clock-crossing from the Transport-to-Link Layers. Software must wait for at least seven periods of the slower clock (clk_asic or hclk) before changing this register.

Bit	Name	Initial	Type	Description
31-20	-	0		Reserved
19-16	PMP	0	R/W	The Port Multiplier Port (PMP) field represents the 4-bit value that is placed in the PM Port field of all transmitted FISes.
15-12	SPM	0	WO	The Select Power management field (SPM) is used to select a power management state. A non-zero value written to this field causes the power management state specified to be initiated. A value written to this field is treated as a one-shot. It is read as 4'0000. • 4'b0000: No power management state transition requested. • 4'b0001: Transition to PARTIAL power management state initiated. • 4'b0010: Transition to SLUMBER power management state initiated. • 4'b0100: Transition to the active power management state initiated. All other values reserved. Note: Normally system software should wait until the interface is in the active state using SStatus register (SCR0.DET = 4'b0011 or SCR0.IPM = 4'b0001) before initiating either PARTIAL or SLUMBER power management state, otherwise the corresponding signal (phy_partial or phy_slumber) is asserted without PMREQ/PMACK handshaking protocol. This feature can be used to put the PHY into power management state even if a device is not connected.
11-8	IPM	0	R/W	This field represents the enabled interface power management states that can be invoked with the SATA interface power management capabilities: • 4'b0000: No interface power management state restrictions • 4'b0001: Transitions to the PARTIAL state disabled • 4'b0010: Transitions to the SLUMBER state disabled • 4'b0011: Transitions to both the PARTIAL and SLUMBER states disabled All other values reserved.
7-4	SPD	0	R/W	This field represents the highest-allowed communication speed that the interface is allowed to negotiate when the speed is established: • 4'b0000: No speed negotiation restrictions • 4'b0001: Limit speed negotiation to a rate not greater than Generation 1 communication rate • 4'b0010: Limit speed negotiation to a rate not greater than Generation 2 communication rate All other values reserved. Note: If the host software needs to change this field's value, it should reset the interface (SCR2.DET = 4'b0001) at the same time to ensure proper speed negotiation.
3-0	DET	0	R/W	This field controls the host adapter device detection and interface initialization: • 4'b0000: No device detection or initialization action requested • 4'b0001: Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communication reinitialized (COMRESET condition). • 4'b0100: Disable SATA interface and put PHY in offline mode. All other values reserved.

SActive Register (SCR3) 0xF0560030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCR3[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SCR3[15:0]															

This register is used for native SATA command queuing and contains the information returned in the SActive field of the Set

Device Bits FIS.

The host software can set bits in the SActive register by a write operation to this register. The value written to the set bits should have ones encoded in the bit positions corresponding to the bits that are to be set. Bits in the SActive register can not be cleared as a result of a register write operation by the host, and host software cannot clear bits in the SActive register.

Set bits in the SActive register are cleared as a result of data returned by the device in the SActive field of the Set Device Bits FIS. The value returned in this field has ones encoded in the bit positions corresponding to the bits that are to be cleared. The device cannot set bits in this register.

All bits in the SActive register are cleared upon issuing a hard reset (COMRESET) signal or as a result of issuing a software reset by setting SRST bit of the Device Control register.

For the native command queuing protocol, the SActive value represents the set of outstanding queued commands that have not completed successfully yet. The value is bit-significant and each bit position represents the status of a pending queued command with a corresponding TAG value.

SNotification Register (SCR4) 0xF0560034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Notify															

This register is used to notify the host software which devices have sent a Set Device Bits FIS with Notification bit set. When a Set Device Bits FIS with Notification bit set to 1 is received, the bit corresponding to the value of the PM Port field in the FIS is set, and an interrupt is generated if the I bit in the FIS is set and interrupt is enabled (nIEN = 0 in the Device Control register).

Set bits in the SNotification register are explicitly cleared by a write operation to the SNotification register, or a power-on reset. The register is not cleared due to a COMRESET condition. The value written to clear set bits should have ones encoded in the bit positions corresponding to the bits that are to be cleared.

Bit	Name	Initial	Type	Description
31-16	-	0		Reserved
15-0	Notify	0	R/W	This field represents whether a particular device with the corresponding PM Port number has sent a Set Device Bits FIS to the host with the Notification bit set.

First-Party DMA Tag Register (FPTAGR) 0xF0560064

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											TAG				

This register is used to notify the host software which devices have sent a Set Device Bits FIS with Notification bit set. When a Set Device Bits FIS with Notification bit set to 1 is received, the bit corresponding to the value of the PM Port field in the FIS is set, and an interrupt is generated if the I bit in the FIS is set and interrupt is enabled (nIEN = 0 in the Device Control register).

Set bits in the SNotification register are explicitly cleared by a write operation to the SNotification register, or a power-on reset. The register is not cleared due to a COMRESET condition. The value written to clear set bits should have ones encoded in the bit positions corresponding to the bits that are to be cleared.

Bit	Name	Initial	Type	Description
31-5	-	0		Reserved
4-0	TAG	0	RO	First-party DMA TAG value. Updated every time a new DMA Setup FIS is received from the device.

First-Party DMA Buffer Register (FPBOR) 0xF0560068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FPBOR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPBOR[15:0]															

This register contains the DMA buffer offset value, which is updated every time a new DMA Setup FIS is received. The device uses the offset to transfer DMA data out of order. Bits 1 and 0 should always be cleared (32-bit-aligned offset). A write access to this location results in the bus error response.

First-Party DMA Transfer Count Register (FPTCR) 0xF056006C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FPTCR [31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FPTCR [15:0]															

This register contains the number of bytes that is transferred. It is updated every time a new DMA Setup FIS is received. Bit 0 should always be cleared (even number of bytes and the value should be non-zero). A write access to this location results in the bus error response.

DMA Control Register (DMACR) 0xF0560070

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
													TXMODE	RXCHEN	TXCHEN

This register contains bits indicating the status of the DMA transmit or receive channel. Power-on or COMRESET clears this register.

Note: Both TXCHEN and RXCHEN bits are cleared by the SATA Host after the First-party DMA command data phase ends (all the data for this command has been transferred).

Bit	Name	Initial	Type	Description
31-3	-	0		Reserved
2	TXMODE	0	R/W	Transmit Mode: • 1'b0: Tx Data FIS is closed when dma_finish_tx is asserted at the end of the DMA block transfer. • 1'b1: Tx Data FIS is closed when TXCHEN bit is cleared at the end of the DMA command transfer. Dma_finish_tx input is ignored in this case.
1	RXCHEN	0	R/W	Receive Channel Enable: • 1'b1: DMA receive channel is enabled and ready for transfer • 1'b0: DMA receive channel is disabled Software must set this bit prior to issuing a DMA read command to the device and clear this bit after DMA controller has finished transferring data for this command. Note: If this bit is cleared during data transfer and SError bits ERR_P, DIAG_C, DIAG_B, and DIAG_D are all cleared, the Link SYNC-escapes the current FIS. If any of these SError bits is set when RXCHEN is cleared, the current Data FIS is flushed from the RxFIFO until the TSM detects the End Status DWORD. This feature is useful for software error recovery operation.
0	TXCHEN	0	R/W	Transmit Channel Enable: • 1'b1: DMA transmit channel is enabled and ready for transfer • 1'b0: DMA transmit channel is disabled Software must set this bit prior to issuing a DMA write command to the device and clear this bit after DMA controller has finished transferring data for this command.

DMA Burst Transaction Size Register (DBTSR) 0xF0560074

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								MRD							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								MWR							

This register is used to set RxFIFO “pop almost empty” and TxFIFO “push almost full” thresholds to the burst transaction size (in DWORDs) for a DMA read or write operation. SATA Host generates corresponding request signals (dma_req_rx or dma_req_tx) to the DMA controller as follows:

- dma_req_rx is asserted when RxFIFO contains enough data for the burst transaction of MRD size
- dma_req_tx is asserted when TxFIFO contains enough free space for the burst transaction of MWR size.

Power-up or COMRESET condition initializes this register to the value shown above.

The MRD/ MWR field can only be written if the corresponding RXCHEN/ TXCHEN bit of the DMACR register is cleared.

Note: It is important that the DMA Read/Write burst transaction size never exceeds MRD/MWR values, otherwise an ERROR response is generated by the slave interface if either the RxFIFO-empty or the TxFIFO-full condition is detected during DMA bus transfer. Host software must ensure that the DMA controller is programmed with the same values prior to enabling a channel for transfer. 8-bit write accesses are ignored. MRD and MWR fields can be accessed separately using 16-bit wide transfers with naturally-aligned addresses:

- DBTSR for MWR field
- DBTSR+2 for MRD field.

Bit	Name	Initial	Type	Description
31-24	-	0		Reserved
23-16	MRD	0x14	R/W	This field is used to set the RxFIFO “pop almost empty” flag to the maximum burst size in DWORDs for the DMA read operation. Can be written if DMACR.RXCHEN = 0, otherwise, the write to this field is ignored. Note: MRD = 0 might result in a bus error response during DMA read transaction. Upper boundary is derived from the fact that the device stops sending data when the host generates HOLDp raf DWORDs from the RxFIFO “full” condition. This might result in possible lock condition (dma_req_rx is never generated) if this value is exceeded. Range: 1 to (256−raf-1), raf = 43 Defaults to 20 DWORDs on reset (64-43-1 = 20).
15-8	-	0		Reserved
7-0	MWR	0x10	R/W	This field is used to set the TxFIFO “push almost full” flag to the maximum burst size in DWORDs for the DMA write operation. Can be written if DMACR.TXCHEN = 0, otherwise, the write to this field is ignored. Note: MWR = 0 might result in bus error response during DMA write transaction. Defaults to 16 DWORDs on reset (32/2 = 16). Range: 1 to (256−2)

Interrupt Pending Register (INTPR) 0xF0560078

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPF					ERR_ADDR										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CMD_GOOD	CMD_ABORT	PRIMERR	NEWBIST	ERR	PMABORT	NEWFP	DMAT

This register contains all SATA host interrupt events before masking. The bits are set by an interrupt event. All the interrupt bits together with the IPF ATA interrupt flag are ORed to generate the SATA host intrq output. The set bits in the INTPR register can be cleared by a write operation to the register, or a reset operation (power-on or COMRESET). The value written to clear set bits should have ones encoded in the bit positions corresponding to the bits that are to be cleared. The reset value of each bit described in the following table is 0.

Bit	Name	Initial	Type	Description
31	IPF	0	RO	This bit reflects the state of the ATA Interrupt Pending Flag (IPF). It is set when an

				FIS with I = 1 is received from the Device, and cleared when Status register CDR7 is read.
30-27	-	0		Reserved
26-16	ERR_ ADDR	0	RO	This field contains lower 11 bits of haddr bus (bits [10:0]) of an illegal bus access detected by the AHB slave interface (see “AHB Error Conditions”), or 0x3FF value if dma_finish_tx is asserted, but no data had been written to the DMADR location during DMA write operation. These bits can not be cleared by a write operation.
15-8	-	0		Reserved
7	CMD_ GOOD	0	R/W	This bit is set when a good Register or Set Device Bits FIS with Status.ERR bit cleared (bit 16 of the first DWORD) is received from the Device.
6	CMD_ ABORT	0	R/W	This bit is set when a good Register or Set Device Bits FIS with Status.ERR bit set (bit 16 of the first DWORD) is received from the Device.
5	PRIMERR	0	R/W	This bit is set when the Link Layer detects invalid K-character (not K28.5 or K28.3) in a primitive.
4	NEWBIST	0	R/W	This bit is set when a supported BIST Activate FIS is received from the device without errors.
3	ERR	0	RO	This bit is set when any of the bits in the SError register is set and the corresponding bit in the ERRMR register is set. This bit can not be cleared by a write operation since it reflects the state of the SError register. It is cleared if either SError register bits are cleared or masked with the ERRMR register.
2	PMABORT	0	R/W	This bit is set when the Link Layer detects a power mode abort condition (power mode is aborted by the device requesting a frame transmission). Note: this bit must be cleared explicitly by software before issuing a power management request to the interface.
1	NEWFP	0	R/W	This bit is set when a DMA Setup FIS is received from the device without errors.
0	DMAT	0	R/W	This bit is set when DMATp is received from the device during Data FIS transmission.

Interrupt Mask Register (INTMR) 0xF056007C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CMD_GOODM	CMD_ABORTM	PRIMERRM	NEWBISTM	ERRM	PMABORTM	NEWFPM	DMATM

This register is used to mask or enable corresponding interrupt events in the INTPR register. An interrupt event is masked if the bit is cleared and is enabled if set. If any of the INTPR bits are set or if IPF is set and enabled (unmasked), the intrq output is asserted. A COMRESET condition clears this register. The reset value of each bit described in the following table is 0.

Bit	Name	Initial	Type	Description
31-8	-	0		Reserved
7	CMD_GOODM	0	R/W	• 1'b1: CMD_GOOD interrupt is enabled • 1'b0: CMD_GOOD interrupt is masked
6	CMD_ABORTM	0	R/W	• 1'b1: CMD_ABORT interrupt is enabled • 1'b0: CMD_ABORT interrupt is masked
5	PRIMERRM	0	R/W	• 1'b1: PRIMERR interrupt is enabled • 1'b0: PRIMERR interrupt is masked
4	NEWBISTM	0	R/W	• 1'b1: NEWBIST interrupt is enabled • 1'b0: NEWBIST interrupt is masked
3	ERRM	0	RO	• 1'b1: ERR interrupt is enabled • 1'b0: ERR interrupt is masked
2	PM_ABORTM	0	R/W	• 1'b1: PMABORT interrupt is enabled • 1'b0: PMABORT interrupt is masked
1	NEWFPM	0	R/W	• 1'b1: NEWFP interrupt is enabled • 1'b0: NEWFP interrupt is masked
0	DMATM	0	R/W	• 1'b1: DMAT interrupt is enabled • 1'b0: DMAT interrupt is masked

Error Mask Register (ERRMR) 0xF0560080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERRMR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRMR[15:0]															

This register is used to mask or enable corresponding bits of the SError register prior to setting the ERR bit of the INTPR. This allows driver software to select the SError bits that can cause the interrupt output intrq to be asserted. The INTPR ERR bit is set if any of the SError bits are set and the corresponding ERRMR bit is set. Clearing the ERRMR bit would mask the corresponding SError bit from setting the INTRP ERR bit. COMRESET condition clears this register.

Link Layer Control Register (LLCR) 0xF0560084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERRMR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRMR[15:0]															

This register provides Link Layer (LL) control capability for the host software. Power-on or COMRESET condition sets SCRAM, DESCRAM, and RPD bits. The SCRAM bit is cleared when the SATA Host enters a BIST device far-end transmit mode with scrambling bypassed.

Note: These bits are static and should not be changed frequently due to the clock-crossing from Transport to Link Layers. Software must wait at least seven periods of the slower clock (clk_asic or hclk) before changing this register.

Bit	Name	Initial	Type	Description
31-5	-	0		Reserved
4	ERREN	0	R/W	Error Enable RW This bit allows or filters (disables) PHY Internal errors outside the FIS boundary to set corresponding SError bits. • 0: Filter errors outside the FIS, allow errors inside the FIS • 1: Allow errors outside or inside the FIS.
3	-	0		Reserved
2	RPD	1	R/W	• 1'b1: Repeat primitive drop function enabled • 1'b0: Repeat primitive drop function disabled
1	DESCRAM	1	R/W	• 1'b1: Descrambler enabled • 1'b0: Descrambler disabled
0	SCRAM	1	R/W	• 1'b1: Scrambler enabled • 1'b0: Scrambler disabled

Received BIST Pattern Definition Register (RXBISTPD) 0xF0560090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RXPD[7:0]							

This register contains the pattern definition field of the received BIST Activate FIS (bits [23:16] of the first DWORD). This field defines the SATA host loopback mode requested by the device. It is updated every time a new BIST Activate FIS is received from the device. A write access to this location results in a bus error response.

Bit	Name	Initial	Type	Description
31-8	-	0		Reserved
7-0	RXPD	0	RO	Pattern definition field of the received BIST Activate FIS (bits [23:16] of the first DWORD). It is used to put the SATA host in one of the following BIST modes: • 0x10: Far-end retimed • 0x08: Far-end analog (if PHY supports this mode) • 0x80: Far-end transmit only • 0xA0: Far-end transmit only with scrambler bypassed All other values should not be used by the device, otherwise the FIS is negatively acknowledged with R_ERRp. For far-end transmit only modes RXBISTD1 register would contain the required data pattern.

Received BIST Data DWORD 1 (RXBISTD1) 0xF0560094

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXBISTD1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXBISTD1[15:0]															

This register contains the second DWORD (first data DWORD) of the received BIST Activate FIS. It is updated every time a new three-DWORD BIST Activate FIS is received from the device and selects a pattern for far-end transmit only loopback mode. A write access to this location results in a bus error response.

Bit	Name	Initial	Type	Description
31-0	RXBISTD1	0	RO	The following patterns are supported : ❖ EE27FEF1h: Low transition density pattern; ❖ B5B5B5B5h: High transition density pattern; ❖ ABABABABh: Low frequency spectral component pattern; ❖ 7F7F7F7Fh: Simultaneous switching outputs pattern.

Received BIST Data DWORD 2 (RXBISTD2) 0xF0560098

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXBISTD2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXBISTD2[15:0]															

This register contains the third DWORD (second data DWORD) of the received BIST Activate FIS. It is updated every time a new three-DWORD BIST Activate FIS is received from the device. A write access to this location results in a bus error response.

If none of these values is detected in the RXBISTD1 register, simultaneous switching pattern is transmitted by default.

Bit	Name	Initial	Type	Description
31-0	RXBISTD2	0	RO	

Transmit BIST Pattern Definition Register (TXBISTPD) 0xF056009C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TXPD[7:0]							

This register contains the pattern definition field in bits [7:0] of the transmit BIST Activate FIS. The host software puts the

device in one of the BIST modes by write access to this register.

Note Host software should reset the interface by toggling SControl[0] and wait for the BSY and DRQ bits to be cleared prior to requesting any of the BIST modes. The device must support the BIST mode requested by the host software, otherwise the result is indeterminate.

Bit	Name	Initial	Type	Description
31-8	-	0		Reserved
7-0	TXPD	0	R/W	Pattern definition field of the transmit BIST Activate FIS (bits [23:16] of the first DWORD). It is used to put the device in one of the loopback modes as defined in SATA specification: • Bit 0 (V): Vendor-specific • Bit 1 (R): Reserved • Bit 2 (P): Primitive bit (valid only with the T-bit) • Bit 3 (F): Far-end analog loopback • Bit 4 (L): Far-end retimed loopback • Bit 5 (S): Bypass scrambling (valid only with T-bit) • Bit 6 (A): Bypass align (valid only with T-bit) • Bit 7 (T): Far-end transmit only Each write access causes BIST Activate FIS to be transmitted to the device. Bits 3 (F), 4 (L), and 7 (F) are mutually exclusive and should not be set at the same time, otherwise the BIST request is ignored. For far-end transmit only mode (T = 1) the software has to load TXBISTD1 and TXBISTD2 registers with the required pattern first, then write to the TXBISTPD.

Transmit BIST Data DWORD 1 (TXBISTD1) 0xF05600A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXBISTD1[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXBISTD1[15:0]															

This register contains the first data DWORD with the BIST pattern. It is updated by the host software prior to requesting a new three-DWORD BIST Activate FIS transmission to the device (by writing to the TXBISTPD with bit 7 set).

Bit	Name	Initial	Type	Description
31-0	TXBISTD1	0	R/W	

Transmit BIST Data DWORD 2 (TXBISTD2) 0xF05600A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXBISTD2[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXBISTD2[15:0]															

This register contains the second data DWORD with the BIST pattern. It is updated by the host software prior to requesting a new three-DWORD BIST Activate FIS transmission to the device (by writing to the TXBISTPD with bit 7 set).

Bit	Name	Initial	Type	Description
31-0	TXBISTD2	0	R/W	

BIST Control Register (BISTCR) 0xF05600A8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														CNTCLR	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
													PATTERN[2:0]		

This register is used in BIST initiator modes. This register is loaded by the host software prior to sending BIST Activate FIS to the device (via TXBISTPD write).

Bit	Name	Initial	Type	Description
31-18	-	0		Reserved
17	CNTCLR	0	WO	This bit is used to clear BIST error count registers. This bit is treated as one-shot and read as zero:

				1: Clear registers: BISTFCTR, BISTSR, BISTDECR.
16	-	0	WO	Reserved
15-3	-	0		Reserved
2-0	PATTERN	0	R/W	This field defines one of the following SATA compliant patterns to be transmitted by SATA host to the device in far-end retimed/ far-end analog modes, or to the host PHY in near-end analog mode: • 3'b000: Simultaneous switching outputs bit pattern • 3'b001: High transition density bit pattern • 3'b010: Low transition bit pattern • 3'b011: Low frequency spectral component bit pattern • 3'b100: Composite pattern All other values are reserved and should not be used.

BIST FIS Count Register (BISTFCTR) 0xF05600AC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BISTFCTR [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BISTFCTR[15:0]															

This register contains the received BIST FIS count in the loopback initiator far-end retimed, far-end analog and near-end analog modes. It is updated each time a new BIST FIS is received. It is cleared by COMRESET or by software setting the BISTCR register bit CNTCLR. This register does not roll over and freezes when the FFFF_FFFFh value is reached. It takes approximately 66 hours of continuous BIST operation at Gen. 1 speed to reach this value. Write access to this location results in a bus error response.

Bit	Name	Initial	Type	Description
31-0	BISTFCTR	0	RO	

BIST Status Register (BISTSR) 0xF05600B0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								BRSTERR [23:16]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRAMERR[15:0]															

This register contains errors detected in the received BIST FIS in the loopback initiator far-end retimed, far-end analog and near-end analog modes. It is updated each time a new BIST FIS is received. It is cleared by a COMRESET condition or by the software setting the BISTCR register bit CNTCLR.

Bit	Name	Initial	Type	Description
31-0	-	0		Reserved
23-16	BRSTERR	0	RO	This field contains the burst error count. It is accumulated each time a burst error condition is detected: DWORD error is detected in the received frame and 1.5 seconds (27,000 frames) passed since the previous burst error was detected. The BRSTERR value does not roll over and freezes at FFh. This field is updated if parameter BIST_MODE=DWORD.
15-0	FRAMERR	0	RO	This field contains the frame error count. It is accumulated (new value is added to the old value) each time a new BIST frame with a CRC error is received. The FRAMERR value does not roll over and freezes at FFFFh.

BIST DWORD Error Count Register (BISTDECR) 0xF05600B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DWERR [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWERR[15:0]															

This register contains the number of DWORD errors detected in the received BIST frame in the loopback initiator far-end retimed, far-end analog and near-end analog modes. It is updated each time a new BIST frame is received. It is cleared by a COMRESET condition or by the software setting the BISTCR register bit CNTCLR. A write access to this location results in bus error response. This register is updated if parameter BIST_MODE=DWORD.

Bit	Name	Initial	Type	Description
31-0	DWERR	0	RO	This field contains the DWORD error count. It is accumulated (new value is added to the old value) each time a new BIST frame is received. The DWERR

				value does not roll over and freezes if it exceeds 0xFFFF_F000.

Test Register (TESTR) 0xF05600F4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															BSYCLR
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
														TIEN	TESTIF

This register is used to put sata host slave interface into a test mode; a condition where the readback value of all the registers match the value written. In normal operation the readback value of some registers is a function of sata host state and does not match the value written. This register is not reset by the COMRESET condition.

The registers are treated as follows when TESTIF = 1:

− CDR0 becomes 32-bit Read/Write register. Only single transfers are allowed.

- CDR1 becomes Read/Write register similar to CDR2–CDR5. CLR0 HOB bit state determines which register is read when accessing CDR1–CDR5: HOB = 1: Feature expanded register, HOB = 0: Feature register and so on. 16-bit read access to CDR1–CDR5 returns both registers as follows (using CDR1 as example):

* HOB = 1: [15:8] = Feature register, [7:0] = Feature expanded register

* HOB = 0: [15:8] = Feature expanded register, [7:0] = Feature register

- CDR7 and CLR0 become Read/Write registers.

- All read-only registers (for example, SStatus, FPTAGR) except VERSIONR, IDR and the ERR_ADDR field of the INTPR become Read/Write registers. Writes to VERSIONR, IDR and the ERR_ADDR field of INTPR registers are ignored.

- DMADR locations become illegal and return ERROR response if accessed (only when RETURN_ERR_RESP = 1).

Note : If the test interface mode is entered with the device connected to the host adapter, interface should be reset by setting SControl[0] = 1 (COMRESET) after TEST_IF bit is cleared.

Bit	Name	Initial	Type	Description
16	BSYCLR	0	WO	This bit is used to clear Status BSY and DRQ bits by software when set. This bit is treated as one-shot type and read as zero. NOTE: This bit should not be used for normal operation, it is intended only for error recovery or debug.
1	TIEN	0	R/W	Test Interrupt Enable: • 1'b0: Intrq output is not asserted in test mode when TESTIF = 1. • 1'b1: Intrq output is asserted in test mode when TEST_IF = 1 and any of the INTPR bits and corresponding INTMR bits are set.
0	TESTIF	0	R/W	• 1'b0: Normal mode. • 1'b1: Test mode: the readback value of all the registers matches the value written. Normal operation is disabled.

Version Register (VERSIONR) 0xF05600F8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VERSION [31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VERSION [15:0]															

This 32-bit read-only register contains hard-coded hexadecimal version. The value represents an ASCII code of the version number. A write access to this register results in the bus error response.

Bit	Name	Initial	Type	Description
31-0	VERSION	0x3139_302A	RO	

ID Register (IDR) 0xF05600FC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID [31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

ID [15:0]

This register contains hard-coded hexadecimal identification value. A write access to this register results in the bus error response.

Bit	Name	Initial	Type	Description
31-0	VERSION	0	RO	

FIFO location in DMA mode (DMADR) **0xF0560400-0xF05607FC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERRMR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRMR[15:0]															

This location is used to transfer data from host to device and from device to host in DMA mode..

Bit	Name	Initial	Type	Description
31-0	DMADR		RO/WO	Dependencies: This location can only be accessed by the host software or DMA controller when the SATA Host is in the corresponding DMA mode: • read (when the Data FIS is being received) • write (when the DMA Activate or DMA Setup FIS is received), and the corresponding DMA handshake signal is asserted (dma_req or dma_single). Both single and burst 32-bit or 16-bit bus transfers are supported in this mode.

Note: The same transfer size (either 16 or 32 bits) should be maintained during the whole DMA transfer. Starting address of the burst transfer should always be 0x400, except when the burst is interrupted by the system arbiter (early burst termination) and continued with the next address within the 0x400 to 0x7FC range. DMA address can be fixed to 0x400 for all the beats in the burst. The burst addresses must be 32-bit-aligned for 32-bit.

10.5.2 SATA PHY Registers

We are using I2C mode control scheme for PHY characterization and debugging.

I2C Write Sequence

Write I2C_ADDR(0x70) -> Write Start_Reg Number -> Write DATA0 -> Write DATA1 ….

I2C Read Sequence

Write I2C_ADDR(0x71) -> Read Reg0 DATA -> Read Reg1 DATA ….

Following Table shows I2C register setting examples.

Table 10.2 I2C Register Setting Examples

Mode	Address	R/W	MESSAGE DATA BYTE (Hex)
Default Values	70	R/W	01 95 25 3F 04 00 30 1B 50 01 00 00 04 63 03 00
Data Mode	70	W	0E 03 80
BIST Mode	70	W	0E E3 80

SATA_PHY_I2C_REG3 **0x3**

7	6	5	4	3	2	1	0
		SSC[5:0]					

Bit	Name	Initial	Type	Description
5-0	SSC[5:0]	0x3F	R/W	SSC amount = 1 / (5*SSC[6:0]) * 106 ppm,(Default, 3170ppm) If you'd like to change the SSC(Spread Spectrum Clocking) modulation amount, you could set these 7-bit registers. For normal case, we recommend the use of the default setting. In the SATA specification, the modulation amount should be 0~5000ppm range.

SATA_PHY_I2C_REG4 **0x4**

7	6	5	4	3	2	1	0
SSC[6]	PE			SWING			

Bit	Name	Initial	Type	Description
7	SSC[6]	0	R/W	SSC amount [6]
6-4	PE	0	R/W	TX Pre-emphasis Level Control Use default setting. If there are long cable(or media), generally high pre-emphasis level setting would be better. But there would be optimization setting points according to the cable length. In addition TX swing level would be changed according to the TX pre-emaphasis level control setting.
3-0	SWING	0x4	R/W	TX Differential Output (TXP/TXN) Amplitude Control 4'b1111: Maximum Swing (~700mVp-p) 4'b0100: Default Swing (~500mVp-p) 4'b0000: Minimum Swing (~400mVp-p) TX voltage swing unit step is different. If you chage this setting, TX swing specification may be changed because we should consider all PVT variation. If you reduce TX swing, TX IO power consumption would be also reduced propotionally. If there are long cable, you'd better large swing setting because of the loss in the channel.

SATA_PHY_I2C_REG5 **0x5**

7	6	5	4	3	2	1	0
	SR		MAN_CAL		TZCODE		

Bit	Name	Initial	Type	Description
6-5	SR	0	R/W	TX Slew-rate Control 2'b00:~100ps 2'b11:~140ps Control the slew-rate according to data rate.
4	MAN_CAL	0	R/W	Manual Impedance Control Enable 0: Auto Calibration (Default Value) 1: Manual Impedance Control Enable When this pin is enabled, TX and RX Impedance can be controlled by manual code.
3	-			Reserved
2-0	TZCODE	0	R/W	Manual TX Impedance Control 3'b000: Small 3'b111: Large When Manual Impedance Control Enable pin is 'high', this code works.

SATA_PHY_I2C_REG7 **0x7**

7	6	5	4	3	2	1	0
RX_INV	EQ				SQTH		

Bit	Name	Initial	Type	Description
7	RX_INV	0	R/W	RX INV Control 0: Normal Mode (Default Value) 1: RXP/RXN Output Data Polarity Inversion This would be used for RXP/RXN polarity inversion.
6-4	EQ	0x1	R/W	RX Equalizer Level Control 3'b000: Normal Mode = Equalizer Off 3'b001: Equalizer 1st stage On (Default Value) 3'b011: Equalizer 1st & 2nd stage On 3'b111: Equalizer 1st, 2nd & 3rd stage On (Maximum) Optimum setting is dependent on the properties of the channel. Generally we recommend the use of this setting change only if there are large ISI(inter-symbol-interference) in the channel.
3	-	1		Reserved but do not change into 0.
2-0	SQTH	0x3	R/W	RX Squelch Detect Threshold Control Use default setting.

SATA_PHY_I2C_REG8 **0x8**

7	6	5	4	3	2	1	0
					RZCODE		

Bit	Name	Initial	Type	Description
2-0	RZCODE	0	R/W	Manual RX Impedance Control 3'b000: Small 3'b111: Large When Manual Impedance Control Enable pin is 'high', this code works.

SATA_PHY_I2C_REGA **0xA**

7	6	5	4	3	2	1	0
						SSCEN	

Bit	Name	Initial	Type	Description
1-0	SSCEN	0	R/W	TX SSC Enable 2'b00: Normal Mode (Default Value) 2'b11: TX SSC(Spread Spectrum Clock) Enable This would be used for TX SSC enable.

SATA_PHY_I2C_REGD **0xD**

7	6	5	4	3	2	1	0
RXC				TX_INV			

Bit	Name	Initial	Type	Description
7-6	RXC	0x1	R/W	
5-4	-			Reserved
3	TX_INV	0	R/W	TX INV Control 0: Normal Mode (Default Value) 1: TXP/TXN Output Data Polarity Inversion This would be used for TXP/TXN polarity inversion.
2-0	-			Reserved

SATA_PHY_I2C_REGE **0xE**

7	6	5	4	3	2	1	0
RBSTEN	TBSTEN		RATE_PD EN	SLUMBER	PARTIAL	RX_RATE	TX_RATE

Bit	Name	Initial	Type	Description
7	RBSTEN	0	R/W	RX BIST EN 0: Normal mode(Default) 1: RX BIST mode Enabled This would be used for BIST mode. You can control RX BIST mode by using this bit.
6-5	TBSTEN	0	R/W	TX BIST EN 0: Normal mode(Default) 1: TX BIST mode Enabled This would be used for BIST mode. One bit will be used for the analog part of SATA PHY control and the other bit will be used for the digital part of that. So you need to change both of the bits to activate TX BIST mode.
4	RATE_PD EN	0	R/W	
3	SLUMBER	0	R/W	
2	PARTIAL	0	R/W	
1	RX_RATE	1	R/W	
0	TX_RATE	1	R/W	

SATA_PHY_I2C_REGF **0xF**

7	6	5	4	3	2	1	0
TESTEN						LBMODE	LBEN

Bit	Name	Initial	Type	Description
7	TESTEN	0	R/W	Test mode Enable 0: Normal model (Default) 1: PHY Test Mode Enabled In the SATA PHY test mode (TESTEN='1'), TESTIN[0:11] will be assigned as TXD[0:9], TBC and TX_EN. And TESTOUT[0:12] will be assigned as RXD[0:9], COMMA, SIGVALID and RBC.
6-2	-			Reserved
1	LBMODE	0	R/W	Loopback MODE 0: TX-to-RX loopback 1: RX-to-TX loopback If you'd like to set RX-to-TX parallel loopback mode, you should set LBEN ='1' & LBMODE='1'. In this mode, parallel RX data will be loopbacked to parallel TX data. So you can check its functionality easily with serial BERT test equipment.
0	LBEN	0	R/W	Loopback Enable 0: Normal mode(Default) 1: Internal Loopback Enabled This LBEN is used with LBMODE. If you'd like to TX-to-RX internal looback (i.e. EWRAP mode), you should set LBEN ='1' & LBMODE='0'.

11 AUDIO DMA Controller

11.1 Overview

11.1.1 AUDIO DMA

AUDIO DMA is a specialized DMA for Audio Interfaces such as DAI, CDIF and SPDIF. For earlier TCC series, only general DMA is available. When a general DMA is used for transferring audio data, it has to read from memory and write to Audio Interface registers. It seems to have redundant bus cycle time from the viewpoint of time-critical applications. So we applied a dedicated DMA to Audio Interface to reduce redundant bus cycle time. AUDIO DMA enable you to utilize audio data for more flexible purpose. The structure of AUDIO DMA is similar to that of general DMA. You also use general DMA instead of AUDIO DMA. You are not recommended to use AUDIO DMA and general DMA at the same time.

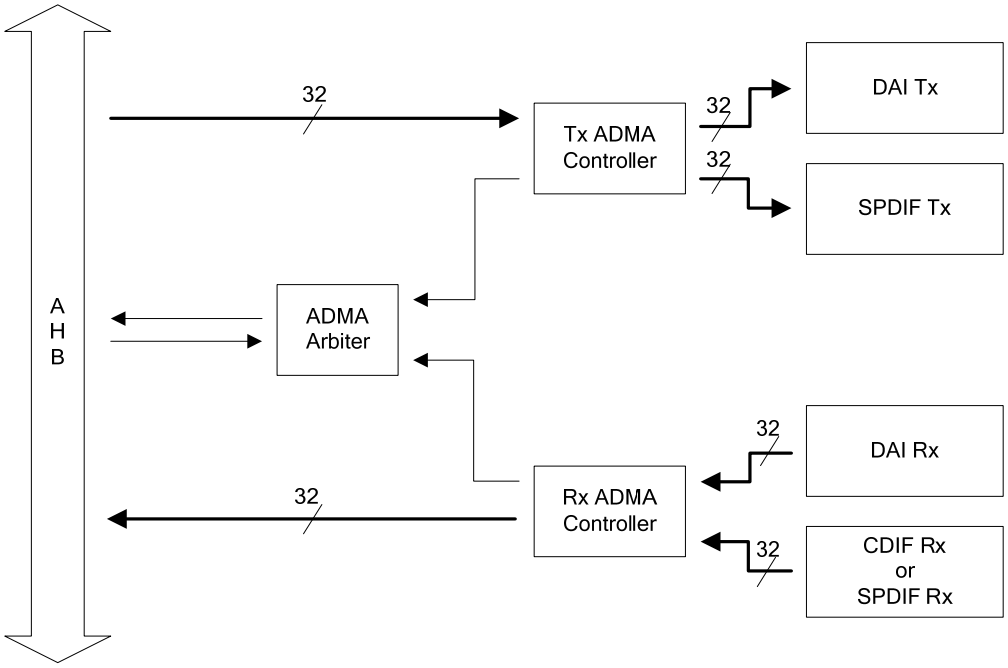


Figure 11.1 AUDIO DMA Top Block Diagram

11.1.2 DAI & CDIF

The block diagram of DAI is shown in below figure.

The TCC8900 provides digital audio interface that complies with IIS (Inter-IC Sound). The DAI has five input/output pins for IIS interface; MCLK, BCLK, LRCK, DAI, DAO. All DAI input/output pins are multiplexed with GPIO pins; GPIO_E[4:0].

The MCLK is the system clock pin that is used for CODEC system clock. In master mode, the MCLK can be generated from clock generator in which that is known as a DCLK, or fed from the outside of chip in slave mode. The DAI can process 256fs, 384fs and 512fs as a system clock. 256fs means that the system clock has 256 times of sampling frequency (fs).

The BCLK is the serial bit clock for IIS data exchange. The DAI can generate 64fs, 48fs and 32fs by dividing a system clock. The polarity of BCLK can be programmed. That is, the serial bit can be stable either rising edge of BCLK or falling edge of BCLK.

The LRCK is the frame clock for the stereo audio channel Left and Right. The frequency of LRCK is known as the “fs” – sampling frequency. Generally, for audio application – such as MP3 Player, CD player, the fs can be set to 8kHz, 16kHz, 11.05kHz, 24kHz, 32kHz, 44.1kHz and 48kHz. For supporting the wide range of sampling frequency in audio application, the DCO function is very useful to generate a system clock. Refer the chapter of clock generator for detail information.

All three clocks (MCLK, BCLK, LRCK) are selectable as master or slave.

The DAI, DAO are the serial data input output pins respectively.

The DAI has two 8-word input/output buffers. It has a banked buffer structure so that one side of buffer is receiving/transmitting data while the other side of that can be read/written through the DADI_XX/DADO_XX registers. The maximum data word size is 24 bit. Data is justified to MSB of 32bits and zeros are padded to LSB.

There are 2 types of interrupt from IIS; transmit done interrupt, receive done interrupt. The transmit-done interrupt is generated when the 8 words are transferred successfully in the output buffer. At this interrupt, user should fill another 8 more words into the other part of the output buffer in the interrupt service routine (ISR). In this ISR routine, 8 consecutive stores of word data to the DADO registers are needed. The receive-done interrupt is generated when the 8 words are received successfully in the input buffer. At this interrupt, user should read 8 received words from the input buffer using 8 consecutive load instructions from the DADI registers.

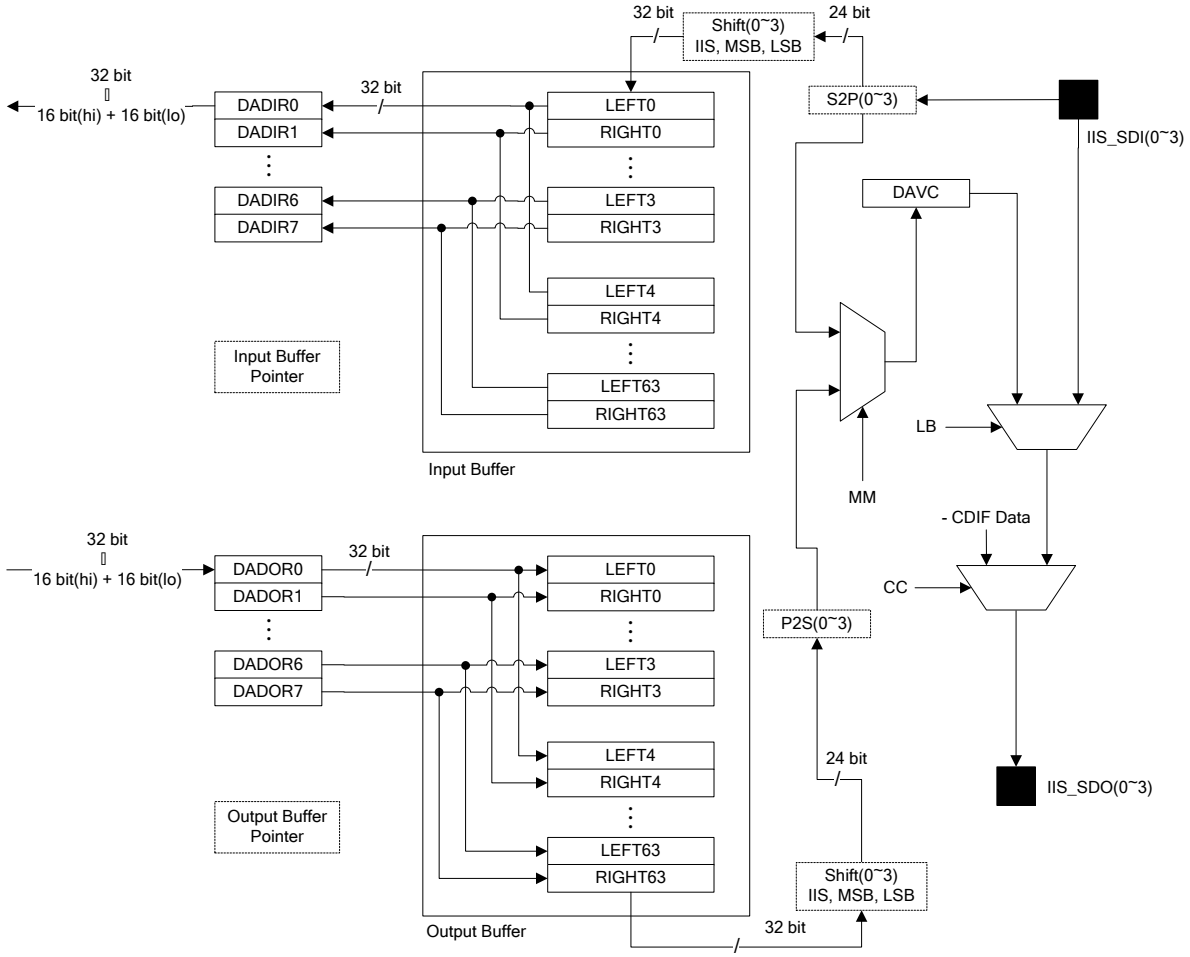


Figure 11.2 DAI Block Diagram

The block diagram of CDIF is illustrated in below figure.

The TCC8900 provides CD-ROM interface for feasible implementation of CD-ROM application such as CD-MP3 player. The CDIF supports the industry standard IIS format and the LSB justified format that is used as the most popular format for CD-ROM interface by Sony and Samsung.

The CDIF has three pins for interface; CBCLK, CLRCK, CDAI. These are multiplexed with GPIO_B[13], GPIO_B[14] and GPIO_B[12]. The CBCLK is the bit clock input pins of which frequency can be programmed by CICR for selection of 48fs and 32fs. The CLRCK is the frame clock input pin that indicates the channel of CD stereo digital audio data. The CDAI is the input data pin.

The CDIF has nine registers; CDDI_0 to CDDI_3 and CICR. The CDDI_0 to the CDDI_3 are the banked read only registers for access of data input buffer. The data input buffer is composed of sixteen 32 bit wide registers of which upper 16 bit is left channel data and lower is right channel data.

The CDIF receive the serial data from CDAI pin and store the data into the buffer through the serial to parallel register. Whenever the half of buffer is filled, the receive interrupt is generated. Only the half of input buffer can be accessible through the CDDI_0 to the CDDI_3.

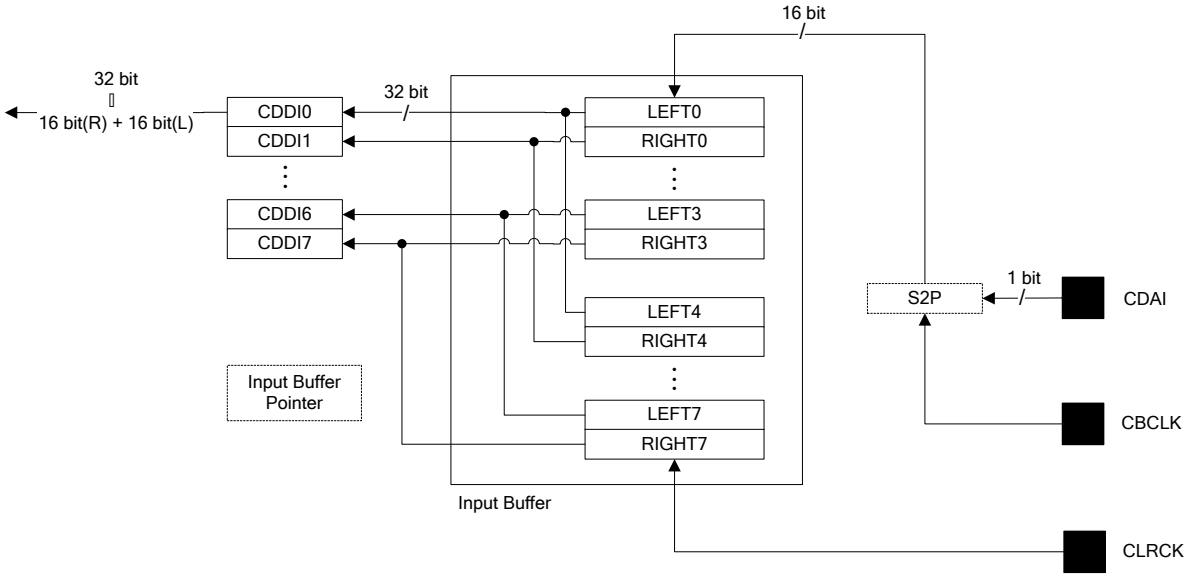


Figure 11.3 CDIF Block Diagram

11.1.3 SPDIF

The SPDIF (or AES/EBU, IEC950 standards) is a point-to-point protocol for serial transmission of digital audio through a single transmission line. The transmission medium can be either electrical or optical (e.g. TosLink). It provides two channels for audio data, a method for communicating control information, and some error detection capabilities. The control information is transmitted as one bit per sample and accumulates in a block structure. The data is by-phase encoded, which enables the receiver to extract a clock from the data. Coding violations, defined as preambles, are used to identify sample and block boundaries.

The SPDIF interfaces are found on most CD/DVD players, audio equipment and computer sound cards.

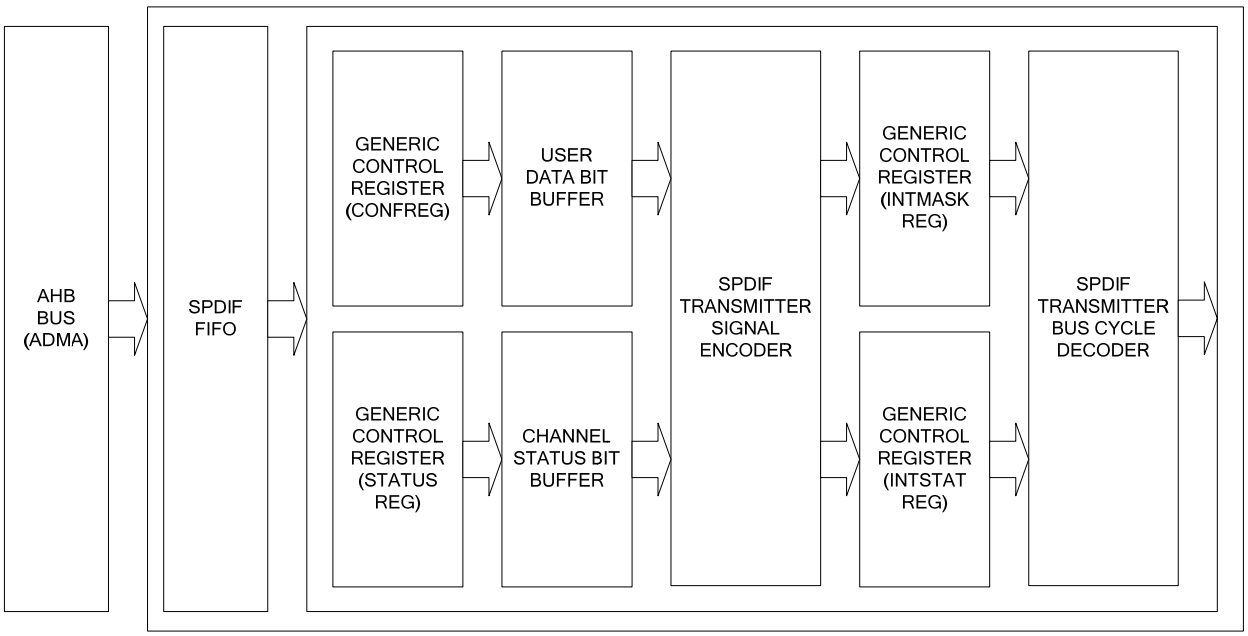


Figure 11.4 SPDIF Tx Block Diagram

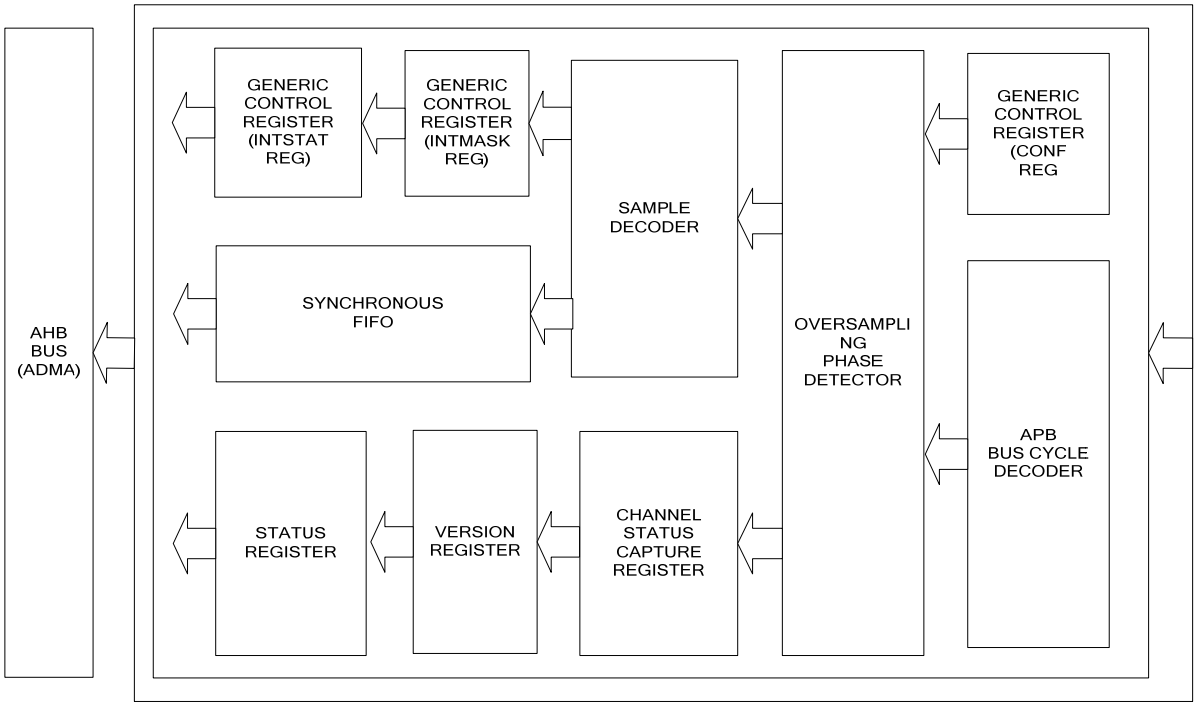


Figure 11.5 SPDIF Rx Block Diagram

11.2 Register Description

Table 11.1 AUDIO DMA Register Map (Base Address = 0xF0533000)

Name	Address	Type	Reset	Description
RxDaDar	0x00	R/W	0x00000000	DAI Rx (Right) Data Destination Address
RxDaParam	0x04	R/W	0x00000000	DAI Rx Parameters
RxDaTCnt	0x08	R/W	0x00000000	DAI Rx Transmission Counter Register
RxDaCdar	0x0C	R	0x00000000	DAI Rx (Right) Data Current Destination Address
RxCdDar	0x10	R/W	0x00000000	CDIF(SPDIF) Rx (Right) Data Destination Address
RxCdParam	0x14	R/W	0x00000000	CDIF(SPDIF) Rx Parameters
RxCdTCnt	0x18	R/W	0x00000000	CDIF(SPDIF) Rx Transmission Counter Register
RxCdCdar	0x1C	R	0x00000000	CDIF(SPDIF) Rx (Right) Data Current Destination Address
RxDaDarL	0x28	R/W	0x00000000	DAI Rx Left Data Destination Address
RxDaCdarL	0x2C	R	0x00000000	DAI Rx Left Data Current Destination Address
RxCdDarL	0x30	R/W	0x00000000	CDIF(SPDIF) Rx Left Data Destination Address
RxCdCdarL	0x34	R	0x00000000	CDIF(SPDIF) Rx Left Data Current Destination Address
TransCtrl	0x38	R/W	0x0000AA00	DMA Transfer Control Register
RptCtrl	0x3C	R/W	0x00000000	DMA Repeat Control Register
TxDaSar	0x40	R/W	0x00000000	DAI Tx (Right) Data Source Address
TxDaParam	0x44	R/W	0x00000000	DAI Tx Parameters
TxDaTCnt	0x48	R/W	0x00000000	DAI Tx Transmission Counter Register
TxDaCsar	0x4C	R	0x00000000	DAI Tx (Right) Data Current Source Address
TxSpSar	0x50	R/W	0x00000000	SPDIF Tx (Right) Data Source Address
TxSpParam	0x54	R/W	0x00000000	SPDIF Tx Parameters
TxSpTCnt	0x58	R/W	0x00000000	SPDIF Tx Transmission Counter Register
TxSpCsar	0x5C	R	0x00000000	SPDIF Tx (Right) Data Current Source Address
TxDaSarL	0x68	R/W	0x00000000	DAI Tx Left Data Source Address
TxDaCsarL	0x6C	R	0x00000000	DAI Tx Left Data Current Source Address
TxSpSarL	0x70	R/W	0x00000000	SPDIF Tx Left Data Source Address
TxSpCsarL	0x74	R	0x00000000	SPDIF Tx Left Data Current Source address
ChCtrl	0x78	R/W	0x00008000	DMA Channel Control Register
IntStatus	0x7C	R/W	0x00000000	DMA Interrupt Status Register
GIntReq	0x80	R/W	0x00000000	General Interrupt Request
GIntStatus	0x84	R	0x00000000	General Interrupt Status
RxDaDar1	0x100	R/W	0x00000000	DAI1 Rx (Right) Data Destination Address
RxDaDar2	0x104	R/W	0x00000000	DAI2 Rx (Right) Data Destination Address
RxDaDar3	0x108	R/W	0x00000000	DAI3 Rx (Right) Data Destination Address
RxDaCar1	0x10C	R	0x00000000	DAI1 Rx (Right) Data Current Destination Address
RxDaCar2	0x110	R	0x00000000	DAI2 Rx (Right) Data Current Destination Address
RxDaCar3	0x114	R	0x00000000	DAI3 Rx (Right) Data Current Destination Address
RxDaDarL1	0x118	R/W	0x00000000	DAI1 Rx Left Data Destination Address
RxDaDarL2	0x11C	R/W	0x00000000	DAI2 Rx Left Data Destination Address
RxDaDarL3	0x120	R/W	0x00000000	DAI3 Rx Left Data Destination Address
RxDaCarL1	0x124	R	0x00000000	DAI1 Rx Left Data Current Destination Address
RxDaCarL2	0x128	R	0x00000000	DAI2 Rx Left Data Current Destination Address
RxDaCarL3	0x12C	R	0x00000000	DAI3 Rx Left Data Current Destination Address
TxDaSar1	0x130	R/W	0x00000000	DAI1 Tx (Right) Data Source Address
TxDaSar2	0x134	R/W	0x00000000	DAI2 Tx (Right) Data Source Address
TxDaSar3	0x138	R/W	0x00000000	DAI3 Tx (Right) Data Source Address
TxDaCsar1	0x13C	R	0x00000000	DAI1 Tx (Right) Data Current Source Address
TxDaCsar2	0x140	R	0x00000000	DAI2 Tx (Right) Data Current Source Address
TxDaCsar3	0x144	R	0x00000000	DAI3 Tx (Right) Data Current Source Address
TxDaDarL1	0x148	R/W	0x00000000	DAI1 Tx Left Data Source Address
TxDaDarL2	0x14C	R/W	0x00000000	DAI2 Tx Left Data Source Address
TxDaDarL3	0x150	R/W	0x00000000	DAI3 Tx Left Data Source Address
TxDaCarL1	0x154	R	0x00000000	DAI1 Tx Left Data Current Source Address
TxDaCarL2	0x158	R	0x00000000	DAI2 Tx Left Data Current Source Address

TxDaCarL3	0x15C	R	0x00000000	DAI3 Tx Left Data Current Source Address

Table 11.2 DAI Register Map (Base Address = 0xF0534000)

Name	Address	Type	Reset	Description
DADIR0	0x00	R	-	Digital Audio Input Register 0
DADIR1	0x04	R	-	Digital Audio Input Register 1
DADIR2	0x08	R	-	Digital Audio Input Register 2
DADIR3	0x0C	R	-	Digital Audio Input Register 3
DADIR4	0x10	R	-	Digital Audio Input Register 4
DADIR5	0x14	R	-	Digital Audio Input Register 5
DADIR6	0x18	R	-	Digital Audio Input Register 6
DADIR7	0x1C	R	-	Digital Audio Input Register 7
DADOR0	0x20	R/W	-	Digital Audio Output Register 0
DADOR1	0x24	R/W	-	Digital Audio Output Register 1
DADOR2	0x28	R/W	-	Digital Audio Output Register 2
DADOR3	0x2C	R/W	-	Digital Audio Output Register 3
DADOR4	0x30	R/W	-	Digital Audio Output Register 4
DADOR5	0x34	R/W	-	Digital Audio Output Register 5
DADOR6	0x38	R/W	-	Digital Audio Output Register 6
DADOR7	0x3C	R/W	-	Digital Audio Output Register 7
DAMR	0x40	R/W	0x00000000	Digital Audio Mode Register
DAVC	0x44	R/W	0x0000	Digital Audio Volume Control Register
MCCR0	0x48	R/W	0x00000000	Multi Channel Control Register 0
MCCR1	0x4C	R/W	0x00000000	Multi Channel Control Register 1

The DAMR register must be set after set the MCCR register.

Table 11.3 CDIF Register Map (Base Address = 0xF0534080)

Name	Address	Type	Reset	Description
CDDI_0	0x80	R		CD Digital Audio Input Register 0
CDDI_1	0x84	R		CD Digital Audio Input Register 1
CDDI_2	0x88	R		CD Digital Audio Input Register 2
CDDI_3	0x8C	R		CD Digital Audio Input Register 3
CDDI_4	0x90	R		CD Digital Audio Input Register 4
CDDI_5	0x94	R		CD Digital Audio Input Register 5
CDDI_6	0x98	R		CD Digital Audio Input Register 6
CDDI_7	0x9C	R		CD Digital Audio Input Register 7
CICR	0xA0	R/W	0x0000	CD Interface Control Register

Table 11.4 SPDIF Tx Register Map (Base Address = 0xF0535000)

Name	Address	Type	Reset	Description
TxVersion	0x00	R	0x00003111	Version Register
TxConfig	0x04	R/W	0x00000000	Configuration Register
TxChStat	0x08	R/W	0x00000000	Channel Status Control Register
TxIntMask	0x0C	R/W	0x00000000	Interrupt Mask Register
TxIntStat	0x10	R/W	0x00000000	Interrupt Status Register
UserData	0x80~0xDC	W	-	User Data Buffer
ChStatus	0x100~0x15C	W	-	Channel Status Buffer
TxBuffer	0x200~0x23C	W	-	Transmit Data Buffer
DMACFG	0x400	R/W	0x00000007	Additional Configuration for DMA
CSBUDB	0x680~0x6DC	W	-	Merged Window for CSB/UDB

Table 11.5 SPDIF Rx Register Map (Base Address = 0xF0535800)

Name	Address	Type	Reset	Description

RxVersion	0x800	R	0x00080111	Version Register
RxConfig	0x804	R/W	0x00000000	Configuration Register
RxStatus	0x808	R	0x00000000	Signal Status Buffer
RxIntMask	0x80C	R/W	0x00000000	Interrupt Mask Register
RxIntStat	0x810	R/W	0x00000000	Interrupt Status Register
RxCapCtl [n]	0x840~0x87C(even)	W	0x00000000	Channel Status Capture Control Register
RxCap [n]	0x840~0x87C(odd)	W	0x00000000	Captured Channel Status / user bit
RxBuffer	0xA00~0xA1C	W	-	Receive Data Buffer

11.2.1 Audio DMA Register

DAI0(CDIF)Rx / DAI0(SPDIF)Tx (Right) Data Destination Address Register **0xF0533000(0x10)/0x40(0x50)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RxDaDar															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RxDaDar															

This register contains the start address of target memory block for DAI0 Rx DMA transfer. The transfer begins reading data from this address. When LRMode is asserted, this is used for Right Data of DAI0 Rx.

** Multi Channel:*
- *DAI1 Rx / DAI1 Tx → 0x100 / 0x130*
- *DAI2 Rx / DAI2 Tx → 0x104 / 0x134*
- *DAI3 Rx / DAI3 Tx → 0x108 / 0x138*

DAI(CDIF) Rx / DAI(SPDIF) Tx Parameters Register **0xF0533004(0x14)/0x44(0x54)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMASK															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SMASK								SINC							

SMASK [31:8]	Source Address Mask Register
0	non-masked
N	Masked so that source address bit doesn't be changed during DMA transfer

Each bit field controls the dedicated bit of source address field. That is, if SMASK[23] is set to 1, the 28th bit of source address is masked, and if SMASK[22] is set to 1, the 27th bit of source address is masked, and so on. If a bit is masked, a corresponding bit of address bus is not changed during DMA transfer. This function can be used to generate circular buffer address.

SINC [7:0]	Source Address Increment Register
N	Source address is added by amount of sinc at every write cycles. sinc is represented as 2's complement, so if SINC[7] is 1, the source address is decremented.

The addresses of DMA transfer have 32bit wide, but the upper 4bit of them are not affected during DMA transfer. If the source or destination address reaches its maximum address space like 0x7FFFFFFF or 0x2FFFFFFF, the next transfer is starting from 0x70000000 or 0x20000000 not from 0x80000000 or 0x30000000.

DAI(CDIF) Rx / DAI(SPDIF) Tx Transmission Counter Register 0xF0533008(0x18)/0x48(0x58)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C_TCOUNT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST_TCOUNT															

C_TCOUNT [31:16]	Current Transmisson Count
N	Represent cn number of Hop transfer remains

ST_TCOUNT [15:0]	Start Transmission Count
N	DMA transfers data by amount of sn Hop transfers

At the beginning of transfer, the C_TCNT is updated by ST_TCNT register. At the end of every hop transfer, this is decremented by 1 until it reaches to zero. When this reaches to zero, the DMA finishes its transfer and may or may not generate its interrupt according to IEN flag of CHCTRL register.

DAI0(CDIF)Rx / DAI0(SPDIF)Tx (Right) Data Current Destination Address Register

0xF053300C(0x1C)/0x4C(0x5C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RxDaCdar															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RxDaCdar															

This register contains the current destination address of DAI0 Rx DMA transfer. It represents that the current transfer read data from this address. This is read only register. When LRMode is asserted, this is used for Right Data of DAI0 Rx.

** Multi Channel:*

- *DAI1 Rx / DAI1 Tx → 0x10C / 0x13C*
- *DAI2 Rx / DAI2 Tx → 0x110 / 0x140*
- *DAI3 Rx / DAI3 Tx → 0x114 / 0x144*

DAI0(CDIF)Rx / DAI0(SPDIF)Tx Left Data Destination Address Register

0xF0533028(0x30)/0x68(0x70)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RxDaCdarL															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RxDaCdarL															

This register contains the start address of target memory block for DAI0 Rx DMA transfer. The transfer begins reading data from this address. When LR Mode is asserted, this is used for Left Data of DAI0 Rx, otherwise this is not available.

** Multi Channel:*

- *DAI1 Rx / DAI1 Tx → 0x118 / 0x148*
- *DAI2 Rx / DAI2 Tx → 0x11C / 0x14C*
- *DAI3 Rx / DAI3 Tx → 0x120 / 0x150*

DAI(CDIF) Rx / DAI(SPDIF) Left Data Current Destination Address Register

0xF053302C(0x34)/0x6C(0x74)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RxDaCdarL															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RxDaCdarL															

This register contains the current destination address of DAI0 Rx DMA transfer. It represents that the current transfer read data from this address. This is read only register. When LRMode is asserted, this is used for Left Data of DAI0 Rx, otherwise this is not available.

** Multi Channel:*

- *DAI1 Rx / DAI1 Tx → 0x124 / 0x154*
- *DAI2 Rx / DAI2 Tx → 0x128 / 0x158*
- *DAI3 Rx / DAI3 Tx → 0x12C / 0x15C*

Transfer Control Register

0xF0533038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R*		RCN	TCN	CRLCK	DRLCK	STLCK	DTLCK	CRTRG	DRTRG	STTRG	DTTRG	CRRPT	DRRPT	STRPT	DTRPT
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRBSIZE		DRBSIZE		STBSIZE		DTBSIZE		CRWSIZE		DRWSIZE		STWSIZE		DTWSIZE	

RCN [29]	Issue Continuous Transfer of Rx DMA
0	DMA transfer begins from source/destination address
1	DMA transfer begins from current source/destination address It must be used after the former transfer has been executed, so that current source/destination address registers contain a meaningful value.

TCN [28]	Issue Continuous Transfer of Tx DMA
0	DMA transfer begins from source/destination address
1	DMA transfer begins from current source/destination address It must be used after the former transfer has been executed, so that current source/destination address registers contain a meaningful value.

CRLCK [27]	Issue Locked Transfer of CDIF Rx DMA
1	DMA transfer executed with lock transfer

Lock field controls the LOCK signal (refer to AHB specification). When the LOCK is set to 1, the DMA transfer doesn't be bothered by other AHB masters like LCD controller, ARM etc.

DRLCK [26]	Issue Locked Transfer of DAI Rx DMA
1	DMA transfer executed with lock transfer

STLCK [25]	Issue Locked Transfer of SPDIF Tx DMA
1	DMA transfer executed with lock transfer

DTLCK [24]	Issue Locked Transfer of DAI Tx DMA
1	DMA transfer executed with lock transfer

CRTRG [23]	Trigger Type of Rx DMA
0	SINGLE edge-triggered detection
1	SINGLE level-sensitive detection

In SINGLE Type, After one Hop data transferring DMA checks External DMA Request (DREQ) and then if its bit is active , DMA transfers next hop data . DREQ is detected level-sensitive or edge-triggered by SINGLE transfer TYPE.

DRTRG [22]	Trigger Type of Rx DMA
0	SINGLE edge-triggered detection
1	SINGLE level-sensitive detection

STTRG [21]	Trigger Type of Tx DMA
0	SINGLE edge-triggered detection

1	SINGLE level-sensitive detection

DTTRG [20]	Trigger Type of Tx DMA
0	SINGLE edge-triggered detection
1	SINGLE level-sensitive detection

CRRPT [19]	Repeat Mode Control of CDIF Rx DMA
0	After all of hop transfer has executed, the DMA channel is disabled
1	The DMA channel remains enabled. When another DMA request has occurred, the DMA channel start transfer data again with the same manner (type, address, increment, mask) as the latest transfer of that channel.

DRRPT [18]	Repeat Mode Control of DAI Rx DMA
0	After all of hop transfer has executed, the DMA channel is disabled
1	The DMA channel remains enabled. When another DMA request has occurred, the DMA channel start transfer data again with the same manner (type, address, increment, mask) as the latest transfer of that channel.

STRPT [17]	Repeat Mode Control of SPDIF Tx DMA
0	After all of hop transfer has executed, the DMA channel is disabled
1	The DMA channel remains enabled. When another DMA request has occurred, the DMA channel start transfer data again with the same manner (type, address, increment, mask) as the latest transfer of that channel.

DTRPT [16]	Repeat Mode Control of DAI Tx DMA
0	After all of hop transfer has executed, the DMA channel is disabled
1	The DMA channel remains enabled. When another DMA request has occurred, the DMA channel start transfer data again with the same manner (type, address, increment, mask) as the latest transfer of that channel.

The 1 Hop of transfer means 1 burst of read followed by 1 burst of write. 1 burst means 1, 2 or 4 consecutive read or write cycles defined by BSIZE field of CHCTRL register. The Figure1.1 illustrates 1 Hop of transfer

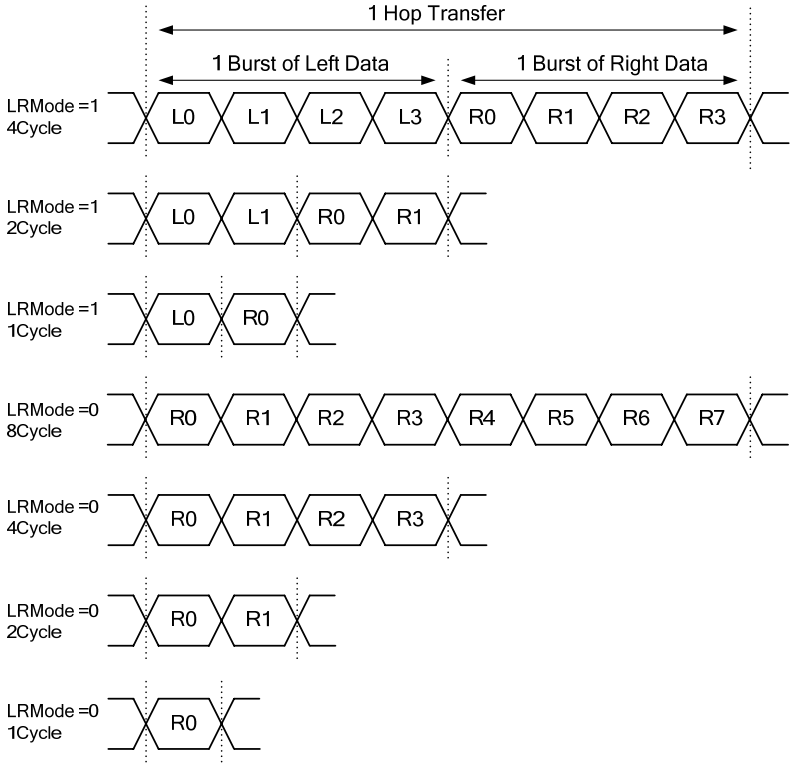


Figure 11.6 Relation between Hop and Burst Transfers

CRBSIZE [15:14]	Burst Size of CDIF Rx DMA
0	1 Burst transfer consists of 1 read or write cycle.
1	1 Burst transfer consists of 2 read or write cycles
2	1 Burst transfer consists of 4 read or write cycles

3	1 Burst transfer consists of 8 read or write cycles

When LRMode is enabled, 8 Read/Write is not available.

DRBSIZE [13:12]	Burst Size of DAI Rx DMA
0	1 Burst transfer consists of 1 read or write cycle.
1	1 Burst transfer consists of 2 read or write cycles
2	1 Burst transfer consists of 4 read or write cycles
3	1 Burst transfer consists of 8 read or write cycles

When LRMode is enabled, 8 Read/Write is not available.

STBSIZE [13:12]	Burst Size of SPDIF Tx DMA
0	1 Burst transfer consists of 1 read or write cycle.
1	1 Burst transfer consists of 2 read or write cycles
2	1 Burst transfer consists of 4 read or write cycles
3	1 Burst transfer consists of 8 read or write cycles

When LRMode is enabled, 8 Read/Write is not available.

DTBSIZE [9:8]	Burst Size of DAI Tx DMA
0	1 Burst transfer consists of 1 read or write cycle.
1	1 Burst transfer consists of 2 read or write cycles
2	1 Burst transfer consists of 4 read or write cycles
3	1 Burst transfer consists of 8 read or write cycles

When LRMode is enabled, 8 Read/Write is not available.

CRWSIZE [7:6]	Word Size of CDIF Rx DMA
0	Each cycle read or write 8bit data
1	Each cycle read or write 16bit data
2,3	Each cycle read or write 32bit data

DRWSIZE [5:4]	Word Size of DAI Rx DMA
0	Each cycle read or write 8bit data
1	Each cycle read or write 16bit data
2,3	Each cycle read or write 32bit data

STWSIZE [3:2]	Word Size of SPDIF Tx DMA
0	Each cycle read or write 8bit data
1	Each cycle read or write 16bit data
2,3	Each cycle read or write 32bit data

DTWSIZE [1:0]	Word Size of DAI Tx DMA
0	Each cycle read or write 8bit data
1	Each cycle read or write 16bit data
2,3	Each cycle read or write 32bit data

Repeat Control Register **0xF053303C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRI	Reserved			DBTH				RPTCNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RPTCNT															

DRI [31]	Disable Repeat Interrupt
0	DMA Interrupt is occurred when the end of each Repeated DMA operation.
1	DMA Interrupt occur is occurred when the last DMA Repeated DMA operation

This bit is meaningful when Repeat Mode is enabled.

DBTH [27:24]	DAI Buffer Threshold
N	DMA transfer buffer threshold determines DMA transfer trigger condition

This field sets the threshold condition of internal 64-depth buffer for DAI DMA. If audio data increases more than

(DBTH+1)*2^DTBSIZE, the internal DMA starts the transfer. When MPE bit is enabled, DBTH should be fixed to 7 and also DTBSIZE should be 2.

RPTCNT [15:0]	Repeat Count
0	DMA transfer will be repeated endlessly.
N	DMA transfer will be repeated (N + 1) * TCOUNT times

This bit is meaningful when Repeat Mode is enabled. When this bit is cleared in repeat mode, DMA transfer will be repeated endlessly. To exit endless repeat mode, clear EN bit of ChCtrl or disable Repeat Mode. It's possible to circular transfer using repeat count.

Channel Control Register **0xF0533078**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CREN	DREN	STEN	DTEN	R*	CSS	DRMCM	DTMCM	DMRSEL		DMTSEL		CRLR	DRLR	STLR	DTLR
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRDW	DREW	STDW	DTDW	CRSEN	DRSEN	STSEN	DTSEN	CRWB	DRWB	STWB	DTWB	CREN	DREN	STEN	DTEN

CREN [31]	DMA Channel Enable of CDIF Rx
0	DMA channel is terminated and disabled. It does not affect the TCOUNT register, so if the current hop counter is not zero when channel is disabled, it is possible that the transfer illegally starts right after channel is re-enabled. Make sure that TCOUNT is zero not to continue transfer after channel is re-enabled.
1	DMA channel is enabled.

DREN [30]	DMA Channel Enable of DAI Rx
0	DMA channel is terminated and disabled.
1	DMA channel is enabled.

STEN [29]	DMA Channel Enable of SPDIF Tx
0	DMA channel is terminated and disabled.
1	DMA channel is enabled.

DTEN [28]	DMA Channel Enable of DAI Tx
0	DMA channel is terminated and disabled.
1	DMA channel is enabled.

CSS [26]	DMA Channel Enable of DAI Tx
0	CDIF Rx Mode
1	SPDIF Rx Mode

DRMCM [25]	DMA Multi Channel Mode of DAI Rx
0	Single Channel Mode
1	Multi Channel Mode

DTMCM [24]	CDIF or SPDIF Select
0	Single Channel Mode
1	Multi Channel Mode

DMRSEL [23:22]	DMA Multi Channel Select of DAI Rx
00	DAI0 & DAI1 Channel Select (3.1Ch)
01	DAI0 & DAI1 & DAI2 Channel Select (5.1Ch)
11	DAI0 & DAI1 & DAI2 & DAI3 channel select (7.1Ch)

DMTSEL [21:20]	DMA Multi Channel Select of DAI Tx
00	DAI0 & DAI1 Channel Select (3.1Ch)
01	DAI0 & DAI1 & DAI2 Channel Select (5.1Ch)
11	DAI0 & DAI1 & DAI2 & DAI3 channel select (7.1Ch)

CRLR [19]	Left/Right Data Mode of CDIF Rx
0	Disable LRMode.
1	Enable LRMode. When LRMode is enabled, audio data will be written to memory separately according to whether it is left or right one.

DRLR [18]	Left/Right Data Mode of DAI Rx
0	Disable LRMode.
1	Enable LRMode.

STLR [17]	Left/Right Data Mode of SPDIF Tx
0	Disable LRMode.
1	Enable LRMode.

DTLR [16]	Left/Right Data Mode of DAI Tx
0	Disable LRMode.

1	Enable LRMode.

CRDW [15]	Width of Audio Data of CDIF Rx
0	Assume width of audio data is 24bits.
1	Assume width of audio data is 16bits. AUDIO DMA manipulates audio data as 16bit-length one. If it gets 24bit-length audio data from audio devices, it will cut out lsb-16bits and put those into DMA FIFO. For more details, refer to attached pictures.

DRDW [14]	Width of Audio Data of DAI Rx
0	Assume width of audio data is 24bits.
1	Assume width of audio data is 16bits.

STDW [13]	Width of Audio Data of SPDIF TX
0	Assume width of audio data is 24bits.
1	Assume width of audio data is 16bits.

DTDW [12]	Width of Audio Data of DAI Tx
0	Assume width of audio data is 24bits.
1	Assume width of audio data is 16bits.

CRSEN [11]	Swapping Half-word/Byte align of CDIF Rx
0	Disable swapping.
1	Swaps half-word or byte align according to WB bits.

DRSEN [10]	Swapping Half-word/Byte align of DAI Rx
0	Disable swapping.
1	Swaps half-word or byte align according to WB bits.

STSEN [9]	Swapping Half-word/Byte align of SPDIF Tx
0	Disable swapping.
1	Swaps half-word or byte align according to WB bits.

DTSEN [8]	Swapping Half-word/Byte align of DAI Tx
0	Disable swapping.
1	Swaps half-word or byte align according to WB bits.

CRWB [7]	Choose Half-word/Byte of CDIF Rx
0	Byte swapping. This is available only when SWP is enabled.
1	Half-word swapping. This is available only when SWP is enabled.

DRWB [6]	Choose Half-word/Byte of DAI Rx
0	Byte swapping. This is available only when SWP is enabled.
1	Half-word swapping. This is available only when SWP is enabled.

STWB [5]	Choose Half-word/Byte of SPDF Tx
0	Byte swapping. This is available only when SWP is enabled.
1	Half-word swapping. This is available only when SWP is enabled.

DTWB [4]	Choose Half-word/Byte of DAI Tx
0	Byte swapping. This is available only when SWP is enabled.
1	Half-word swapping. This is available only when SWP is enabled.

CRIEN [3]	Interrupt Enable of CDIF Rx
1	At the same time the CRI goes to 1, DMA interrupt request is generated.

To generate IRQ or FIQ interrupt, the DMA flag of IEN register in the interrupt controller must be set to 1 ahead.

DRIEN [2]	Interrupt Enable of DAI Rx
1	At the same time the CRI goes to 1, DMA interrupt request is generated.

STIEN [1]	Interrupt Enable of SPDIF Tx
1	At the same time the CRI goes to 1, DMA interrupt request is generated.

DTIEN [0]	Interrupt Enable of DAI Tx
1	At the same time the CRI goes to 1, DMA interrupt request is generated.

DMA Interrupt Status Register **0xF053307C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CRI	DRI	STI	DTI	CRMI	DRMI	STMI	DTMI

CRI [7]	DMA Interrupt Status of CDIF Rx
0	No interrupt occurred while CDIF Rx DMA transfer.
1	Interrupt occurred while CDIF Rx DMA transfer.

Without regard to Interrupt enable bit(CRIEN) of ChCtrl, this bit indicates the CDIF Rx DMA interrupt status.

This bit is automatically cleared when FLAG bit of ChCtrl is cleared. This bit is read only.

DRI [6]	DMA Interrupt Status of DAI Rx
0	No interrupt occurred while DAI Rx DMA transfer.
1	Interrupt occurred while DAI Rx DMA transfer.

Without regard to Interrupt enable bit(DRIEN) of ChCtrl, this bit indicates the DAI Rx DMA interrupt status.

This bit is automatically cleared when FLAG bit of ChCtrl is cleared. This bit is read only.

STI [5]	DMA Interrupt Status of SPDIF Tx
0	No interrupt occurred while SPDIF Tx DMA transfer.
1	Interrupt occurred while SPDIF Tx DMA transfer.

Without regard to Interrupt enable bit(STIEN) of ChCtrl, this bit indicates the SPDIF Tx DMA interrupt status.

This bit is automatically cleared when FLAG bit of ChCtrl is cleared. This bit is read only.

DTI [4]	DMA Interrupt Status of DAI Tx
0	No interrupt occurred while DAI Tx DMA transfer.
1	Interrupt occurred while DAI Tx DMA transfer.

Without regard to Interrupt enable bit(DTIEN) of ChCtrl, this bit indicates the DAI Tx DMA interrupt status.

This bit is automatically cleared when FLAG bit of ChCtrl is cleared. This bit is read only.

CRMI [3]	DMA Masked Interrupt Status of CDIF Rx
0	No interrupt occurred while CDIF Rx DMA transfer.
1	Interrupt occurred while CDIF Rx DMA transfer.

This bit indicates the CDIF Rx DMA interrupt status when CRIEN is asserted.

DRMI [2]	DMA Masked Interrupt Status of DAI Rx
0	No interrupt occurred while DAI Rx DMA transfer.
1	Interrupt occurred while DAI Rx DMA transfer.

This bit indicates the DAI Rx DMA interrupt status when DRIEN is asserted.

STMI [1]	DMA Masked Interrupt Status of CDIF Rx
0	No interrupt occurred while SPDIF Tx DMA transfer.
1	Interrupt occurred while SPDIF Tx DMA transfer.

This bit indicates the SPDIF Tx DMA interrupt status when STIEN is asserted.

DTMI [0]	DMA Masked Interrupt Status of CDIF Rx
0	No interrupt occurred while DAI Tx DMA transfer.
1	Interrupt occurred while DAI Tx DMA transfer.

This bit indicates the DAI Tx DMA interrupt status when DTIEN is asserted.

General Interrupt Register (Status Register) **0xF0533080(0x84)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								GSTDI	GSTUI	GSTI	GSRDI	GSRI	GDTI	GDRI	GCRI
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved											GDSTDI	GDSTUI	GDDTI	GDDRI	GDCRI

GSTDI [23]	General Interrupt Enable of SPDIF Data Tx
0	Disable
1	Enable

GSTUI [22]	General Interrupt Enable of SPDIF User Tx
0	Disable
1	Enable

GSTI [21]	General Interrupt Enable of SPDIF Tx
0	Disable
1	Enable

GSRDI [20]	General Interrupt Enable of SPDIF Data Rx
0	Disable
1	Enable

GSRI [19]	General Interrupt Enable of SPDIF Rx
0	Disable
1	Enable

GDTI [18]	General Interrupt Enable of DAI Tx
0	Disable
1	Enable

GDRI [17]	General Interrupt Enable of DAI Rx
0	Disable
1	Enable

GCRI [16]	General Interrupt Enable of CDIF Rx
0	Disable
1	Enable

GDSTDI [4]	General DMA Interrupt Enable of SPDIF Data Tx
0	Disable
1	Enable

GDSTUI [3]	General DMA Interrupt Enable of SPDIF User Tx
0	Disable
1	Enable

GDDTI [2]	General DMA Interrupt Enable of DAI Tx
0	Disable
1	Enable

GDDRI [1]	General DMA Interrupt Enable of DAI Rx
0	Disable
1	Enable

GDCRI [0]	General DMA Interrupt Enable of CDIF Rx
0	Disable
1	Enable

11.2.2 DAI Register

Digital Audio Mode Register (DAMR) **0xF0534040**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

BPS	LPS	DBTEN	MPE	R*	DDL	DSP	NEWR	NEWT	RXE	RXS		TXS		LBT	SP
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EN	TE	RE	MD	SM	BM	FM	CC	BD		FD		BP	CM	MM	LB

BPS [31]	BCLK Pad Select
0	BCLK source select BCLK Pad
1	BCLK direct at master mode

LPS [30]	LRCK Pad Select
0	LRCK source select LRCK Pad
1	LRCK direct at master mode

DBTEN [29]	DAI Buffer Threshodl Enable
0	Disable
1	Enable

This bit enables DAI buffer threshold (DBTH) control register. If it is set to low, DAI DMA transfer starts right after new data is stored to DAI buffer. When MPE bit is set to high, this field should be set to high not to get wrong transfer in multi-port condition.

MPE [28]	Multi-Port Enable
0	Disable
1	Enable

In multi-port condition, DAI DMA transfer mode is fixed to one mode. That is, DBTEN, DBTH, DTBSIZE should be fixed to 1, 7 and 2 each.

DDL [26]	DSP Mode Word Length
0	Data Length 24bit
1	Data Length 16bit

This selects the number of bit width in Wolfson DSP & TDM mode. In Wolfson TDM mode, MCCR1 must be set additionally.

DSP [25]	DSP Mode
0	IIS
1	DSP or TDM Mode

MD =‘0’ & DSP=‘1’→ DSP Mode
MD =‘0’ & DSP=‘1’& FD= “11” → TDM Mode

NMDR [24]	Rx Justfied Mode
0	Left-justified Mode
1	Right-justified Mode

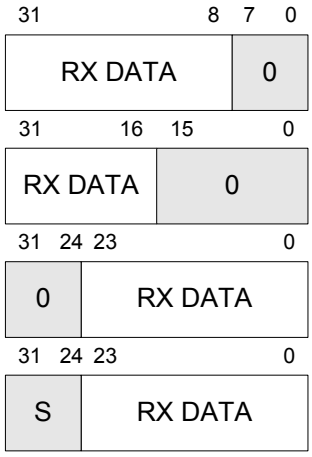
MD =‘1’ & DSP=‘0’ & NMDR=‘0’ → Left-justified Rx Mode
MD =‘1’ & DSP=‘0’ & NMDR=‘1’ → Right-justified Rx Mode

NMDT [23]	Tx Justified Mode
0	Left-justified Mode
1	Right-justified Mode

MD =‘1’ & DSP=‘0’ & NMDT=‘0’ → Left-justified Tx Mode
MD =‘1’ & DSP=‘0’ & NMDR=‘1’ → Right-justified Tx Mode

RXE [22]	DAI RX Data Sign Extension
0	Disable (zero extension)
1	Enable (sign bit extension)

RXS [21:20]	DAI RX Shift
00	Bit-pack MSB and 24bit mode.
01	Bit-pack MSB and 16bit mode.
10	Bit-pack LSB and 24bit mode.
11	Bit-pack LSB and 16bit mode.



TXS [19:18]	DAI TX Shift
0x	Bit-pack MSB mode.
10	Bit-pack LSB and 24bit mode.
11	Bit-pack LSB and 16bit mode.

If DMA transfer bit width is 16, DAI TX Shift register should be set to "Bit-pack LSB and 16bit mode"(TXS=3). When DAI TX Shift register is "Bit-pack LSB and 24bit mode", Most Significant Byte of the second 16bit is filled with 0.

LBT [17]	Loop Back Test(Tx to Rx)
0	Disable
1	Enable

This is for loop back test, i.e. TX port of DAI will be feedback into RX port.

SP [16]	DAI System Clock Polatity
0	Disable
1	Enable

EN [15]	DAI Master Enable
0	Disable DAI module
1	Enable DAI module

TE [14]	DAI Transmitter Enable
0	Disable DAI transmitter
1	Enable DAI transmitter

RE [13]	DAI Receiver Enable
0	Disable DAI receiver
1	Enable DAI receiver

MD [12]	DAI Sync Mode
0	Set DAI sync as IIS, DSP or TDM sync mode
1	Set DAI sync as Left/Right-justified mode

SM [11]	DAI System Clock Master Select
0	Set that DAI system clock is come from external pin
1	Set that DAI system clock is generated by the clock generator block

The DAI system clock in clock generator is known as DCLK. Its frequency can be determined by PCK_DAI from the CKC.

BM [10]	DAI Bit Clock Master Select
0	Set that DAI bit clock is from external pin
1	Set that DAI bit clock is generated by dividing DAI system clock

FM [9]	DAI Frame Clock Master Select
0	Set that DAI frame clock is come from external pin
1	Set that DAI frame clock is generated by dividing DAI bit clock

CC [8]	CDIF Clock Select
0	Disable CDIF Clock master mode
1	Enable CDIF Clock master mode

BD [7:6]	DAI Bit Clock Divider select
00	Select Div 4 (256fs->64fs)
01	Select Div 6 (384fs->64fs)
10	Select Div 8 (512fs->64fs, 384fs->48fs , 256fs->32fs)
11	Select Div16 (512fs->32fs)

FD [5:4]	DAI Frame Clock Divider select
00	Select Div 32 (32fs->fs)
01	Select Div 48 (48fs->fs)
10	Select Div 64 (64fs->fs)
11	Select Div use (xfs->fs) ← Only TDM Mode

The combination of BD & FD determines that the ratio between main system clock and the sampling frequency. The multiplication between the division factor of BD and FD must be equal to this ratio. In TDM mode, FD should be set to 3.

BP [3]	DAI Bit Clock Polarity
0	Set that data is captured at positive edge of bit clock
1	Set that data is captured at negative edge of bit clock

CM [2]	CDIF Monitor Mode
0	Disable CDIF monitor mode
1	Enable CDIF monitor mode. Data bypass from CDIF

MM [1]	DAI Monitor Mode
0	Disable DAI monitor mode
1	Enable DAI monitor mode. Transmitter should be enabled. (TE = 1)

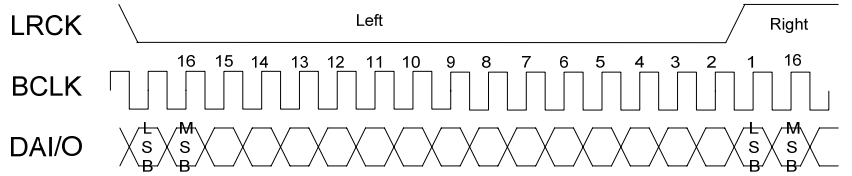
LB [0]	DAI Loop-back Mode
0	Disable DAI Loop back mode
1	Enable DAI Loop back mode

Digital Audio Volume Control Register (DAVC) 0xF0534044

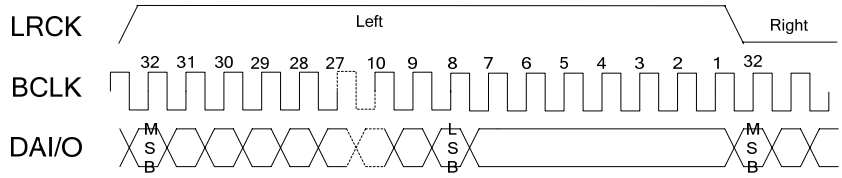
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved											VC<4:0>				

The volume of audio output can be manipulated by this register. It has –6dB unit so the output volume can be set from 0 dB to –90 dB as the following table.

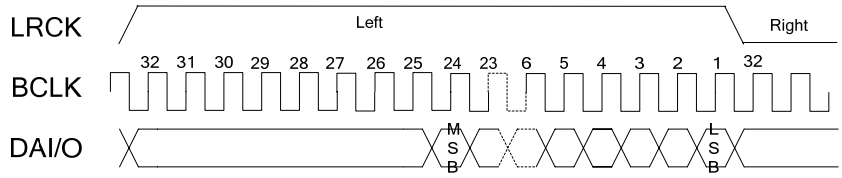
VC [4:0]	DAI Volume control
00000	0dB
00001	-6dB
00010	-12dB
00011	-18dB
00100	-24dB
00101	-30dB
00110	-36dB
00111	-42dB
01000	-48dB
01001	-54dB
01010	-60dB
01011	-66dB
01100	-72dB
01101	-78dB
01110	-84dB
01111	-90dB
10000	-96dB



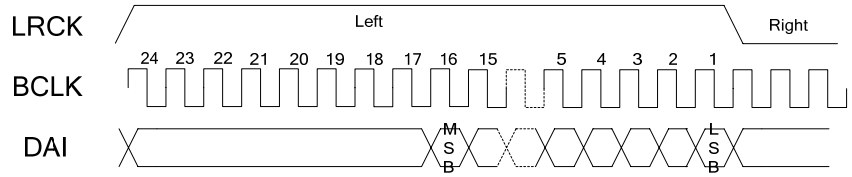
NMDR(T)=0, DSP=0, MD=0 (IIS mode), BP=0, BCLK = 32fs



NMDR(T)=0, DSP=0, MD=1(MSB justified mode), BP=0, BCLK=64fs



NMDR(T)=1, DSP=0, MD=1(LSB justified mode), BP=0, BCLK=64fs



NMDR(T)=1, DSP=0, MD=1(MSB justified mode), BP=1, BCLK=48fs

Figure 11.7 DAI Bus Timing Diagram

Multi Channel Control Register 0 (MCCR0) 0xF0534048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0	CH1	CH2	CH3	Reserved		FBP	FEP								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TFI		TBD		FD		FI	FS								

CH0 [31]	Channel 0
0	Disable Channel 0
1	Enable Channel 0

CH1 [30]	Channel 1
0	Disable Channel 1
1	Enable Channel 1

CH2 [29]	Channel 2
0	Disable Channel 2
1	Enable Channel 2

CH3 [28]	Channel 3
0	Disable Channel 3
1	Enable Channel 3

FBP [25]	Frame Begin Position(DSP & TDM)
0	Frame Begin Position → Early Mode
1	Frame Begin Position → Late Mode

FBP [24:16]	Frame End Position(DSP & TDM)
N	Frame pulse ends after n base clock has generated

LRCK Duty of TDM Mode can be adjusted.

TFI [15:14]	Format Interface of TDM
00	TDM Mode 0 (CIRRUSS, DSP Mode like)
01	TDM Mode 1 (Wolfson, DSP Mode like)
10	TDM Mode 2 (I2S Mode like)

TBD [13:12]	Bit Clock Divider Select of TDM
00	Disable
01	Select Div 1 (256fs -> 256fs)
10	Select Div 2 (512fs -> 256fs, 256fs -> 128fs)

FD [11:10]	Frame Clock Divider Select (DSP & TDM)
00	Select Div 32(32fs -> fs)
01	Select Div 48 (48fs -> fs)
10	Select Div 64 (64fs -> fs)
11	Select Div use (xfs -> fs)

FI [9]	Frame Invert(DSP & TDM)
0	Disable
1	Enable

FS [8:0]	Frame Size(DSP & TDM)
N	Frame pulse ends after n base clock has generated

In DSP or TDM mode, the number of audio samples in a single LRCK duration.

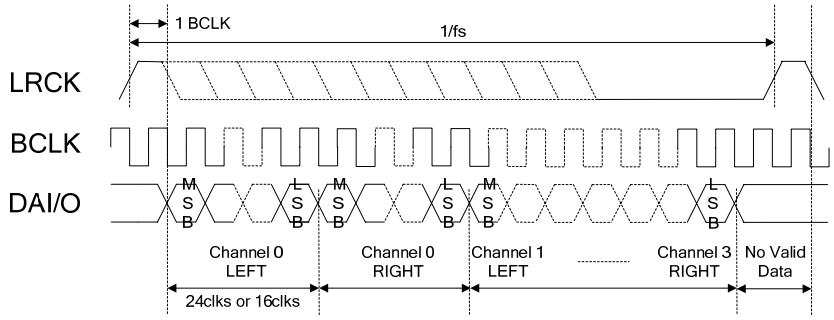
Multi Channel Control Register 1 (MCCR1) 0xF053404C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							VDE								

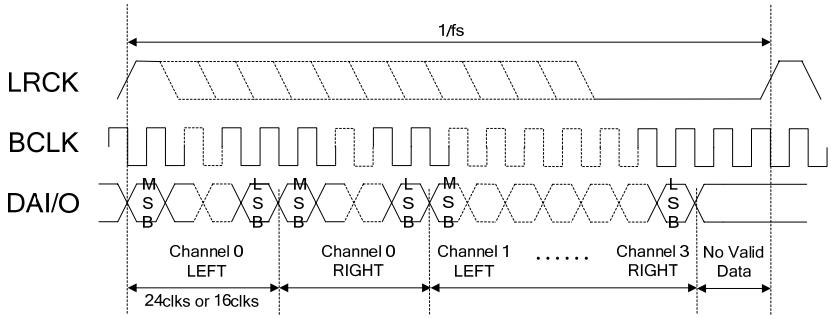
VDE [8:0]	Valid Data End

N	Valid Data ends after n base clock has generated

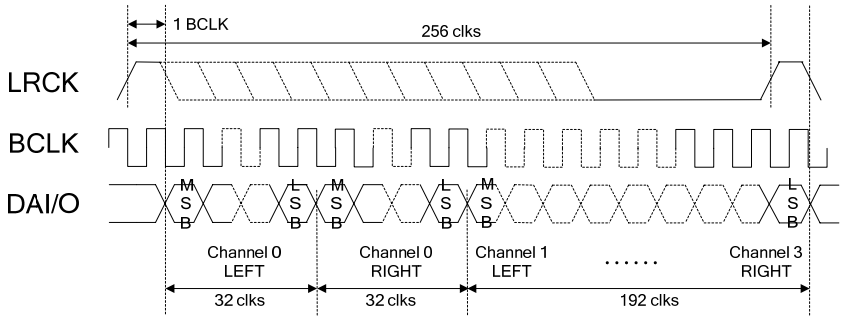
When in TDM mode 1, VDE bits sets the valid data length out of 16bit or 24bit input data.



(a) TDM Mode Example 1 (Wolfson – late mode)
(DAMR → DSP=1, MD=0, FD=3, BP=1
MCCR0 → FBP=0, TFI=1(W), TBD=1, FD=11
MCLK1=BCLK=256fs)



(b) TDM Mode Example 2 (Wolfson – early mode)
(DAMR → DSP=1, MD=0, FD=3, BP=1
MCCR0 → FBP=1, TFI=1(W), TBD=1, FD=11
MCLK1=BCLK=256fs)



(c) TDM Mode Example 3 (Cirrus)
(DAMR → DSP=1, MD=0, FD=3, BP=1
MCCR0 → FBP=0, TFI=0(C), TBD=1, FD=11
MCLK1=BCLK=256fs)

Figure 11.8 DAI TDM Mode Timing Diagram

11.2.3 CDIF Registers

CD Data Input (CDDI0~CDDI7) **0xF0534080 ~ 0xF053409C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Left Channel Data															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Right Channel Data															

CD Interface Control Register (CICR) **0xF05340A0**

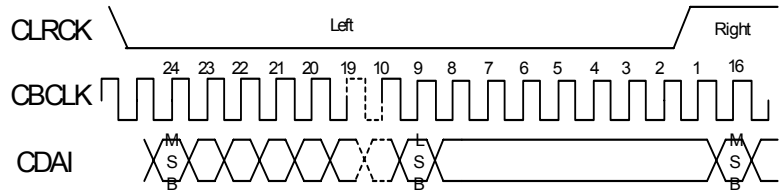
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved								EN	Rserved			BS		MD	BP

EN [7]	CDIF Enable
0	Disable CDIF
1	Enable CDIF

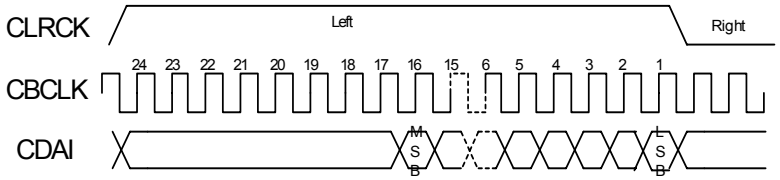
BS [3:2]	CDIF Bit Clock select
00	64fs
01	32fs
10	48fs

MD [1]	Interface Mode select
0	Select IIS format
1	Select LSB justified format

BP [0]	CDIF Bit Clock Polarity
0	Set that data is captured at positive edge of bit clock
1	Set that data is captured at negative edge of bit clock



MD=0 (IIS mode), BP=0, CBCLK=48fs



MD=1(LSB justified mode), BP=0, CBCLK=48fs

Figure 11.9 CDIF Bus Timing Diagram

11.2.4 SPDIF Registers

SPDIF Transmitter Version Register **0xF0535000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CSB	UDB	AW						DW	VER			

CSB [12]	Channel Status Buffer Available Bit
0	Channel status buffer is not available
1	Channel status buffer is available

UDB[11]	User Data Buffer Available Bit
0	User data buffer is not available
1	User data buffer is available

AW[10:5]	Value of Address Width
n	Value of Address Width

DW[4]	Value of Data Width
n	Value of Data Width

VER[3:0]	Design Version
n	Version Number

SPDIF Transmit Configuration Register 0xF0535004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								MODE				Reserved			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RATIO								UDATEN		CHSTEN		R*	IEN	TXD	TXEN

MODE[23:20]	Sample Format Mode Indicator
n	16 + n Bits Transmit Sample Format
9 ~ 15	Reserved

RATIO[15:8]	Clock Divider Ratio
n	Clock divider for the transmit frequency. The SPDIF clock is divided by a factor of (RATIO + 1) to generate the serial transmit clock.

UDATEN[7:6]	User Data Enable Bits
0	User data A & B set to 0.
1	User data A & B generated from UserData bit 7-0
2	User data A generated from UserData bit 7-0, B generated from UserData bit 15-8
3	Reserved

CHSTEN[5:4]	Channel Status Enable Bits
0	Channel status A & B generated from TxChStat
1	Channel status A & B generated from ChStat bit 7-0
2	Channel status A generated from ChStat bit 7-0, B generated from ChStat bit 15-8
3	Reserved

IEN[2]	Interrupt Output Enable Bit
n	'1' for enabling the interrupt output.

TXD[1]	Data Valid Bit
n	'1' for data being valid.

TXEN[0]	Transmitter Enable Bit
n	'1' for enabling the transmission.

SPDIF Transmit Channel Status Control Register 0xF0535008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					SCS	SUD	SV	FREQ		Reserved		GSTS	PRE	CPY	AU

SCS[10]	Store Channel Status bit
0	No store
1	Store

SUD[9]	Store User Data bit
0	No store
1	Store

SV[8]	Store Validity bit
0	No store
1	Store

FREQ[7:6]	Sampling Frequency
0	44.1kHz
1	48kHz
2	32kHz
3	Sample Rate Converter

GSTS[3]	Status Generation
0	No indication
1	Original/Commercially Pre-recorded data

PRE[2]	Pre-emphasis
0	None
1	50/15us

CPY[1]	Copyright
0	Copy inhibited
1	Copy permitted

AU[0]	Data Format
0	Audio Format
1	Data Format

SPDIF Transmit Interrupt Mask Register 0xF053500C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											HCSB	LCSB	HSB	LSB	R*

BIT	NAME	DESCRIPTION
4	HCSB	Higher Channel Status/User Data Buffer Empty '1' for enable for higher channel status/user data buffer empty interrupt
3	LCSB	Lower Channel Status/User Data Buffer Empty '1' for enable for lower channel status/user data buffer empty interrupt
2	HSB	Higher Data Buffer Empty '1' for enable for higher data buffer empty interrupt
1	LSB	Lower Data Buffer Empty '1' for enable for lower data buffer empty interrupt

SPDIF Transmit Interrupt Status Register 0xF0535010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											HCSB	LCSB	HSB	LSB	R*

Bit	Name	Description
4	HCSB[4]	Higher Channel Status/User Data Buffer Empty '1' for higher channel status/user data buffer empty interrupt activated
3	LCSB[3]	Lower Channel Status/User Data Buffer Empty '1' for lower channel status/user data buffer empty interrupt activated
2	HSB[2]	Higher Data Buffer Empty '1' for higher data buffer empty interrupt activated
1	LSB[1]	Lower Data Buffer Empty '1' for lower data buffer empty interrupt activated

SPDIF Transmit User Data Buffer Register 0xF0535080 ~ 0xF05350DC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHBUD								CHAUD							

The user data is 24 bytes per sample block. Bit 0 of byte 0 is transmitted first. Bit 7 of byte 23 is transmitted last.

Bit	Name	Description
15:8	CHBUD[15:8]	User Data for Channel B 8 Bits User Data for Channel B
7:0	CHAUD[7:0]	User Data for Channel A 8 Bits User Data for Channel A

SPDIF Transmit Channel Status Buffer Register 0xF0535100 ~ 0xF053515C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHBCS								CHACS							

The channel status data is 24 bytes per sample block. Bit 0 of byte 0 is transmitted first. Bit 7 of byte 23 is transmitted last.

bit	name	description
15:8	CHBCS[15:8]	Channel Status for Channel B 8 Bits Channel Status for Channel B
7:0	CHACS[7:0]	Channel Status for Channel A 8 Bits Channel Status for Channel A

SPDIF Transmit Sample Data Buffer Register 0xF0535200 ~ 0xF05353FC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								DATH							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATL															

The format of data words in transmit sample buffer. Channel A is transmitted first, and must be stored on even addresses, while channel B is stored on odd addresses in the sample buffer.

BIT	name	description
23:16	DATH	Upper 8 Bits for Sample Buffer Data
		Audio data if > 16 bits resolution. Unused bits are 0
15:0	DATL	Lower 16 Bits for Sample Buffer Data
		Audio data (16 bits mode). Bits 0 is LSB

DMA Configuration (DMACFG) **0xF0535400**

Field	Name	RW	Reset	Description
31 ~ 21	Reserved	R	0	Reserved
20 ~ 16	VCNT	R	X	FIFO Valid Entry Count
15	Reserved	R	0	Reserved
14	SS	R/W	0	Swap Sample 0 : Disable 1 : Enable
13 ~ 12	RALR	R/W	0	Read Address LR Mode 0 : Read Address 24bit Mode 1 : Read Address 24bit LR Mode 2 : Read Address 16bit Mode 3 : Read Address 16bit LR Mode
11	DRQEN1	RW	0	DMA Request Enable for User Data Buffer
10	DRQEN0	RW	0	DMA Request Enable for Sample Data Buffer
9	AMODE1	RW	0	Sample Data Buffer Address Mode. Buffer would be written with sequence below. (0, 2, 4, 6, 1, 3, 5, 7)
8	AMODE0	RW	0	0 : Ignore Address (FIFO Mode) 1 : Enable Address (16 Entries can be written with address)
7	FIFOCLR	RW	0	Clear FIFO. Should be written with “0” for normal operation
6 ~ 4	Reserved	R	0	Reserved
3 ~ 0	FIFOTH	RW	X	FIFO Threshold for DMA Request DMA request will be asserted if VCNT <= FIFOTH

SPDIF Receiver Version Register 0xF0535800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CSC			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				AW							DW	VER			

The version register allows the SW to read out all the parameter that was used to generate the receiver.

CSC [12]	Channel Status Buffer Available Bit
0	Channel status buffer is not available
1	Channel status buffer is available

AW[10:5]	Value of Address Width
n	Value of Address Width

DW[4]	Value of Data Width
0	Data Width is 16bit
1	Data Width is 24bit

VER[3:0]	Design Version
n	Version Number

SPDIF Receiver Configuration Register 0xF0535804

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							BLKEN	MODE				PAREN	STATEN	USEREN	VALEN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											VALID	CHAS	RINTEN	SAMPLE	RXEN

BLKEN[24]	Block Boundary Marking
0	Do not use lock boundary marking
1	Mark the first sample in each block with a 1 in bit 27

MODE[23:20]	Sample Format Mode Indicator
0	Store sample as 16bit
1	Store sample as 17bit
2~6	Store sample as 18, 19, 20, 21, 22bit
7	Store sample as 23bit
8	Store sample as 24bit
9~15	Reserved

PAREN[19]	Store Parity Bit
0	Do not use block boundary marking
1	Store parity bit 31 in sample buffer

STATEN[18]	Store Channel Status Bit
0	Do not store channel status bit
1	Store channel status bit in bit 30 in sample buffer

USEREN[17]	Store User Data Bit
0	Do not store user data bit
1	Store user data bit in bit 29 in sample buffer

VALEN[16]	Store Validity Bit
0	Do not store validity bit
1	Store validity bit 28 in sample buffer

VALID[4]	Sample Data Store
0	Sample data stored in buffer regardless of sub_frame validity bit
1	Sample data stored only when sub_frame validity bit is 0

CHAS[3]	RxStatus Register Hold Channel
0	RxStatus register hold status from channel B
1	RxStatus register hold status from channel A

RINREN[2]	Interrupt Output
0	Interrupt output is disabled
1	Interrupt output is enabled

SAMPLE[1]	Stored Data
0	No data is stored in the sample buffer
1	Data is stored in the sample buffer

RXEN[0]	Receiver Disable Enable
0	Receiver is disabled
1	Receiver is enabled

SPDIF Receiver Status Register 0xF0535808

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									COPY	EMPH			AUDIO	PRO	LOCK

The signal status register holds information about the received signal. When the receiver is disable, all bits read 0. When lock bit is 0, all other bits are set to 0.

COPY [6]	Copy Information
0	Copy inhibited
1	Copy permitted(copy bit only applicable in consumer mode)

EMPH [5:3]	Copy Information
N	Emphasis code from the channel status block. Note that the interpretation of these bits is different in consumer and professional mode.

AUDIO [2]	Signal Data
0	Signal is non-audio
1	Signal is audio

PRO [1]	Signal format
0	Signal format is consumer
1	Signal format is professional

LOCK [0]	LOCK to SPDIF Signal
0	Receiver has no valid input signal
1	Receiver is locked to SPDIF signal

SPDIF Receiver Interrupt Mask Register 0xF053580C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					DBS		CRD	CAP7	CAP6	CAP5	CAP4	CAP3	CAP2	CAP1	CAP0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved											PEB	PEA	HSBF	LSBF	LOCK

DBS [26:25]	DMA Burst Select
0	DMA 1 Burst
1	DMA 2 Burst
2	DMA 4 Burst
3	DMA 8 Burst

CRD [24]	Capture Register DMA Interrupt Mask
N	'1' for enable for capture register DMA interrupt

CAP7 [23]	Capture Register 7 Interrupt Mask
N	'1' for enable for capture register 7 interrupt

CAP6 [22]	Capture Register 6 Interrupt Mask
N	'1' for enable for capture register 6 interrupt

CAP5 [21]	Capture Register 5 Interrupt Mask
N	'1' for enable for capture register 5 interrupt

CAP4 [20]	Capture Register 4 Interrupt Mask
N	'1' for enable for capture register 4 interrupt

CAP3 [19]	Capture Register 3 Interrupt Mask
N	'1' for enable for capture register 3 interrupt

CAP2 [18]	Capture Register 2 Interrupt Mask
N	'1' for enable for capture register 2 interrupt

CAP1 [17]	Capture Register 1 Interrupt Mask
N	'1' for enable for capture register 1 interrupt

CAP0 [16]	Capture Register 0 Interrupt Mask
N	'1' for enable for capture register 0 interrupt

PEB [4]	Parity Error Channel B Interrupt Mask
N	'1' for enable for parity error channel B interrupt

PEA [3]	Parity Error Channel A Interrupt Mask
N	'1' for enable for parity error channel A interrupt

HSBF [2]	Higher Sample Buffer Full Interrupt Mask
N	'1' for enable for higher sample buffer full interrupt

LSBF [1]	Lower Sample Buffer Full Interrupt Mask
N	'1' for enable for lower sample buffer full interrupt

LOCK [0]	Change in Lock Bit Interrupt Mask
N	'1' for enable for change in lock bit interrupt

SPDIF Receiver Interrupt Status Register 0xF0535810

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							CRD	CAP7	CAP6	CAP5	CAP4	CAP3	CAP2	CAP1	CAP0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved											PEB	PEA	HSBF	LSBF	LOCK

CRD [24]	Capture Register DMA Interrupt
N	A bit in this register is set to '1' when an event occurs.

CAP7 [23]	Capture Register 7 Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

CAP6 [22]	Capture Register 6 Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

CAP5 [21]	Capture Register 5 Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

CAP4 [20]	Capture Register 4 Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

CAP3 [19]	Capture Register 3 Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

CAP2 [18]	Capture Register 2 Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

CAP1 [17]	Capture Register 1 Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

CAP0 [16]	Capture Register 0 Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

PEB [4]	Parity Error Channel B Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

PEA [3]	Parity Error Channel A Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

HSBF [2]	Higher Sample Buffer Full Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

LSBF [1]	Lower Sample Buffer Full Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

LOCK [0]	Change in Lock Bit Interrupt
N	A bit in this register is set to ‘1’ when an event occurs.

If the corresponding bit in RxIntMask is set to 1, an interrupt is generated (if enabled). Write 1 to a bit to clear the event. The interrupt signal goes inactive when all events been cleared

SPDIF Receiver Channel Status Capture [n] Register 0xF0535840 ~ 0xF053587C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BITPOS								CDATA	CHID	BITLEN					

BITPOS[15:8]	First Bit Position of Bt-Filed to be Captured
n	Valid Range 0 to 191

CDATA[7]	DATA Captured
0	User data is captured
1	Channel status data is captured

CHID[6]	Source Channel
0	Source is channel A
1	Source is channel B

BITLEN[5:0]	Bit Length
0	Capture function disable
1~32	Length of bit-field to be captured

The channel status capture register can be set up to capture a specified bit-field of the channel status frame or user data frame, and also generate an interrupt when the bit-field changes. The bit-field can be from 1 to 32 bits long. There can be up to eight sets of bit-field capture registers

SPDIF Receiver Channel Status Data [n] Register 0xF0535840~0xF053587C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA															

DATA[31:0]	Captured Channel Status
N	First bit captured is stored in bit 0

Captured channel status/user data bits

SPDIF Receiver Sample Data Descripton **0xF0535A00 ~ 0xF0535A1C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PARITY	CHSTAT	USRDAT	VALID	BLKS	Reserved			DATAH							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATAL															

PARITY [31]	Parity Bit
N	If enable, otherwise 0

CHSTAT [30]	Channel Status Bit
N	If enable, otherwise 0

USRDAT [29]	User Data Bit
N	If enable, otherwise 0

VALID [28]	Validity Bit
N	If enable, otherwise 0

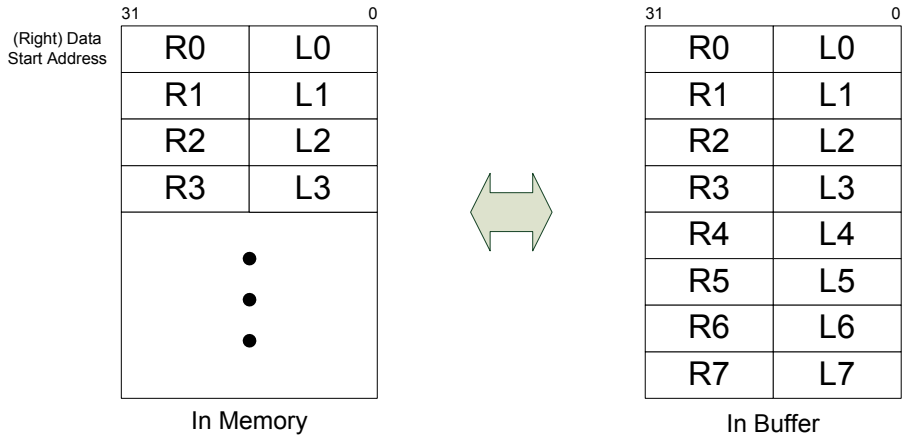
BLKSTART [27]	Start of Sample Block Bit
N	If enable, otherwise 0

DATAH [23:16]	DATA High
N	A udio Data(if > 16bit Resolution) - Unused bit are 0

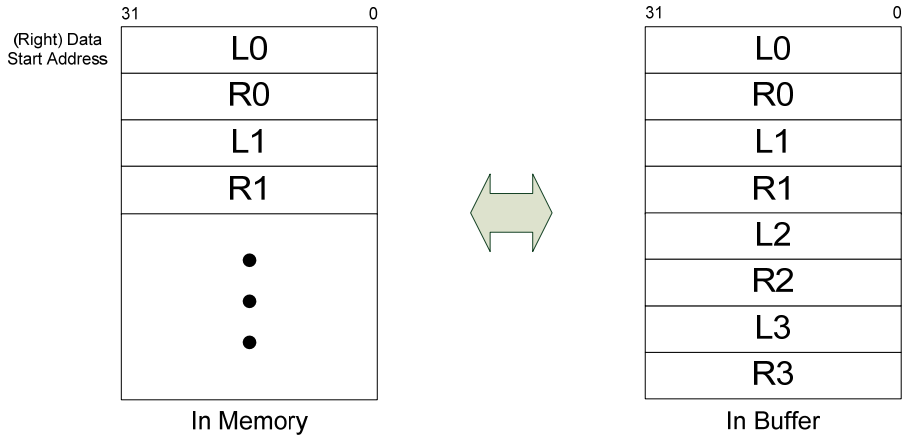
DATAL [15:0]	DATA Low
N	Audio Data - Bit 0 is LSB

Format of data words in receive sample buffer

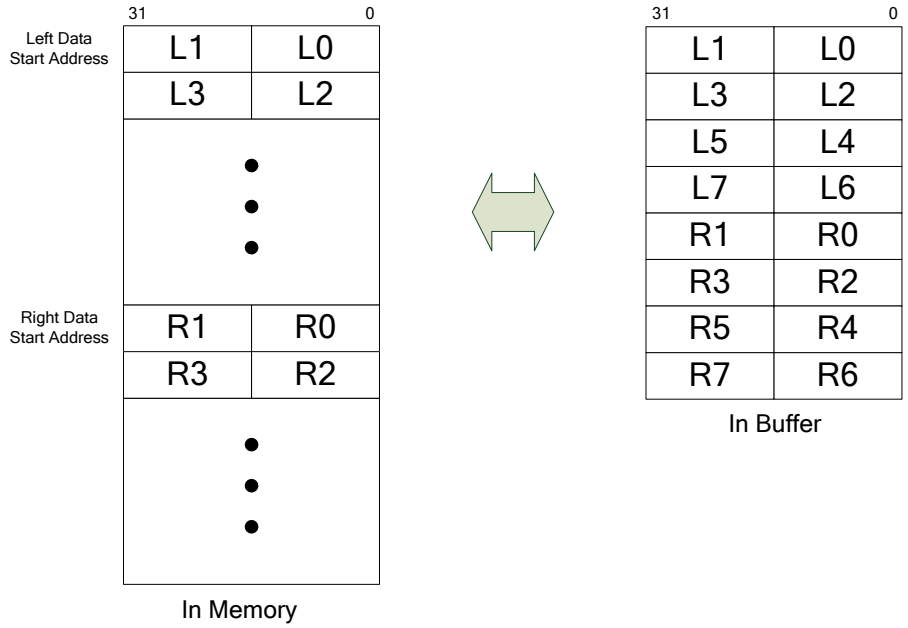
11.2.5 Audio Data Format



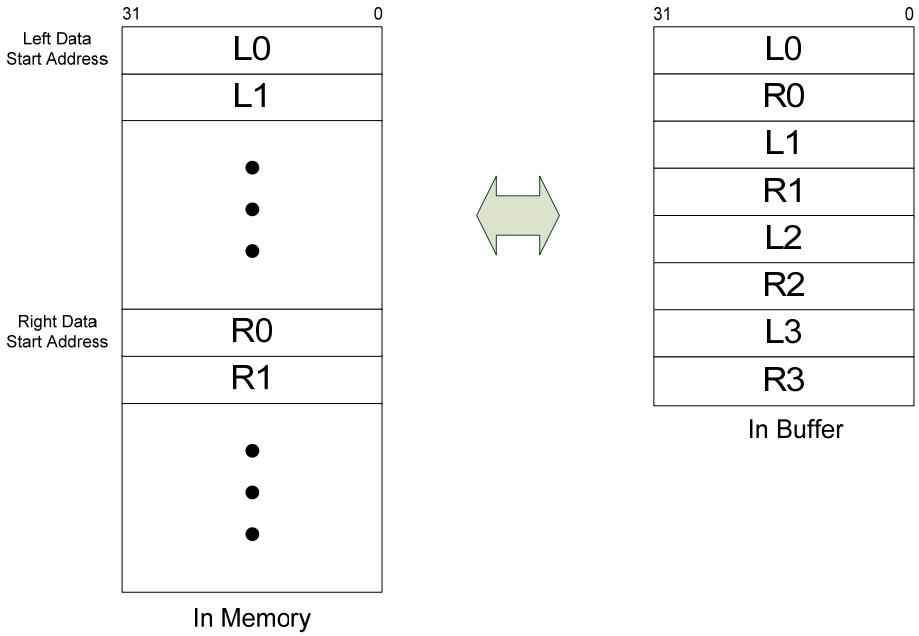
LRMode=0, Data_Width=16 bit



LRMode=0, Data_Width=24 bit



LRMode=1, Data_Width=16 bit



LRMode=1, Data_Width=24 bit

Figure 11.10 Data Format between Memory and Buffer

12 DAI & CD Interface

12.1 DAI (Digital Audio Interface)

The block diagram of DAI is shown in Figure 12.1.

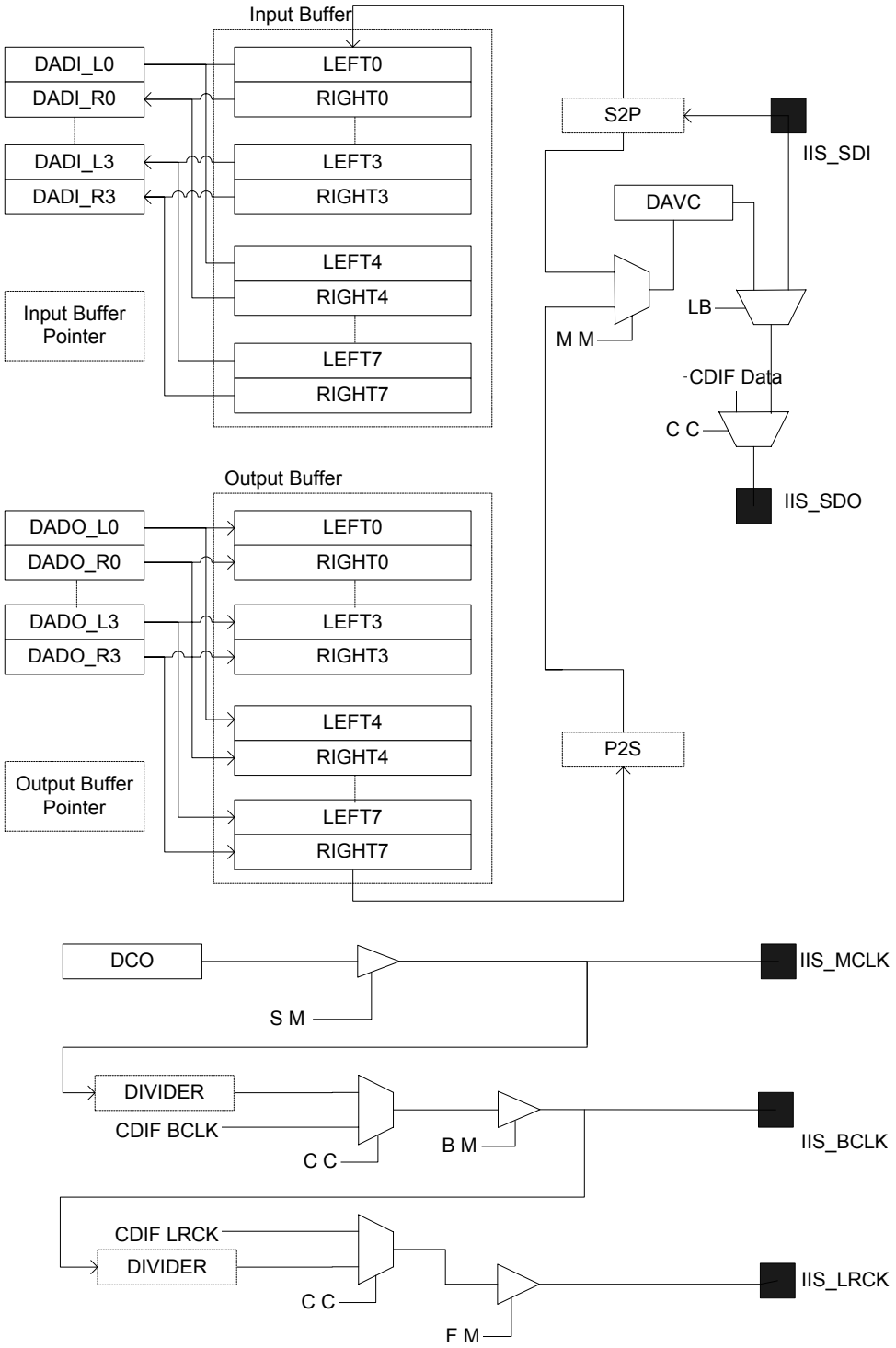


Figure 12.1 DAI Block Diagram

The TCC8900 provides digital audio interface that complies with IIS (Inter-IC Sound). The DAI has five input/output pins for IIS interface; MCLK, BCLK, LRCK, DAI, DAO.

The MCLK is the system clock pin that is used for CODEC system clock. In master mode, the MCLK can be generated from clock generator known as a DCLK, or fed from the outside of the TCC8900 in slave mode. The DAI can process 256fs, 384fs

and 512fs as a system clock. 256fs means that the system clock has 256 times of sampling frequency (fs).

The BCLK is the serial bit clock for IIS data exchange. The DAI can generate 64fs, 48fs and 32fs by dividing a system clock. The polarity of BCLK can be programmed. That is, the serial bit can be stable either at the rising edge of BCLK or falling edge of BCLK.

The LRCK is the frame clock for the stereo audio channel Left and Right. The frequency of LRCK is known as the “fs” – sampling frequency. Generally, for audio application – such as MP3 Player, CD player, the fs can be set to 8kHz, 16kHz, 11.05kHz, 24kHz, 32kHz, 44.1kHz and 48kHz. For supporting the wide range of sampling frequency in audio application, the DCO function is very useful to generate a system clock. Refer the chapter of clock generator for detail information.

All three clocks (MCLK, BCLK, LRCK) are selectable as master or slave.

The DAI, DAO are the serial data input output pins respectively.

The DAI has two 8-word input/output buffers. It has a banked buffer structure so that one side of buffer is receiving/transmitting data while the other side of that can be read/written through the DADI_XX/DADO_XX registers. The maximum data word size is 24 bit. Data is justified to MSB of 32bits and zeros are padded to LSB.

There are 2 types of interrupt from IIS; transmit done interrupt, receive done interrupt. The transmit-done interrupt is generated when the 8 words are transferred successfully in the output buffer. At this interrupt, user should fill another 8 more words into the other part of the output buffer in the interrupt service routine (ISR). In this ISR routine, 8 consecutive stores of word data to the DADO registers are needed. The receive-done interrupt is generated when the 8 words are received successfully in the input buffer. At this interrupt, user should read 8 received words from the input buffer using 8 consecutive load instructions from the DADI registers.

Table 12.1 DAI Register Map (Base Address = 0xF0537000)

Name	Address	Type	Reset	Description
DADI_L0	0x00	R	-	Digital Audio Left Input Register 0
DADI_R0	0x04	R	-	Digital Audio Right Input Register 0
DADI_L1	0x08	R	-	Digital Audio Left Input Register 1
DADI_R1	0x0C	R	-	Digital Audio Right Input Register 1
DADI_L2	0x10	R	-	Digital Audio Left Input Register 2
DADI_R2	0x14	R	-	Digital Audio Right Input Register 2
DADI_L3	0x18	R	-	Digital Audio Left Input Register 3
DADI_R3	0x1C	R	-	Digital Audio Right Input Register 3
DADO_L0	0x20	R/W	-	Digital Audio Left Output Register 0
DADO_R0	0x24	R/W	-	Digital Audio Right Output Register 0
DADO_L1	0x28	R/W	-	Digital Audio Left Output Register 1
DADO_R1	0x2C	R/W	-	Digital Audio Right Output Register 1
DADO_L2	0x30	R/W	-	Digital Audio Left Output Register 2
DADO_R2	0x34	R/W	-	Digital Audio Right Output Register 2
DADO_L3	0x38	R/W	-	Digital Audio Left Output Register 3
DADO_R3	0x3C	R/W	-	Digital Audio Right Output Register 3
DAMR	0x40	R/W	0x00000000	Digital Audio Mode Register
DAVC	0x44	R/W	0x0000	Digital Audio Volume Control Register

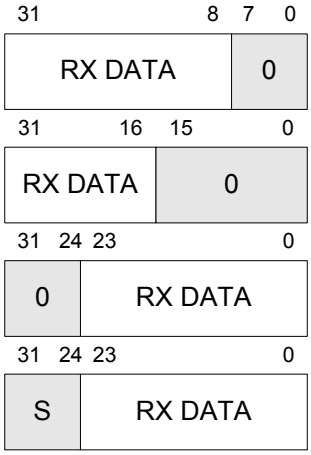
Digital Audio Mode Register (DAMR) 0xF0537040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MS	Reserved								RXE	RXS<2:0>		TXS<1:0>		0	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EN	TE	RE	MD	SM	BM	FM	CC	BD<1:0>		FD<1:0>		BP	CM	MM	LB

MS [31]	Master Clock Selection Register
0	Internal BCLK is used. (Master Mode)
1	External BCLK is used. (Slave Mode)

RXE [22]	DAI RX Data Sign Extension
0	Disable (zero extension)
1	Enable (sign bit extension)

RXS [21:20]	DAI RX Shift
00	Bit-pack MSB and 24bit mode.
01	Bit-pack MSB and 16bit mode.
10 or 11	Bit-pack LSB and 24bit mode.



RXS = 0 **RXE = Don't care**

RXS = 1 **RXE = Don't care**

RXS = 2 or 3 **RXE = 0**

RXS = 2 or 3 **RXE = 1**

TXS [19:18]	DAI TX Shift
0x	Bit-pack MSB mode.
10	Bit-pack LSB and 24bit mode.
11	Bit-pack LSB and 16bit mode.

EN [15]	DAI Master Enable
0	Disable DAI module
1	Enable DAI module

TE [14]	DAI Transmitter Enable
0	Disable DAI transmitter
1	Enable DAI transmitter

RE [13]	DAI Receiver Enable
0	Disable DAI receiver
1	Enable DAI receiver

MD [12]	DAI Bus Mode
0	Set DAI bus as IIS bus mode
1	Set DAI bus as MSB justified mode

SM [11]	DAI System Clock Master Select
0	Set that DAI system clock is come from external pin
1	Set that DAI system clock is generated by the clock generator block

The DAI system clock in clock generator is known as DCLK. Its frequency can be determined by PCK_DAI from the CKC.

BM [10]	DAI Bit Clock Master Select
0	Set that DAI bit clock is from external pin
1	Set that DAI bit clock is generated by dividing DAI system clock

FM [9]	DAI Frame Clock Master Select
0	Set that DAI frame clock is come from external pin
1	Set that DAI frame clock is generated by dividing DAI bit clock

CC [8]	CDIF Clock Select
0	Disable CDIF Clock master mode
1	Enable CDIF Clock master mode

BD [7:6]	DAI Bit Clock Divider select
00	Select Div 4 (256fs->64fs)
01	Select Div 6 (384fs->64fs)
10	Select Div 8 (512fs->64fs, 384fs->48fs , 256fs->32fs)
11	Select Div16 (512fs->32fs)

FD [5:4]	DAI Frame Clock Divider select
00	Select Div 32 (32fs->fs)
01	Select Div 48 (48fs->fs)
10	Select Div 64 (64fs->fs)

The combination of BD & FD determines that the ratio between main system clock and the sampling frequency. The multiplication between the division factor of BD and FD must be equal to this ratio.

BP [3]	DAI Bit Clock Polarity
0	Set that data is captured at positive edge of bit clock
1	Set that data is captured at negative edge of bit clock

CM [2]	CDIF Monitor Mode
0	Disable CDIF monitor mode
1	Enable CDIF monitor mode. Data bypass from CDIF

MM [1]	DAI Monitor Mode
0	Disable DAI monitor mode
1	Enable DAI monitor mode. Transmitter should be enabled. (TE = 1)

LB [0]	DAI Loop-back Mode
0	Disable DAI Loop back mode
1	Enable DAI Loop back mode

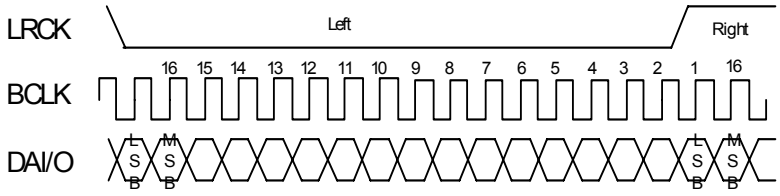
Digital Audio Volume Control Register (DAVC) 0xF0537044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											VC<4:0>				

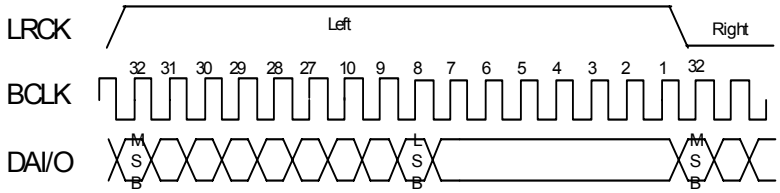
The volume of audio output can be manipulated by this register. It has –6dB unit so the output volume can be set from 0 dB to –90 dB as the following table.

VC [4:0]	DAI Volume control
00000	0dB
00001	-6dB
00010	-12dB
00011	-18dB
00100	-24dB
00101	-30dB
00110	-36dB
00111	-42dB
01000	-48dB
01001	-54dB
01010	-60dB
01011	-66dB
01100	-72dB

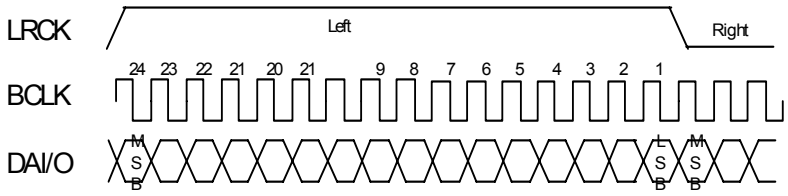
01101	-78dB
01110	-84dB
01111	-90dB
10000	-96dB



MD=0 (IIS mode), BP=0, BCLK = 32fs



MD=1(MSB justified mode), BP=0, BCLK=64fs



MD=1(MSB justified mode), BP=1, BCLK=48fs

Figure 12.2 DAI Bus Timing Diagram

12.2 CDIF (CD Media Interface)

The block diagram of CDIF is illustrated in Figure 12.3.

The TCC8900 provides CD-ROM interface for feasible implementation of CD-ROM application such as CD-MP3 player. The CDIF supports the industry standard IIS format and the LSB justified format used as the most popular format for CD-ROM interface by Sony and Samsung.

The CDIF has three pins for interface; CBCLK, CLRCK, CDAI. The CBCLK is the bit clock input pin of which frequency can be programmed by CICR for selection of 48fs and 32fs. The CLRCK is the frame clock input pin that indicates the channel of CD stereo digital audio data. The CDAI is the input data pin.

The CDIF has nine registers; CDDI_0 to CDDI_3 and CICR. The CDDI_0 to the CDDI_3 are the banked read only registers for access of data input buffer. The data input buffer is composed of sixteen 32 bit wide registers of which upper 16 bit is left channel data and lower is right channel data.

The CDIF receive the serial data from CDAI pin and store the data into the buffer through the serial to parallel register. Whenever the half of buffer is filled, the receive interrupt is generated. Only the half of input buffer can be accessible through the CDDI_0 to the CDDI_3.

Table 12.2 CDIF Register Map (Base Address = 0xF0537000)

Name	Address	Type	Reset	Description
CDDI_0	0x80	R		CD Digital Audio Input Register 0
CDDI_1	0x84	R		CD Digital Audio Input Register 1
CDDI_2	0x88	R		CD Digital Audio Input Register 2
CDDI_3	0x8C	R		CD Digital Audio Input Register 3
CICR	0x90	R/W	0x0000	CD Interface Control Register

CD Data Input (CDDI0) **0xF0537080**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Left Channel Data															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Right Channel Data															

CD Data Input (CDDI1) **0xF0537084**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Left Channel Data															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Right Channel Data															

CD Data Input (CDDI2) **0xF0537088**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Left Channel Data															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Right Channel Data															

CD Data Input (CDDI3) **0xF053708C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Left Channel Data															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Right Channel Data															

CD Interface Control Register (CICR) **0xF0537090**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EN	Reserved			BS		MD	BP

EN [7]	CDIF Enable
0	Disable CDIF
1	Enable CDIF

BS [3:2]	CDIF Bit Clock select
00	64fs
01	32fs
10	48fs

MD [1]	Interface Mode select
0	Select IIS format
1	Select LSB justified format

BP [0]	CDIF Bit Clock Polarity
0	Set that data is captured at positive edge of bit clock
1	Set that data is captured at negative edge of bit clock

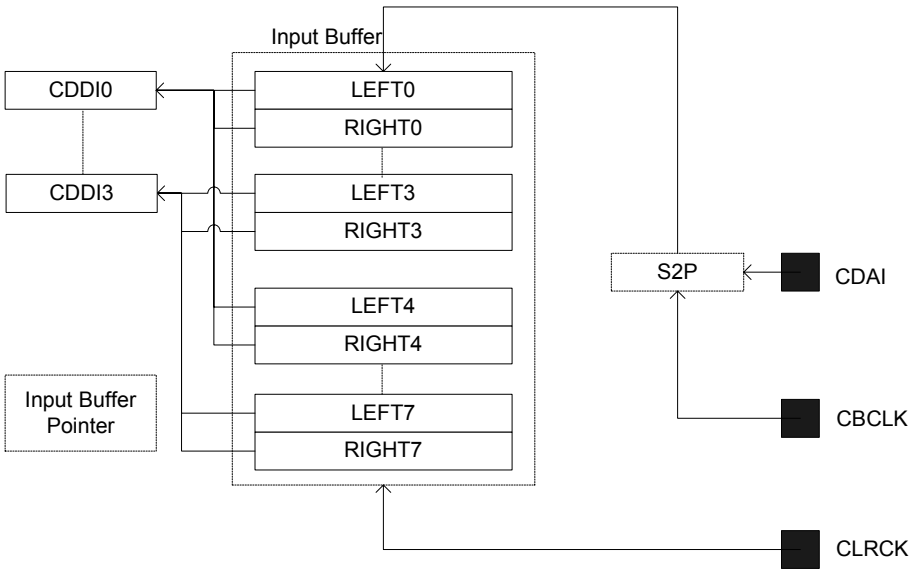
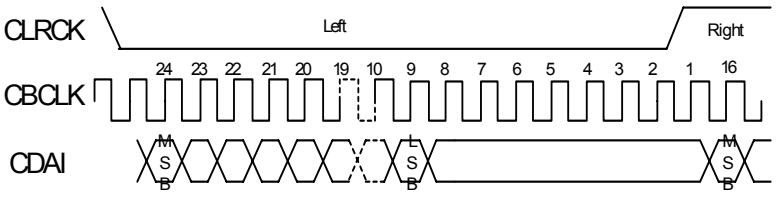
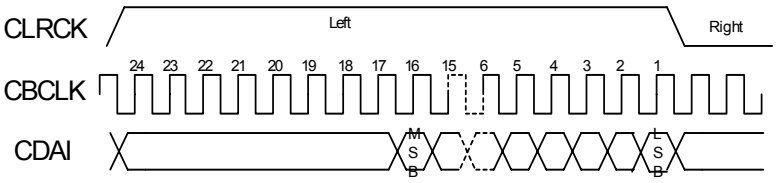


Figure 12.3 CDIF Block Diagram



MD=0 (IIS mode), BP=0, CBCLK=48fs



MD=1(LSB justified mode), BP=0, CBCLK=48fs

Figure 12.4 CDIF Bus Timing Diagram

13 SPDIF Transmitter

13.1 Overview

The SPDIF (or AES/EBU, IEC950 standards) is a point-to-point protocol for serial transmission of digital audio through a single transmission line. The transmission medium can be either electrical or optical (e.g. TosLink). It provides two channels for audio data, a method for communicating control information, and some error detection capabilities. The control information is transmitted as one bit per sample and accumulates in a block structure. The data is by-phase encoded, which enables the receiver to extract a clock from the data. Coding violations, defined as preambles, are used to identify sample and block boundaries.

The SPDIF interfaces are found on most CD/DVD players, audio equipment and computer sound cards.

The transmitter architecture is shown below.

The sub-frame assembler creates 32bit data words from sample data, register settings and optionally channel status/user data buffers. A parity bit is added in each sub-frame. The frame assembler adds preambles to create a frame of two sub-frames. 192 frames add up to a block. The block structure is bi-phase encoded before transmitting.

The size of the sample buffer is 4words. The sample buffer is addressed by setting the most significant address (0x00000200) bit to '1'. The sample buffer is divided in two equal parts, lower and upper, and the user will be notified when either is emptied of audio data.

The channel status can be generated from a dedicated 192bit buffer. Two interrupts can be generated when the transmitter reads from the buffer, one in the middle and one at the end. The user data buffer operates in an identical way.

Table 13.1 SPDIF Register Map (Base Address = 0xF0538000)

Name	Address	Type	Reset	Description
TxVersion	0x00	R		Version Register
TxConfig	0x04	R/W		Configuration Register
TxChStat	0x08	R/W		Channel Status Control Register
TxIntMask	0x0C	R/W		Interrupt Mask Register
TxIntStat	0x10	R/W		Interrupt Status Register
UserData	0x80~0xDC	W	-	User Data Buffer
ChStatus	0x100~0x15C	W	-	Channel Status Buffer
TxBuffer	0x200~0x210	W	-	Transmit Data Buffer
DMACFG	0x400	R/W	-	Additional Configuration for DMA

13.2 Register Descriptions

SPDIF Transmitter Version Register **0xF0538000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0			CSB	UDB	AW						DW	VER			

CSB [12]	Channel Status Buffer Available Bit
0	Channel status buffer is not available
1	Channel status buffer is available

UDB[11]	User Data Buffer Available Bit
0	User data buffer is not available
1	User data buffer is available

AW[10:5]	Value of Address Width
n	Value of Address Width

DW[4]	Value of Data Width
n	Value of Data Width

VER[3:0]	Design Version
n	Version Number

SPDIF Transmit Configuration Register **0xF0538004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								MODE				0			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RATIO								UDATEN		CHSTEN		0	IEN	TXD	TXEN

MODE[23:20]	Sample Format Mode Indicator
n	16 + n Bits Transmit Sample Format
9 ~ 15	Reserved

RATIO[15:8]	Clock Divider Ratio
n	Clock divider for the transmit frequency. The SPDIF clock is divided by a factor of (RATIO + 1) to generate the serial transmit clock.

UDATEN[7:6]	User Data Enable Bits
0	User data A & B set to 0.
1	User data A & B generated from UserData bit 7-0
2	User data A generated from UserData bit 7-0, B generated from UserData bit 15-8
3	Reserved

CHSTEN[5:4]	Channel Status Enable Bits
0	Channel status A & B generated from TxChStat
1	Channel status A & B generated from ChStat bit 7-0
2	Channel status A generated from ChStat bit 7-0, B generated from ChStat bit 15-8
3	Reserved

IEN[2]	Interrupt Output Enable Bit
n	'1' for enabling the interrupt output.

TXD[1]	Data Valid Bit
n	'1' for data being valid.

TXEN[0]	Transmitter Enable Bit
n	'1' for enabling the transmission.

SPDIF Transmit Channel Status Control Register 0xF0538008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0					SCS	SUD	SV	FREQ		0		GSTS	PRE	CPY	AU

SCS[10]	Store Channel Status bit
0	No store
1	Store

SUD[9]	Store User Data bit
0	No store
1	Store

SV[8]	Store Validity bit
0	No store
1	Store

FREQ[7:6]	Sampling Frequency
0	44.1kHz
1	48kHz
2	32kHz
3	Sample Rate Converter

GSTS[3]	Status Generation
0	No indication
1	Original/Commercially Pre-recorded data

PRE[2]	Pre-emphasis
0	None
1	50/15us

CPY[1]	Copyright
0	Copy inhibited
1	Copy permitted

AU[0]	Data Format
0	Audio Format
1	Data Format

SPDIF Transmit Interrupt Mask Register **0xF053800C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0											HCSB	LCSB	HSB	LSB	0

BIT	NAME	DESCRIPTION
4	HCSB	Higher Channel Status/User Data Buffer Empty '1' for enable for higher channel status/user data buffer empty interrupt
3	LCSB	Lower Channel Status/User Data Buffer Empty '1' for enable for lower channel status/user data buffer empty interrupt
2	HSB	Higher Data Buffer Empty '1' for enable for higher data buffer empty interrupt
1	LSB	Lower Data Buffer Empty '1' for enable for lower data buffer empty interrupt

SPDIF Transmit Interrupt Status Register **0xF0538010**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0											HCSB	LCSB	HSB	LSB	0

Bit	Name	Description
4	HCSB[4]	Higher Channel Status/User Data Buffer Empty '1' for higher channel status/user data buffer empty interrupt activated
3	LCSB[3]	Lower Channel Status/User Data Buffer Empty '1' for lower channel status/user data buffer empty interrupt activated
2	HSB[2]	Higher Data Buffer Empty '1' for higher data buffer empty interrupt activated
1	LSB[1]	Lower Data Buffer Empty '1' for lower data buffer empty interrupt activated

SPDIF Transmit User Data Buffer Register 0xF0538080 ~ 0xF05380DC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHBUD								CHAUD							

The user data is 24 bytes per sample block. Bit 0 of byte 0 is transmitted first. Bit 7 of byte 23 is transmitted last.

Bit	Name	Description
15:8	CHBUD[15:8]	User Data for Channel B 8 Bits User Data for Channel B
7:0	CHAUD[7:0]	User Data for Channel A 8 Bits User Data for Channel A

SPDIF Transmit Channel Status Buffer Register 0xF0538100 ~ 0xF053815C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHBCS								CHACS							

The channel status data is 24 bytes per sample block. Bit 0 of byte 0 is transmitted first. Bit 7 of byte 23 is transmitted last.

bit	name	description
15:8	CHBCS[15:8]	Channel Status for Channel B 8 Bits Channel Status for Channel B
7:0	CHACS[7:0]	Channel Status for Channel A 8 Bits Channel Status for Channel A

SPDIF Transmit Sample Data Buffer Register 0xF0538200 ~ 0xF05383FC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								DATH							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATL															

The format of data words in transmit sample buffer. Channel A is transmitted first, and must be stored on even addresses, while channel B is stored on odd addresses in the sample buffer.

BIT	name	description
23:16	DATH	Upper 8 Bits for Sample Buffer Data
		Audio data if > 16 bits resolution. Unused bits are 0
15:0	DATL	Lower 16 Bits for Sample Buffer Data
		Audio data (16 bits mode). Bits 0 is LSB

DMA Configuration (DMACFG) **0xF0538400**

Field	Name	RW	Reset	Description
31 ~ 21	Reserved	R	0	Reserved
20 ~ 16	VCNT	R	X	FIFO Valid Entry Count
15 ~ 12	Reserved	R	0	Reserved
11	DRQEN1	RW	0	DMA Request Enable for User Data Buffer
10	DRQEN0	RW	0	DMA Request Enable for Sample Data Buffer
9	AMODE1	RW	0	Sample Data Buffer Address Mode. Buffer would be written with sequence below. (0, 2, 4, 6, 1, 3, 5, 7)
8	AMODE0	RW	0	0 : Ignore Address (FIFO Mode) 1 : Enable Address (16 Entries can be written with address)
7	FIFOCLR	RW	0	Clear FIFO. Should be written with “0” for normal operation
6 ~ 4	Reserved	R	0	Reserved
3 ~ 0	FIFOTH	RW	X	FIFO Threshold for DMA Request DMA request will be asserted if VCNT <= FIFOTH

13.3 Operation & Timing Diagram

13.3.1.1 Resetting

Except for the system reset, the transmitter can be disabled by clearing the TXEN bit in the TxConfig register.

13.3.1.2 Selecting Transmit Data Rate

The data rate of the SPDIF signal is a function of the SPDIF clock from CKC and the RATIO bits in the TxConfig register. The bit rate is 64 times of the sampling frequency – each sample is encoded as 32 bits and there are two channels. Sample frequency is given by the following equation:

$S_{FREQ} = C_{SPDIF} / (128 \times (RATIO + 1)))$;

Example : Data rate is 48kHz and C_{SPDIF} is 12.288MHz. The RATIO bits must then be set to 1. Output data rate is 3.072Mbps.

13.3.1.3 Selecting Data Format

Any sample resolution from 16 to 24 bit can be transmitted. Data format is selected by the MODE bits in the TxConfig register.

13.3.1.4 Setting up Channel Status Bits

If output format is standard consumer audio, set CHSTEN to 0 in TxConfig, and set TxChStat to desired format. Otherwise set up the ChStatus buffer with desired channel status data (192bits). If the channel status bits do not change from blocks to blocks, it is only necessary to program the buffer once. Otherwise the HCSBF/LCSBF bits in TxIntMask must be set, and the buffer will need to be updated when every half block is transmitted.

13.3.1.5 Setting up User Data Bits

User data bits are normally set to zero, but if required user data can be transmitted using the UserData buffer. If the user data bits do not change from block to block, it is only necessary to program the buffer once. Otherwise the HCSBF/LCSBF bits in TxIntMask must be set, and the buffer will need to be updated every block.

13.3.1.6 Preparing Sample Buffer

Before the TXDATA bit in TxConfig is set, fill up the complete sample buffer with audio data. The transmitter will generate an interrupt when lower half or upper half of sample buffer is emptied.

13.3.1.7 Start Transmission

Transmission of SPDIF signal starts when the TXEN bit in TxConfig is set. If TXDATA bit is not set, the transmitted data will be all zeros with the sub-frame validity bit set.

Once the TXDATA bit is set, audio data from the sample buffer will be transmitted and the validity bit is cleared.

14 USB 1.1 Host Controller & Transceiver

14.1 Overview

The USB host controller complies with OHCI Rev 1.0. Refer to the specification of OHCI (Open Host Controller Interface) Rev 1.0 for more detailed information.

- Open HCI Rev 1.0 compatible
- USB Rev 1.1 compatible
- RootHub is user configurable
  (e.g., Number of DownStreamPorts PowerSwitching Options etc)
- Support for both LowSpeed and FullSpeed USB Devices
- No Bi-Directional or Tri-State Buses
- No level sensitive Latches
- Very simple Application Bus Interface
- Support of SMI(System Management Interrupt)
- Hooks for Legacy Device Support
- Down stream port 0 interfaces with USB1.1 transceiver
- Down stream port 1 interfaces with USB2.0 OTG PHY

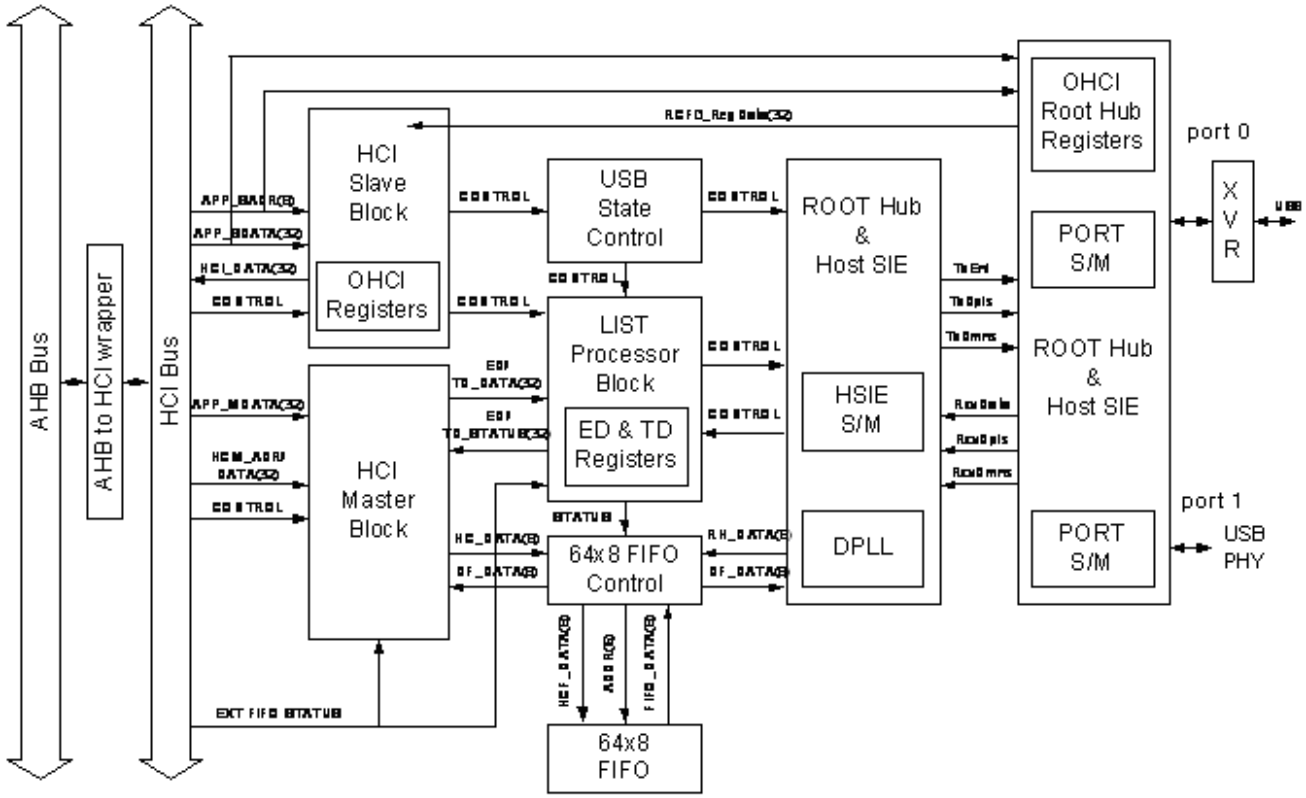


Figure 14.1 USB Host Controller Block Diagram

The analog transceiver consists of a LS(Low Speed)/FS(Full Speed) Driver, LS/FS Differential Receiver and SE(Single-Ended) Receivers required by USB2.0 Specification; Data-line pulsing blocks required by OTG supplement. It converts logic-level signals to USB signals, and USB signals to logic-level signals.

- 65nm Low Power(L6LP) CMOS technology
- 3.3V ±10% analog power supply
- 1.2V ±10% digital power supply
- Complies with USB specification rev 2.0
- Full-speed (12Mbit/s) and low-speed (1.5Mbit/s) data rates
- Supports OTG supplement feature:
  SRP request by "data-line pulsing" method
- Two single-ended receivers with hysteresis
- Three-state outputs

- Full industrial operating temperature range: –40°C ~ 85°C

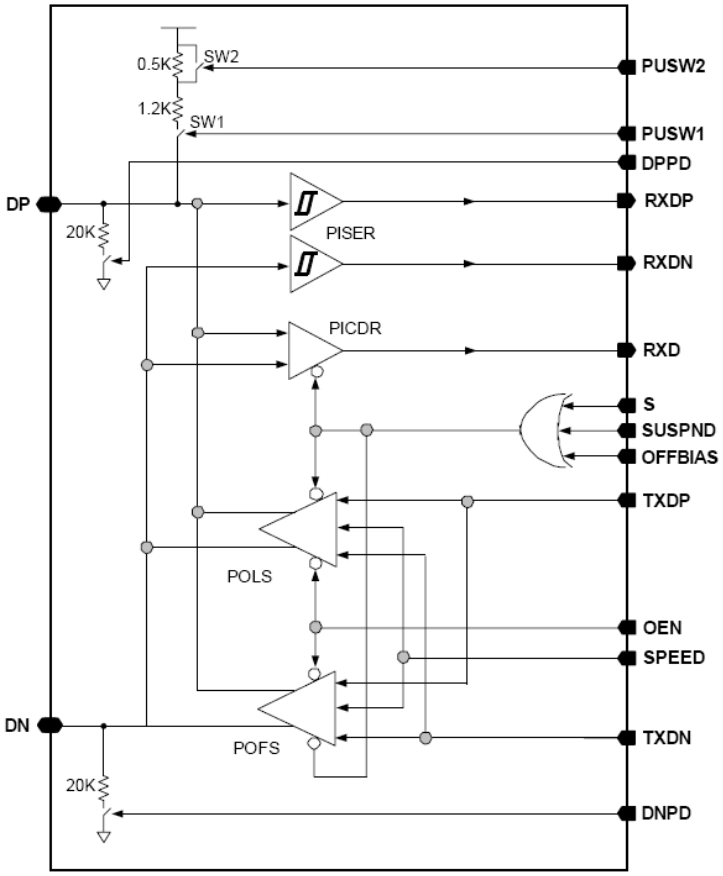


Figure 14.2 USB 1.1 Transceiver Block Diagram

14.2 Register Description for USB 1.1 Host Controller & Transceiver

The following table in next page describes the address mapping of OHCI Rev1.0 registers.

Table 14.1 USB Host Register Map (Base Address = 0xF0500000)

Name	Address	Type	Reset	Description
HcRevision	0x00	R	0x00000010	Control and status registers
HcControl	0x04	R/W	0x00000000	
HcCommandStatus	0x08	R	0x00000000	
HcInterruptStatus	0x0C	R	0x00000000	
HcInterruptEnable	0x10	R/W	0x00000000	
HcInterruptDisable	0x14	W	0x00000000	
HcHCCA	0x18	R/W	0x00000000	Memory pointer registers
HcPeriodCurrentED	0x1C	R	0x00000000	
HcControlHeadED	0x20	R/W	0x00000000	
HcControlCurrentED	0x24	R/W	0x00000000	
HcBulkHeadED	0x28	R/W	0x00000000	
HcBulkCurrentED	0x2C	R/W	0x00000000	
HcDoneHead	0x30	R	0x00000000	
HcRmInterval	0x34	R/W	0x00002EDF	Frame counter registers
HcFmRemaining	0x38	R/W	0x00000000	
HcFmNumber	0x3C	R/W	0x00000000	
HcPeriodStart	0x40	R/W	0x00000000	
HcLSThreshold	0x44	R/W	0x00000628	
HcRhDescriptorA	0x48	R/W	0x02001202	Root hub registers
HcRhDescriptorB	0x4C	R/W	0x00000000	
HcRhStatus	0x50	R/W	0x00000000	
HcRhPortStatus1	0x54	R/W	0x00000100	
HcRhPortStatus2	0x58	R/W	0x00000100	

Table 14.2 USB 1.1 Host Configuration Register(Base Address = 0xF05F5000)

Name	Address	Type	Reset	Description
USB11H	0x4	R/W	0x00000010	USB1.1 Host Configuration register

USB 1.1 Host Configuration Register (USB11H) 0xF05F5004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-						ID	SLEEP	OFFBIAS	PUSW2	PUSW1	DNPD0	DPPD0	CNT	OVRC	

Field	Name	RW	Reset	Description
1 ~ 0	OVRC [Port1:Port0]	RW	0x0	Over Current Indication from Application When asserted by Application the corresponding Port will enter DISCONNECTstate if PowerSwitching is not implemented or PoweredOff state if PowerSwitching is implemented irrespective of its present state. In either case, this signal should be cleared before Port can be reused. Besides if PowerSwitching is implemented, Power needs to be turned on (by writing to OHCI Registers) before Port detects connect event of any peripheral downstream. Also when this signal is asserted, HC sets either HcRhPortStatus.POCI(if PowerSwitchingMode is Global) if OverCurrent Protection is supported (HcRhDescriptorA.NOCP is cleared). If any of the above two bits are set(OCI or POCI) and PowerSwitching is implemented(HcRhDescriptorA.NPS is cleared) then HcRhPortStatus.PPS(PortPowerStatus) is cleared which causes the Port enter PoweredOff state. Another condition where this signal is used is GangedModePower Switching. Ports are said to be in a GANG if PowerSwitchingMode is PerPort(HcRhDescriptorA.PSM is set) and the corresponding HcRhDescriptorB.PPCM bit is cleared. In this case if OverCurrentCondition exists on any Port, PPS bits of all the Ganged Ports are cleared provided PowerSwitching is implemented. But OverCurrentCondition(OCI) bit set for only Ports whose PRT_OvrCurrent signal is set. This signal can be asynchronous as it is double synchronized internally to CI.
2	CNT	RW	0x0	Counter value Select Selecting the Counter value for Simulation or Real time for 1ms. This is the select signal for the counter that generates 1 ms clock pulses based on 12Mhz clock. In simulation this can be tied to “1”, so that the counter scales down the 1-ms time. This signal is introduced to cut-down the simulation time specially when sending PortReset and PortResume. When it is tied to ‘1’ in simulation, 1ms duration (12000 clocks of Clk12) is scaled down to 7 clocks of Clk12. Thus when HC supposed to send PortReset of 10 ms, it only drive for 10x7=70 clocks. Scaled down time is only used for PortReset, Port Resume and 5 ms time measured when RootHub is suspended before recognizing any upstream RmtWkp event.
3	DPPD0	RW	0x0	Pull down(DP) control; DPPD=1, DP pull down enable DPPD=0, DP pull down disable
4	DNPD0	RW	0x0	Pull down(DN) control; DNPD=1, DN pull down enable DNPD=0, DN pull down disable
5	PUSW1	RW	0x0	Pull up control of SW1; PUSW1=1, pull up SW1 enable PUSW1=0, pull up SW1 disable
6	PUSW2	RW	0x0	Pull up control of SW2; PUSW2=1, pull up SW2 enable PUSW2=0, pull up SW2 disable

7	OFFBIAS	RW	0x0	BIAS circuit standby mode control, OFFBIAS=1, bias circuits in picdr&potls are off OFFBIAS=0, bias circuits in picdr&potls are on
8	SLEEP	RW	0x0	Sleep mode control, SLEEP=1, sleep mode SLEEP=0, normal mode
9	ID	RW	0x0	Decide the value of ID[17] of USBPHY ID=0, host mode ID=1, device mode
31 ~ 10	-	-	-	Undefined

[17] This field is effective whether the owner of USB PHY is USB1.1 Host or USB2.0 OTG.

15 USB 2.0 OTG Controller

15.1 Overview

The TCC8900 supports Dual-Role Device (DRD) controller, which supports both device and host functions and is fully compliant with the On-The-Go Supplement to the USB 2.0 Specification, Revision 1.0a.

It can also be configured as a Host-only or Device-only controller, fully compliant with the USB 2.0 Specification. The USB 2.0 configurations support high-speed (HS, 480-Mbps), full-speed (FS, 12-Mbps), and low-speed (LS, 1.5-Mbps) transfers. Additionally, it can be configured as a USB 1.1 full-speed/low-speed DRD.

The main features of TCC8900 USB 2.0 OTG controller are as follows.

[GENERAL FEATURES]

- Supports Slave, External DMA Controller Interface
- Includes USB power management features
- Includes power-saving features (clock gating, two power rails for advanced power management)
- Supports packet-based, Dynamic FIFO memory allocation for endpoints for small FIFOs and flexible, efficient use of RAM
- Uses single-port RAM
- Provides support to change an endpoint's FIFO memory size during transfers
- Supports endpoint FIFO sizes that are not powers of 2, to allow the use of contiguous memory locations
- Supports the Keep-Alive in Low-Speed mode and SOFs in High/Full-Speed modes
- Optional support for Transmit and Receive thresholding in DMA mode when dedicated Tx FIFO is selected in Device mode. Thresholding and threshold length selectable through global registers. For supporting thresholding, the AHB must be run at 60 MHz or higher.

[SOFTWARE FEATURES]

- Software handles USB commands (SETUP transactions are detected and their command payloads are forwarded to the application for decoding).
- Software handles USB errors.

[APPLICATION FEATURES]

- Interfaces for the application via the AHB:
- − AHB Slave interface for accessing Control and Status Registers (CSRs), the Data FIFO, and queues
- − Optional AHB Master interface for Data FIFO access when Internal DMA is enabled
- Supports all AHB burst types in AHB Slave interface
- Software-selectable AHB burst type on AHB Master interface
- Takes care of the 1KB boundary breakup.
- Optional support for a dedicated transmit FIFO for each of the device IN endpoints in Slave and DMA modes. Each FIFO can hold multiple packets.

[USB 2.0 SUPPORTED FEATURES]

- Complies with the On-The-Go Supplement to the USB 2.0 Specification (Revision 1.0a)
- Operates in High-Speed (HS, 480-Mbps), Full-Speed (FS, 12-Mbps) and Low-Speed (LS, 1.5-Mbps) modes
  Caution : In host role controller, Low-Speed device connection through Full-Speed Hub is not supported.
- Supports the UTMI+ Level 3 interface (Revision 1.0, February 25th, 2004). 8-, 16-, and 8/16-bit data buses are supported.
- Supports Session Request Protocol (SRP)
- Supports Host Negotiation Protocol (HNP)
- Supports up to 16 bidirectional endpoints, including control endpoint 0

- Supports up to 16 host channels. In Host mode, when the number of device endpoints to be supported is more than the number of host channels, software can reprogram the channels to support up to 127 devices, each having 32 endpoints (IN + OUT), for a maximum of 4, 064 endpoints.
- Supports a generic root hub
- Includes automatic ping capabilities

[POWER FEATURES]

- PHY clock gating support during USB Suspend mode and Session-Off mode
- AHB clock gating support during USB Suspend mode and Session-Off mode

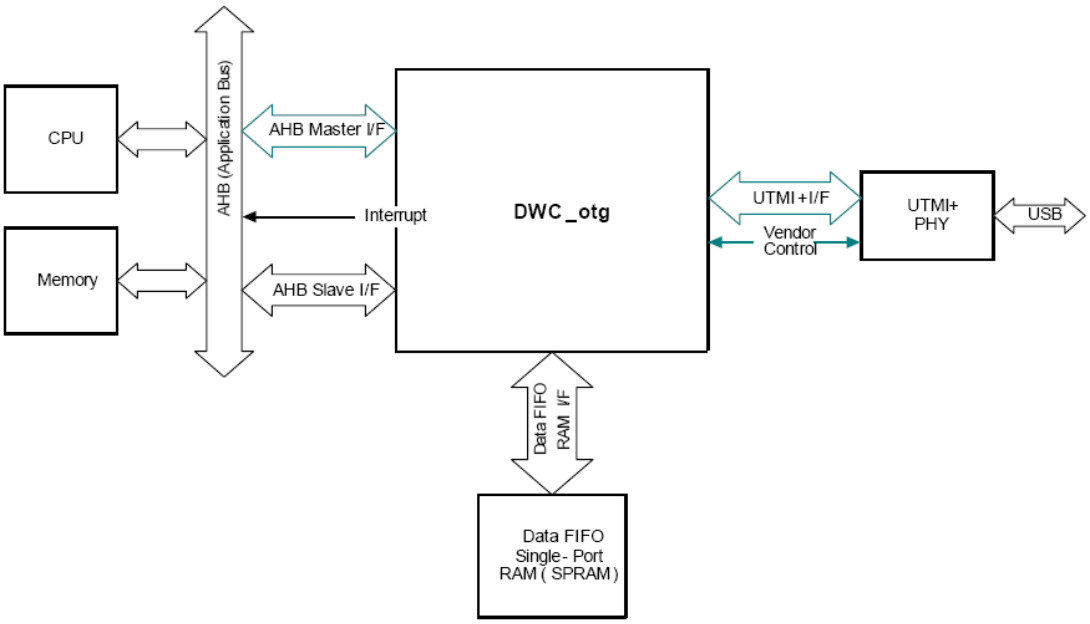


Figure 15.1 USB Controller Block Diagram

15.2 Register Description for USB 2.0 OTG Controller

By reading from and writing to the Control and Status Registers (CSRs) through the AHB Slave interface. CSRs are classified as follows:

- Core Global Registers
- Host Mode Registers
  - Host Global Registers
  - Host Port CSRs
  - Host Channel-Specific Registers
- Device Mode Registers
  - Device Global Registers
  - Device Endpoint-Specific Registers
- Power and Clock-Gating Registers
- Data FIFO (DFIFO) Access Registers

Only the Core Global, Power and Clock Gating, Data FIFO Access, and Host Port registers can be accessed in both Host and Device modes. When operating in one mode, either Device or Host, the application must not access registers from the other mode. If an illegal access occurs, a Mode Mismatch interrupt is generated and reflected in the Core Interrupt register (GINTSTS.ModeMis).

When the core switches from one mode to another, the registers in the new mode of operation must be reprogrammed as they would be after a power-on reset.

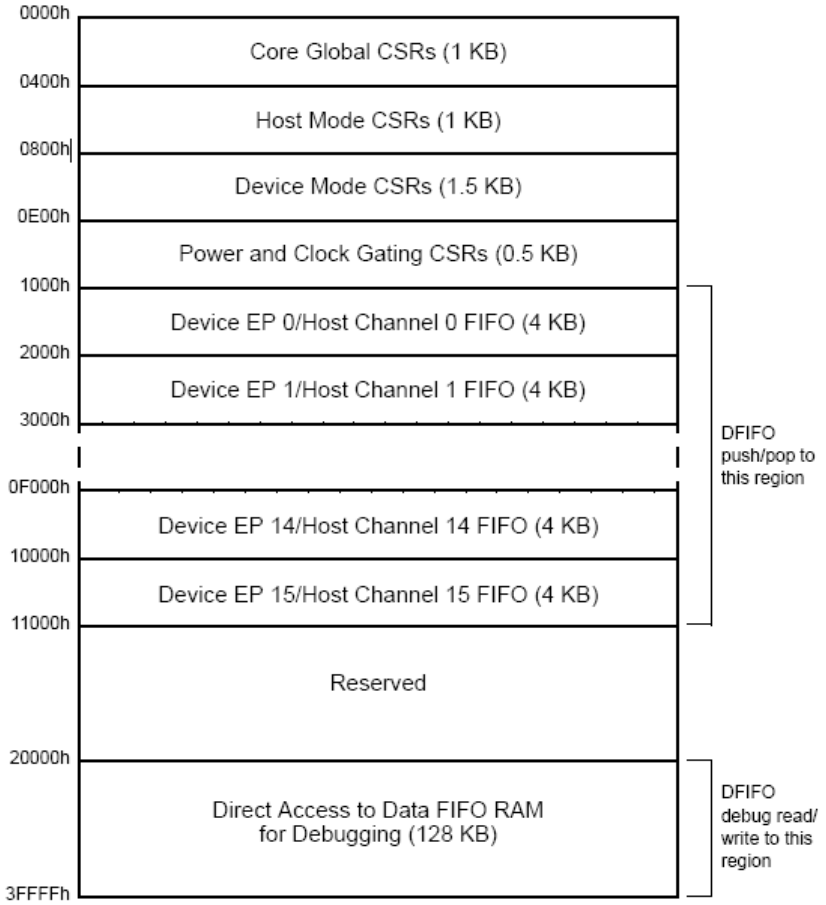


Figure 15.2 USB OTG CSR Memory Map

Table 15.1 USB Register Map (Base Address = 0xF0550000)

Core Global CSR Map					
Name	**Abbr.**	**Addr.**	**Name**	**Abbr.**	**Addr.**
Core Global Registers		0x000-0x3FF	Reserved		0x030-0x038
OTG Control and Status Register	GOTGCTL	0x000	User ID Register	GUID	0x03C

OTG Interrupt Register	GOTGINT	0x004	Reserved		0x040
Core AHB Configuration Register	GAHBCFG	0x008	User HW Config1 Register(Read Only)	GHWCFG1	0x044
Core USB Configuration Register	GUSBCFG	0x00C	User HW Config2 Register(Read Only)	GHWCFG2	0x048
Core Reset Register	GRSTCTL	0x010	User HW Config3 Register(Read Only)	GHWCFG3	0x04C
Core Interrupt Register	GINTSTS	0x014	User HW Config4 Register(Read Only)	GHWCFG4	0x050
Core Interrupt Mask Register	GINTMSK	0x018	Reserved		0x054-0x0FF
Receive Status Debug Read Register (Read Only)	GRXSTSR	0x01C	Host Periodic Transmit FIFO Size Register	HPTXFSIZ	0x100
Receive Status Read /Pop Register (Read Only)	GRXSTSP	0x020	Device IN Endpoint Transmit FIFO Size Register	DIEPTXFn	0x104-0x13C
Receive FIFO Size Register	GRXFSIZ	0x024	Reserved		0x140-0x3FF
Non-periodic Transmit FIFO Size Register	GNPTXFSIZ	0x028			
Non-periodic Transmit FIFO/Queue Status Register (Read Only)	GNPTXSTS	0x02C			
Host Mode CSR Map					
Name	**Abbr.**	**Addr.**	**Name**	**Abbr.**	**Addr.**
Host Global Registers		0x400-0x7FF	Host Channel-Specific Registers(n=0 to 15)		0x500-0x6FC
Host Configuration Register	HCFG	0x400	Host Channel 0 Characteristics Register	HCCHARn	0x500
Host Frame Interval Register	HFIR	0x404	Host Channel 0 Split Control Register	HCSPLTn	0x504
Host Frame Number/Frame Time Remaining Register	HFNUM	0x408	Host Channel 0 Interrupt Register	HCINTn	0x508
Reserved		0x40C	Host Channel 0 Interrupt Mask Register	HCINTMSKn	0x50C
Host Periodic Transmit FIFO/Queue Status Register	HPTXSTS	0x410	Host Channel 0 Transfer Size Register	HCTSIZn	0x510
Host All Channels Interrupt Register	HAINT	0x414	Host Channel 0 DMA Address Register	HCDMAn	0x514
Host All Channels Interrupt Mask Register	HAINTMSK	0x418	Reserved		0x518-0x51C
Host Port Control and Status Register		0x440-0x47C	Host Channel 1 Registers		0x520-0x53C
Host Port Control and Status Register	HPRT	0x440	Host Channel 2 Registers		0x540-0x55C
Reserved		0x444-0x4FC	~	~	~
			Host Channel 14 Registers		0x6C0-0x6DC
			Host Channel 15 Registers		0x6E0-0x6FC
			Reserved		0x6FD-0x7FF
Device Mode CSR Map					
Name	**Abbr.**	**Addr.**	**Name**	**Abbr.**	**Addr.**
Device Global Registers		0x800-0xBFF	Device Logical IN Endpoint-Specific Registers		0x900-0xAFC
Device Configuration Register	DCFG	0x800	Device Control IN Endpoint 0 Control Register	DIEPCTLn	0x900
Device Control Register	DCTL	0x804	Reserved		0x904
Device Status Register (Read Only)	DSTS	0x808	Device IN Endpoint 0 Interrupt Register	DIEPINTn	0x908
Reserved		0x80C	Reserved		0x90C
Device IN Endpoint Common Interrupt Mask Register	DIEPMSK	0x810	Device IN Endpoint 0 Transfer Size Register	DIEPTSIZn	0x910
Device OUT Endpoint Common Interrupt Mask Register	DOEPMSK	0x814	Device IN Endpoint 0 DMA Address Register	DIEPDMAn	0x914
Device All Endpoints Interrupt Register	DAINT	0x818	Device IN Endpoint Transmit FIFO Status Register	DTXFSTSn	0x918
Device All Endpoints Interrupt Mask Register	DAINTMSK	0x81C	Reserved	0x918-0x91C	
Reserved		0x820-0x824	Device IN Endpoint 1 Registers		0x920-0x93C
Reserved		0x830-	Device IN Endpoint 2 Registers		0x940-

		0x834			0x95C
Device VBUS Discharge Time Register	DVBUSDIS	0x828	~	~	~
Device VBUS Pulsing Time Register	DVBUSPULSE	0x82C	Device IN Endpoint 14 Registers		0xAC0-0xADC
Device Threshold Control Register	DTHRCTL	0x830	Device IN Endpoint 15 Registers		0xAE0-0xAFC
Device IN Endpoint FIFO Empty Interrupt Mask Register	DIEPEMPMSK	0x834	Device Logical OUT Endpoint-Specific Registers		0xB00-0xCFC
Reserved		0x838-0x8FF	Device Control OUT Endpoint 0 Control Register	DOEPCTLn	0xB00
			Reserved		0xB04
			Device OUT Endpoint 0 Interrupt Register	DOEPINTn	0xB08
			Reserved		0xB0C
			Device OUT Endpoint 0 Transfer Size Register	DOEPTSIZn	0xB10
			Device OUT Endpoint 0 DMA Address Register	DOEPDMAn	0xB14
			Reserved		0xB18-0xB1C
			Device OUT Endpoint 1 Registers		0xB20-0xB3C
			Device OUT Endpoint 2 Registers		0xB40-0xB5C
			~	~	~
			Device OUT Endpoint 14 Registers		0xCC0-0xCDC
			Device OUT Endpoint 15 Registers		0xCE0-0xCFC
			Reserved		0xCFD-0xDFF

Power and Clock Gating CSR Map

Name	Abbr.	Addr.	Name	Abbr.	Addr.
Power and Clock Gating Control Register	PCGCR	0xE00	Reserved		0xE04-0xFFF

Data FIFO(DFIFO) Access Register Map

Name	Addr.	Access	Name	Addr.	Access
Device IN Endpoint 0/Host OUT Channel 0: DFIFO Write Access	0x1000-0x1FFC	WO/RO	Device IN Endpoint140/Host OUT Channel 14: DFIFO Write Access	0xF000-0xFFFC	WO/RO
Device OUT Endpoint 0/Host IN Channel 0: DFIFO Read Access			Device OUT Endpoint 14/Host IN Channel 14: DFIFO Read Access		
Device IN Endpoint 1/Host OUT Channel 0: DFIFO Write Access	0x2000-0x1FFC	WO/RO	Device IN Endpoint 15/Host OUT Channel 15: DFIFO Write Access	[18]0x10000-0x10FFC	WO/RO
Device OUT Endpoint 1/Host IN Channel 1: DFIFO Read Access			Device OUT Endpoint 15/Host IN Channel 15: DFIFO Read Access		
~	~				

Table 15.2 USB OTG Register (Base Address = 0xF05F5000)

Name	Abbr.	Addr.
USBOTG Configuration Register	OTGCR	0x0

The Access column of each register description that follows specifies how the application and the core can access the register fields of the CSRs. The following conventions are used:

- *Read Only* (**RO**) : Register field can only be read by the application. Writes to read-only fields have no effect.
- *Write Only* (**WO**) : Register field can only be written by the application.
- *Read and Write* (**R_W**) : Register field can be read and written by the application. The application can set this field by writing 1'b1 and can clear it by writing 1'b0.
- *Read, Write, and Self Clear* (**R_W_SC**) : Register field can be read and written by the application (Read and Write), and is cleared to 1'b0 by the core (Self Clear). The

[18] Prior to access the address range "0x10000~0x10FFC", address map of OTG control register(OTGCR.OTGAMAP, 0xf05f0000) must be changed.

conditions under which the core clears this field are explained in detail in the field's description.

- *Read, Write, Self Set, and Self Clear* (**R_W_SS_SC**) : Register field can be read and written by the application (Read and Write), set to 1'b1 by the core on certain USB events (Self Set), and cleared to 1'b0 by the core (Self Clear). The conditions under which the core sets and clears this field are explained in the field's description. (Only the Port Resume bit of the Host Port Control and Status register, HPRT.PrtRes, uses this access type).
- *Read, Self Set, and Write Clear* (**R_SS_WC**) : Register field can be read by the application (Read), can be set to 1'b1 by the core on a certain internal or USB or AHB event (Self Set), and can be cleared to 1'b0 by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 has no effect on this field. The conditions under which the core sets this field are explained in detail in the field's description. (For example, interrupt bits.)
- *Read, Write Set, and Self Clear* (**R_WS_SC**) : Register field can be read by the application (Read), can be set to 1'b1 by the application with a register write of 1'b1 (Write Set), and is cleared to 1'b0 by the core (Self Clear). The application cannot clear this type of field, and a register write of 1'b0 to this bit has no effect on this field. The conditions under which the core clears this field are explained in detail in the field's description. (For example, reset signals)
- *Read, Self Set, and Self Clear or Write Clear* (**R_SS_SC_WC**) : Register field can be read by the application (Read), can be set to 1'b1 by the core on certain internal or USB or AHB events (Self Set), and can be cleared to 1'b0 either by the core itself (Self Clear) or by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 to this bit has to no effect on this field. The conditions under which the core sets or clears this field are explained in the field's description. (Only the Port Enable bit of the Host Port Control and Status register, HPRT.PrtEna, and the VStatus Done bit of the PHY Vendor Control register, GPVNDCTL.VStsDone, use this access type.)

OTG Control and Status Register (GOTGCTL) 0xF0550000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												BSesVld	ASesVld	DbncTime	ConIDSts
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				DevHNPEn	HstSetHNPEn	HNPReq	HstNegScs							SesReq	SesReqScs

The OTG Control and Status register controls the behavior and reflects the status of the OTG function of the core.

Bit	Name	Mode	Initial	Type	Description
31-20	-		0		Reserved
19	BSesVld	Device	0	RO	B-Session Valid Indicates the Device mode transceiver status. • 0: B-session is not valid. • 1: B-session is valid. In OTG mode, you can use this bit to determine if the device is connected or disconnected.
18	ASesVld	Host	0	RO	A-Session Valid Indicates the Host mode transceiver status. • 0: A-session is not valid • 1: A-session is valid
17	DbncTime	Host	0	RO	Long/Short Debounce Time Indicates the debounce time of a detected connection. • 0: Long debounce time, used for physical connections (100 ms + 2.5 µs) • 1: Short debounce time, used for soft connections (2.5 µs)
16	ConIDSts	Host and Device	1	RO	Connector ID Status Indicates the connector ID status on a connect event. • 0: The OTG core is in A-Device mode • 1: The OTG core is in B-Device mode
15:12	-		0		Reserved
11	DevHNPEn	Device	0	R_W	Device HNP Enabled The application sets this bit when it successfully receives a SetFeature.SetHNPEnable command from the connected USB host. • 0: HNP is not enabled in the application • 1: HNP is enabled in the application
10	HstSetHNPEn	Host	0	R_W	Host Set HNP Enable The application sets this bit when it has successfully enabled HNP (using the SetFeature. SetHNPEnable command) on the connected device. • 0: Host Set HNP is not enabled • 1: Host Set HNP is enabled
9	HNPReq	Device	0	R_W	HNP Request The application sets this bit to initiate an HNP request to the connected USB host. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. • 0: No HNP request • 1: HNP request
8	HstNegScs	Device	0	RO	Host Negotiation Success The core sets this bit when host negotiation is successful. The core clears this bit when the HNP Request (HNPReq) bit in this register is set. • 0: Host negotiation failure • 1: Host negotiation success

7-2	Reserved		0		
1	SesReq	Device	0	R_W	Session Request The application sets this bit to initiate a session request on the USB. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. If you use the USB 1.1 Full-Speed Serial Transceiver interface to initiate the session request, the application must wait until the VBUS discharges to 0.2 V, after the B-Session Valid bit in this register (GOTGCTL.BSesVld) is cleared. This discharge time varies between different PHYs and can be obtained from the PHY vendor. • 0: No session request • 1: Session request
0	SesReqScs	Device	0	RO	Session Request Success The core sets this bit when a session request initiation is successful. • 0: Session request failure • 1: Session request success

OTG Interrupt Register (GOTGINT) 0xF0550004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												DbnceDone	ADevTOUTChg	HstNegDet	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
						HstNegSucStsChng	SesReqSucStsChng						SesEndDet		

The application reads this register whenever there is an OTG interrupt and clears the bits in this register to clear the OTG interrupt.

Bit	Name	Mode	Initial	Type	Description
31-19	-		0		Reserved
19	DbnceDone	Host	0	R_SS_WC	Debounce Done The core sets this bit when the debounce is completed after the device connect. The application can start driving USB reset after seeing this interrupt. This bit is only valid when the HNP Capable or SRP Capable bit is set in the Core USB Configuration register (GUSBCFG.HNPCap or GUSBCFG.SRPCap, respectively).
18	ADevTOUTChg	Host and Device	0	R_SS_WC	A-Device Timeout Change () The core sets this bit to indicate that the A-device has timed out while waiting for the B-device to connect.
17	HstNegDet	Host and Device	0	R_SS_WC	Host Negotiation Detected The core sets this bit when it detects a host negotiation request on the USB.
16-10	-		0		Reserved
9	HstNegSucStsChng	Host and Device	0	R_SS_WC	Host Negotiation Success Status Change The core sets this bit on the success or failure of a USB host negotiation request. The application must read the Host Negotiation Success bit of the OTG Control and Status register (GOTGCTL.HstNegScs) to check for success or failure.
8	SesReqSucStsChng	Host and Device	0	R_SS_WC	Session Request Success Status Change The core sets this bit on the success or failure of a session request. The application must read the Session Request Success bit in the OTG Control and Status register (GOTGCTL.SesReqScs) to check for success or failure.
7-3	-		0		Reserved
2	SesEndDet	Host and Device	0	R_SS_WC	Session End Detected The core sets this bit when the utmiotg_bvalid signal is deasserted.
1-0	-		0		Reserved

Core AHB Configuration Register (GAHBCFG) 0xF0550008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							PTxFEmpLvl	NPTxFEmpLvl		DMAEn	HBstLen				GlblIntrMsk

This register can be used to configure the core after power-on or a change in mode of operation. This register mainly contains AHB system-related configuration parameters. Do not change this register after the initial programming. The application must program this register before starting any transactions on either the AHB or the USB..

Bit	Name	Mode	Initial	Type	Description
31-9	-		0		Reserved
8	PTxFEmpLvl	Host	0	R_W	Periodic TxFIFO Empty Level Indicates when the Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. • 1'b0: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty • 1'b1: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty
7	NPTxFEmpLvl	Host and Device	0	R_W	Non-Periodic TxFIFO Empty Level This bit is used only in Slave mode. In host mode and with Shared FIFO with device mode, this bit indicates when the Non-Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.NPTxFEmp) is triggered. With dedicated FIFO in device mode, this bit indicates when IN endpoint Transmit FIFO empty interrupt (DIEPINTn.TxFEmp) is triggered. • 1'b0: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is half empty • 1'b1: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is completely empty
6	-		0		Reserved
5	DMAEn	Host and Device	0	R_W	DMA Enable • 1'b0: Core operates in Slave mode • 1'b1: Core operates in a DMA mode
4-1	HBstLen	Host and Device	0	R_W	Burst Length/Type () AHB Master burst type: • 4'b0000 Single • 4'b0001 INCR • 4'b0011 INCR4 • 4'b0101 INCR8 • 4'b0111 INCR16 • Others: Reserved
0	GlblIntrMsk	Host and Device	0	R_W	Global Interrupt Mask The application uses this bit to mask or unmask the interrupt line assertion to itself. Irrespective of this bit's setting, the interrupt status registers are updated by the core. • 1'b0: Mask the interrupt assertion to the application. • 1'b1: Unmask the interrupt assertion to the application.

Core USB Configuration Register (GUSBCFG) 0xF055000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ForceDevMode	ForceHstMode							TermSelDLPulse						OtgI2CSel
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PhyLPwrClkSel		USBTrdTim				HNPCap	SRPCap		PHYSel	FSIntf	ULPI_UTMI_Sel	PHYIf	TOutCal		

This register can be used to configure the core after power-on or a changing to Host mode or Device mode. It contains USB and USB-PHY related configuration parameters. The application must program this register before starting any transactions on either the AHB or the USB. Do not make changes to this register after the initial programming.

Bit	Name	Mode	Initial	Type	Description
31		Host and Device	0	R_W	Corrupt Tx packet This bit is for debug purposes only. Never set this bit to 1.
30	ForceDevMode	Host and Device	0	R_W	Force Device Mode Writing a 1 to this bit will force the core to device mode irrespective of utmiotg_iddig input pin. • 1'b0 : Normal Mode. • 1'b1 : Force Device Mode. After setting the force bit, the application must wait at least 25ms before the change to take effect.
29	ForceHstMode	Host and Device	0	R_W	Force Host Mode Writing a 1 to this bit will force the core to host mode irrespective of utmiotg_iddig input pin. • 1'b0 : Normal Mode. • 1'b1 : Force Host Mode. After setting the force bit, the application must wait at least 25ms before the change to take effect.
28-23	-		0		Reserved
22	TermSelDLPulse	Device	0	R_W	TermSel DLine Pulsing Selection This bit selects utmi_termselect to drive data line pulse during SRP. • 1'b0: Data line pulsing using utmi_txvalid (default). • 1'b1: Data line pulsing using utmi_termsel.
21-17	-		0		Reserved
16	OtgI2CSel	Host and Device	0	RO/R_W	UTMIFS or I2C Interface Select The application uses this bit to select the I2C interface. • 1'b0: UTMI USB 1.1 Full-Speed interface for OTG signals • 1'b1: I2C interface for OTG signals(not supported)
15	PhyLPwrClkSel	Host and Device	0	R_W	PHY Low-Power Clock Select Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY can usually operate on a 48-MHz clock to save power. • 1'b0: 480-MHz Internal PLL clock • 1'b1: 48-MHz External Clock In 480 MHz mode, the UTMI interface operates at either 60 or 30-MHz, depending upon whether 8- or 16-bit data width is selected. In 48-MHz mode, the UTMI interface operates at 48MHz in FS mode and at either 48 or 6 MHz in LS mode (depending on the PHY vendor). This bit drives the utmi_fsls_low_power core output signal, and is valid only for UTMI+ PHYs.
14	-		0		Reserved

13-10	USBTrdTim	Device	4'h5	R_W	USB Turnaround Time Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM). This must be programmed to • 4'h5: When the MAC interface is 16-bit UTMI+ . • 4'h9: When the MAC interface is 8-bit UTMI+ . Note: The values above are calculated for the minimum AHB frequency of 30MHz. USB turnaround time is critical for certification where long cables and 5-Hubs are used, so if you need the AHB to run at less than 30 MHz, and if USB turnaround time is not critical, these bits can be programmed to a larger value.
9	HNPCap	Host and Device	0	RO/R_W	HNP-Capable The application uses this bit to control the OTG core's HNP capabilities. • 1'b0: HNP capability is not enabled. • 1'b1: HNP capability is enabled..
8	SRPCap	Host and Device	0	RO/R_W	SRP-Capable The application uses this bit to control SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session. • 1'b0: SRP capability is not enabled. • 1'b1: SRP capability is enabled.
7	-		0		Reserved
6	PHYSel	Host and Device	0	WO/R_W	USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select The application uses this bit to select either a high-speed UTMI+, or a full-speed transceiver. • 1'b0: USB 2.0 high-speed UTMI+ • 1'b1: USB 1.1 full-speed serial transceiver
5	FSIntf	Host and Device	0	WO/R_W	Full-Speed Serial Interface Select The application uses this bit to select either a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface. • 1'b0: 6-pin unidirectional full-speed serial interface • 1'b1: 3-pin bidirectional full-speed serial interface
4	ULPI_UTMI_Sel	Host and Device	0	RO/R_W	ULPI or UTMI+ Select The application uses this bit to select either a UTMI+ interface or ULPI Interface. • 1'b0: UTMI+ Interface • 1'b1: ULPI Interface(not supported)
3	PHYIf	Host and Device	0	RO/R_W	PHY Interface The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. • 1'b0: 8 bits (not supported) • 1'b1: 16 bits
2-0	TOutCal	Host and Device	0	R_W	HS/FS Timeout Calibration The number of PHY clocks that the application programs in this field is added to the high-speed/full-speed interpacket timeout duration in the core to account for any additional delays introduced by the PHY. This can be required, because the delay introduced by the PHY in generating the linestate condition can vary from one PHY to another. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed

					operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of enumeration. The number of bit times added per PHY clock are: High-speed operation: • One 30-MHz PHY clock = 16 bit times • One 60-MHz PHY clock = 8 bit times Full-speed operation: • One 30-MHz PHY clock = 0.4 bit times • One 60-MHz PHY clock = 0.2 bit times • One 48-MHz PHY clock = 0.25 bit times

Core Reset Register (GRSTCTL) 0xF0550010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHBIdle	DMAReq														
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					TxFNum					TxFFlsh	RxFFlsh	INTknQFlsh	FrmCntrRst	HSftRst	CSftRst

The application uses this register to reset various hardware features inside the core..

Bit	Name	Mode	Initial	Type	Description
31	AHBIdle	Host and Device	1	RO	AHB Master Idle Indicates that the AHB Master State Machine is in the IDLE condition.
30	DMAReq	Host and Device	0	RO	DMA Request Signal Indicates that the DMA request is in progress. Used for debug.
29-11	-		0		Reserved
10-6	TxFNum	Host and Device	0	R_W	TxFIFO Number This is the FIFO number that must be flushed using the TxFIFO Flush bit. This field must not be changed until the core clears the TxFIFO Flush bit. • 5'h0: - Non-periodic TxFIFO flush in Host mode - Tx FIFO 0 flush in device mode • 5'h1: - Periodic TxFIFO flush in Host mode - TXFIFO 1 flush in device mode • 5'h2: - TXFIFO 2 flush in device mode ... • 5'hF: - TXFIFO 15 flush in device mode • 5'h10: Flush all the transmit FIFOs in device or host mode.
5	TxFFlsh	Host and Device	0	R_WS_SC	TxFIFO Flush This bit selectively flushes a single or all transmit FIFOs, but cannot do so if the core is in the midst of a transaction. The application must write this bit only after checking that the core is neither writing to the TxFIFO nor reading from the TxFIFO. Verify using these registers: • Read—NAK Effective Interrupt ensures the core is not reading from the FIFO • Write—GRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO. Flushing is normally recommended when FIFOs are reconfigured. FIFO flushing is also recommended during device endpoint disable. The application must wait until the core clears this bit before performing any operations. This bit takes eight clocks to clear, using the slower clock of phy_clk or hclk.
4	RxFFlsh	Host and Device	0	R_WS_SC	RxFIFO Flush The application can flush the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking that the core is neither reading from the RxFIFO nor writing to the RxFIFO. The application must wait until the bit is cleared before performing any other operations. This bit requires 8 clocks (slowest of PHY or AHB clock)

					to clear.
3	INTknQFlsh	Device	0	R_WS_SC	IN Token Sequence Learning Queue Flush The application writes this bit to flush the IN Token Sequence Learning Queue.
2	FrmCntrRst	Host	0	R_WS_SC	Host Frame Counter Reset The application writes this bit to reset the (micro)frame number counter inside the core. When the (micro)frame counter is reset, the subsequent SOF sent out by the core has a (micro)frame number of 0.
1	HSftRst	Host and Device	0	R_WS_SC	HClk Soft Reset The application uses this bit to flush the control logic in the AHB Clock domain. Only AHB Clock Domain pipelines are reset. • FIFOs are not flushed with this bit. • All state machines in the AHB clock domain are reset to the Idle state after terminating the transactions on the AHB, following the protocol. • CSR control bits used by the AHB clock domain state machines are cleared. • To clear this interrupt, status mask bits that control the interrupt status and are generated by the AHB clock domain state machine are cleared. • Because interrupt status bits are not cleared, the application can get the status of any core events that occurred after it set this bit. This is a self-clearing bit that the core clears after all necessary logic is reset in the core. This can take several clocks, depending on the core's current state.
0	CSftRst	Host and Device	0	R_WS_SC	Core Soft Reset Resets the hclk and phy_clock domains as follows: • Clears the interrupts and all the CSR registers except the following register bits: - PCGCCTL.RstPdwnModule - PCGCCTL.GateHclk - PCGCCTL.PwrClmp - PCGCCTL.StopPPhyLPwrClkSelclk - GUSBCFG.PhyLPwrClkSel - GUSBCFG.DDRSel - GUSBCFG.PHYSel - GUSBCFG.FSIntf - GUSBCFG.ULPI_UTMI_Sel - GUSBCFG.PHYIf - HCFG.FSLSPclkSel - DCFG.DevSpd - GGPIO • All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. • Any transactions on the AHB Master are terminated as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which can take several clocks, depending on the current state of the core. Once this bit is cleared software must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). Software must also

					must check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation. Typically software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain has to be reset for proper operation.

Core Interrupt Register (GINTSTS) 0xF0550014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WkUpInt	SessReqInt	DisconnInt	ConIDStsChng		PTxFEmp	HChInt	PrtInt		FetSusp	incompIP/ incompISOOUT	incompISOIN	OEPInt	IEPInt	EPMis	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EOPF	ISOOutDrop	EnumDone	USBRst	USBSusp	ErlySusp			GOUTNakEff	GINNakEff		RxFLvl	Sof	OTGInt	ModeMis	CurMod

This register interrupts the application for system-level events in the current mode of operation (Device mode or Host mode). Some of the bits in this register are valid only in Host mode, while others are valid in Device mode only. This register also indicates the current mode of operation. In order to clear the interrupt status bits of type R_SS_WC, the application must write 1'b1 into the bit.

The FIFO status interrupts are read only; once software reads from or writes to the FIFO while servicing these interrupts, FIFO interrupt conditions are cleared automatically.

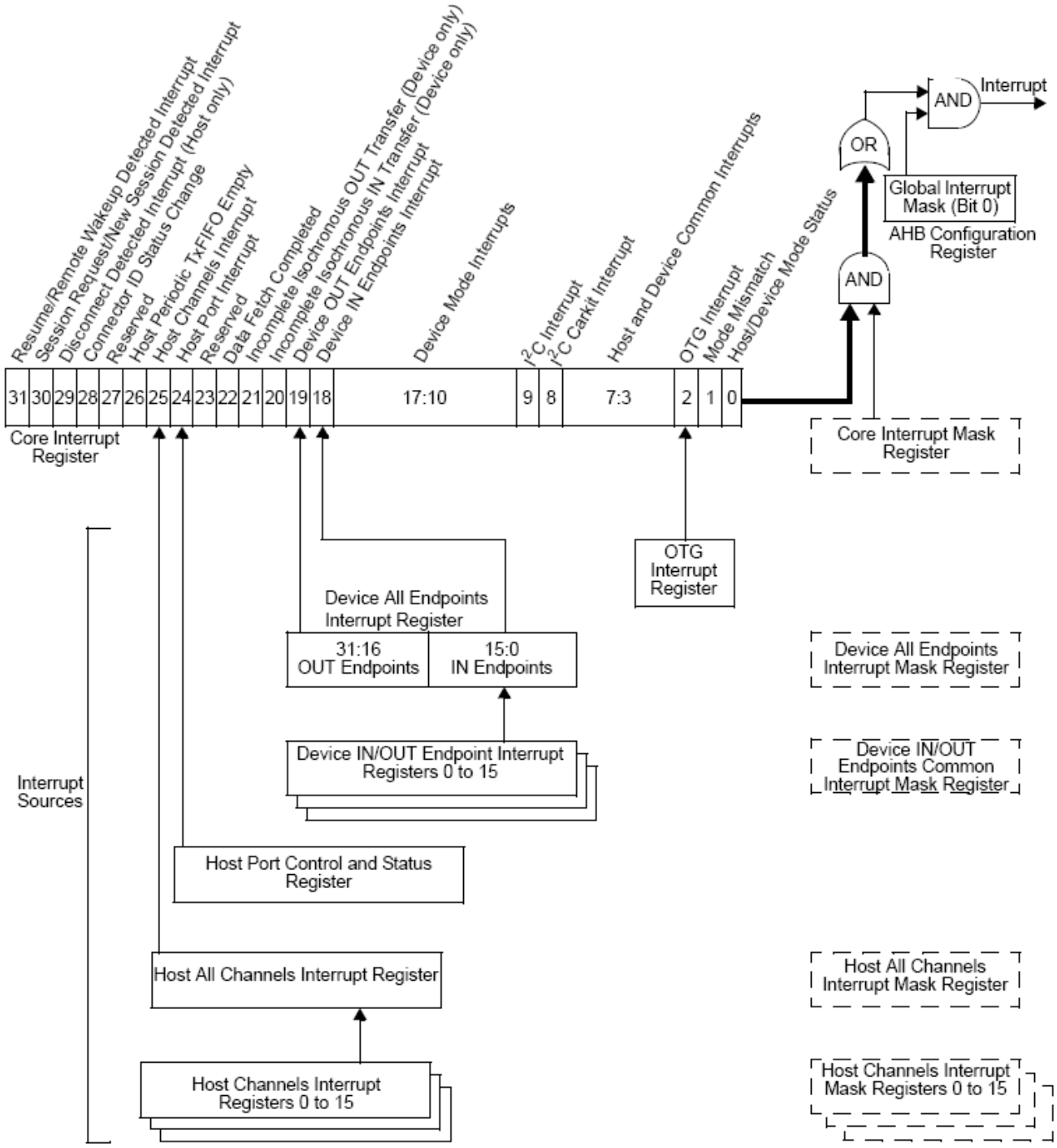
The Application must clear the GINTSTS register at initialization before unmasking the interrupt bit to avoid any interrupts generated prior to initialization.

Bit	Name	Mode	Initial	Type	Description
31	WkUpInt	Host and Device	0	R_SS_WC	Resume/Remote Wakeup Detected Interrupt In Device mode, this interrupt is asserted when a resume is detected on the USB. In Host mode, this interrupt is asserted when a remote wakeup is detected on the USB.
30	SessReqInt	Host and Device	0	R_SS_WC	Session Request/New Session Detected Interrupt In Host mode, this interrupt is asserted when a session request is detected from the device. In Device mode, this interrupt is asserted when the utmiotg_bvalid signal goes high.
29	DisconnInt	Host	0	R_SS_WC	Disconnect Detected Interrupt Asserted when a device disconnect is detected.
28	ConIDStsChng	Host and Device	0	R_SS_WC	Connector ID Status Change The core sets this bit when there is a change in connector ID status.
27	-		0		Reserved
26	PTxFEmp	Host	1	RO	Periodic TxFIFO Empty Asserted when the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.PTxFEmpLvl).
25	HChInt	Host	0	RO	Host Channels Interrupt The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred, and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.
24	PrtInt	Host	0	RO	Host Port Interrupt The core sets this bit to indicate a change in port status of one of ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host

					Port Control and Status register to clear this bit.
23	-		0		Reserved
22	FetSusp	Device	0	R_SS_WC	Data Fetch Suspended This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data for IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application: • Sets a global non-periodic IN NAK handshake • Disables In endpoints • Flushes the FIFO • Determines the token sequence from the IN Token Sequence Learning Queue • Re-enables the endpoints • Clears the global non-periodic IN NAK handshake If the global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token is received: the core generates an “IN token received when FIFO empty” interrupt. The OTG then sends the host a NAK response. To avoid this scenario, the application can check the GINTSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a global NAK handshake. Alternatively, the application can mask the “IN token received when FIFO empty” interrupt when clearing a global IN NAK handshake.
21	incomplP	Host	0	R_SS_WC	Incomplete Periodic Transfer In Host mode, the core sets this interrupt bit when there are incomplete periodic transactions still pending which are scheduled for the current microframe.
	incomplSOOUT	Device			Incomplete Isochronous OUT Transfer The Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.
20	incomplSOIN	Device	0	R_SS_WC	Incomplete Isochronous IN Transfer The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.
19	OEPInt	Device	0	RO	OUT Endpoints Interrupt The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding Device OUT Endpoint-n Interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.

18	IEPInt	Device	0	RO	IN Endpoints Interrupt The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on which the interrupt occurred, and then read the corresponding Device IN Endpoint-n Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit.
17	EPMis	Device	0	RO	Endpoint Mismatch Interrupt
16	-				Reserved
15	EOPF	Device	0	R_SS_WC	End of Periodic Frame Interrupt Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe.
14	ISOOutDrop	Device	0	R_SS_WC	Isochronous OUT Packet Dropped Interrupt The core sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO doesn't have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.
13	EnumDone	Device	0	R_SS_WC	Enumeration Done The core sets this bit to indicate that speed enumeration is complete. The application must read the Device Status (DSTS) register to obtain the enumerated speed.
12	USBRst	Device	0	R_SS_WC	USB Reset The core sets this bit to indicate that a reset is detected on the USB.
11	USBSusp	Device	0	R_SS_WC	USB Suspend The core sets this bit to indicate that a suspend was detected on the USB. The core enters the Suspended state when there is no activity on the phy_line_state_i signal for an extended period of time.
10	ErlySusp	Device	0	R_SS_WC	Early Suspend The core sets this bit to indicate that an Idle state has been detected on the USB for 3 ms.
9-8	-		0		Reserved
7	GOUTNakEff	Device	0	R_SS_WC	Global OUT NAK Effective Indicates that the Set Global OUT NAK bit in the Device Control register (DCTL.SGOUTNak), set by the application, has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the Device Control register (DCTL.CGOUTNak).

6	GINNakEff	Device	0	R_SS_WC	Global IN Non-periodic NAK Effective Indicates that the Set Global Non-periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak), set by the application, has been taken effect in the core. That is, the core has sampled the Global IN NAK bit set by the application. This bit can be cleared by clearing the Clear Global Non-periodic IN NAK bit in the Device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit.
5	-		0		Reserved
4	RxFLvl	Host and Device	0	RO	RxFIFO Non-Empty Indicates that there is at least one packet pending to be read from the RxFIFO.
3	Sof	Host and Device	0	R_SS_WC	Start of (micro)Frame In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF (HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In Device mode, in the core sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device Status register to get the current (micro)frame number. This interrupt is seen only when the core is operating at either HS or FS.
2	OTGInt	Host and Device	0	RO	OTG Interrupt The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit.
1	ModeMis	Host and Device	0	R_SS_WC	Mode Mismatch Interrupt The core sets this bit when the application is trying to access: • A Host mode register, when the core is operating in Device mode • A Device mode register, when the core is operating in Host mode The register access is completed on the AHB with an OKAY response, but is ignored by the core internally and doesn't affect the operation of the core.
0	CurMod	Host and Device	0	RO	Current Mode of Operation Indicates the current mode of operation. • 1'b0: Device mode • 1'b1: Host mode



Note: Because an interrupt mask only masks an interrupt, software must clear an interrupt before unmasking it, to avoid servicing an old interrupt.

Figure 15.3 USB OTG Controller Interrupt Hierarachy

Core Interrupt Mask Register (GINTMSK) 0xF0550018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WkUpIntMsk	SessReqIntMsk	DisconnIntMsk	ConIDStsChngMsk		PTxFEmpMsk	HChIntMsk	PrtIntMsk		FetSuspMsk	incomplPMsk/ incompISOOUTMsk	incompISOINMsk	OEPIntMsk	IEPIntMsk	EPMisMsk	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EOPFMsk	ISOOutDropMsk	EnumDoneMsk	USBRstMsk	USBSuspMsk	ErlySuspMsk			GOUTNakEffMsk	GINNakEffMsk		RxFLvlMsk	SofMsk	OTGIntMsk	ModeMisMsk	

This register works with the Core Interrupt register to interrupt the application. When an interrupt bit is masked, the interrupt associated with that bit is not generated. However, the Core Interrupt (GINTSTS) register bit corresponding to that interrupt is still set.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

Bit	Name	Mode	Initial	Type	Description
31	WkUpIntMsk	Host and Device	0	R_W	Resume/Remote Wakeup Detected Interrupt Mask
30	SessReqIntMsk	Host and Device	0	R_W	Session Request/New Session Detected Interrupt Mask()
29	DisconnIntMsk	Host and Device	0	R_W	Disconnect Detected Interrupt Mask
28	ConIDStsChngMsk	Host and Device	0	R_W	Connector ID Status Change Mask
27	-		0		Reserved
26	PTxFEmpMsk	Host	0	R_W	Periodic TxFIFO Empty Mask
25	HChIntMsk	Host	0	R_W	Host Channels Interrupt Mask
24	PrtIntMsk	Host	0	R_W	Host Port Interrupt Mask
23	-		0		Reserved
22	FetSuspMsk	Device	0	R_W	Data Fetch Suspended Mask
21	incomplPMsk	Host	0	R_W	Incomplete Periodic Transfer Mask
	incomplSOOUTMsk	Device			Incomplete Isochronous OUT Transfer Mask
20	incomplSOINMsk	Device	0	R_W	Incomplete Isochronous IN Transfer Mask
19	OEPIntMsk	Device	0	R_W	OUT Endpoints Interrupt Mask
18	IEPIntMsk	Device	0	R_W	IN Endpoints Interrupt Mask
17	EPMisMsk	Device	0	R_W	Endpoint Mismatch Interrupt Mask
16	-				Reserved
15	EOPFMsk	Device	0	R_W	End of Periodic Frame Interrupt Mask
14	ISOOutDropMsk	Device	0	R_W	Isochronous OUT Packet Dropped Interrupt Mask
13	EnumDoneMsk	Device	0	R_W	Enumeration Done Mask
12	USBRstMsk	Device	0	R_W	USB Reset Mask
11	USBSuspMsk	Device	0	R_W	USB Suspend Mask
10	ErlySuspMsk	Device	0	R_W	Early Suspend Mask
9-8	-		0		Reserved
7	GOUTNakEffMsk	Device	0	R_W	Global OUT NAK Effective Mask
6	GINNakEffMsk	Device	0	R_W	Global Non-periodic IN NAK Effective Mask
5	-		0		Reserved
4	RxFLvlMsk	Host and Device	0	R_W	Receive FIFO Non-Empty Mask
3	SofMsk	Host and Device	0	R_W	Start of (micro)Frame Mask ()
2	OTGIntMsk	Host and Device	0	R_W	OTG Interrupt Mask
1	ModeMisMsk	Host and Device	0	R_W	Mode Mismatch Interrupt Mask
0	-		0		Reserved

Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP)

0xF055001C(Read)/0xF0550020(Pop)

A read to the Receive Status Debug Read register returns the contents of the top of theReceive FIFO. A read to the Receive Status Read and Pop register additionally pops the top data entry out of the RxFIFO.

The receive status contents must be interpreted differently in Host and Device modes. The core ignores the receive status pop/read when the receive FIFO is empty and returns a value of 32'h0000_0000. The application must only pop the Receive Status FIFO when the Receive FIFO Non-Empty bit of the Core Interrupt register (GINTSTS.RxFLvl) is asserted.

The following table shows the use of these registers in Host mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											PktSts				DPID[1]
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DPID[0]	BCnt											ChNum			

Bit	Name	Mode	Initial	Type	Description
31-21	-		0		Reserved
20-17	PktSts	Host	0	RO	Packet Status Indicates the status of the received packet • 4'b0010: IN data packet received • 4'b0011: IN transfer completed (triggers an interrupt) • 4'b0101: Data toggle error (triggers an interrupt) • 4'b0111: Channel halted (triggers an interrupt) • Others: Reserved
16-15	DPID	Host	0	RO	Data PID Indicates the Data PID of the received packet • 2'b00: DATA0 • 2'b10: DATA1 • 2'b01: DATA2 • 2'b11: MDATA
14-4	BCnt	Host	0	RO	Byte Count Indicates the byte count of the received IN data packet.
3-0	ChNum	Host	0	RO	Channel Number () Indicates the channel number to which the current received packet belongs.

The following table shows the use of these registers in Device mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							FN				PktSts				DPID[1]
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DPID[0]	BCnt											EPNum			

Bit	Name	Mode	Initial	Type	Description
31-25	-		0		Reserved
24-21	FN	Device	0	RO	Frame Number This is the least significant 4 bits of the (micro)frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.
20-17	PktSts	Device	0	RO	Packet Status () Indicates the status of the received packet • 4'b0001: Global OUT NAK (triggers an interrupt) • 4'b0010: OUT data packet received

					• 4'b0011: OUT transfer completed (triggers an interrupt) • 4'b0100: SETUP transaction completed (triggers an interrupt) • 4'b0110: SETUP data packet received • Others: Reserved
16-15	DPID	Device	0	RO	Data PID Indicates the Data PID of the received OUT data packet • 2'b00: DATA0 • 2'b10: DATA1 • 2'b01: DATA2 • 2'b11: MDATA
14-4	BCnt	Device	0	RO	Byte Count Indicates the byte count of the received IN data packet.
3-0	EPNum	Device	0	RO	Endpoint Number Indicates the endpoint number to which the current received packet belongs.

Receive FIFO Size Register (GRXFSIZ) **0xF0550024**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RxFDep															

The application can program the RAM size and the memory start address for the Nonperiodic TxFIFO.

Bit	Name	Mode	Initial	Type	Description
31-16	-		0		Reserved
15	RxFDep	Host and Device	0	RO/R_W	RxFIFO Depth This value is in terms of 32-bit words. • Minimum value is 16 • Maximum value is 4160

Non-Periodic Transmit FIFO Size Register (GNPTXFSIZ) 0xF0550028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NPTxFDep/ INEPTxF0Dep															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NPTxFStAddr/ INEPTxF0StAddr															

The application can program the RAM size that must be allocated to the RxFIFO.

Bit	Name	Mode	Initial	Type	Description
31-16	NPTxFDep	Host	0	RO/R_W	Non-periodic TxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16. Maximum value is 4160.
	INEPTxF0Dep	Device			IN Endpoint TxFIFO 0 Depth This value is in terms of 32-bit words. Minimum value is 16. Maximum value is 4160.
15-0	NPTxFStAddr	Host	0	RO/R_W	Non-periodic Transmit RAM Start Address This field contains the memory start address for Non-periodic Transmit FIFO RAM.
	INEPTxF0StAddr	Device			IN Endpoint FIFO0 Transmit RAM Start Address This field contains the memory start address for IN Endpoint Transmit FIFO# 0.

Non-Periodic Transmit FIFO/Queue Status Register (GNPTXSTS) 0xF055002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NPTxQTop							NPTxQSpcAvail							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NPTxFSpcAvail															

In Device mode, this register is invalid.

This read-only register contains the free space information for the Non-periodic TxFIFO and the Non-periodic Transmit Request Queue.

Bit	Name	Mode	Initial	Type	Description
31	-		0		Reserved
30-24	NPTxQTop	Host	0	RO	Top of the Non-periodic Transmit Request Queue Entry in the Non-periodic Tx Request Queue that is currently being processed by the MAC. • Bits [30:27]: Channel/endpoint number • Bits [26:25]: - 2'b00: IN/OUT token - 2'b01: Zero-length transmit packet (device IN/host OUT) - 2'b10: PING/CSPLIT token - 2'b11: Channel halt command • Bit [24]: Terminate (last entry for selected channel/endpoint)
23-16	NPTxQSpcAvail	Host	0	RO	Non-periodic Transmit Request Queue Space Available Indicates the amount of free space available in the Non-periodic Transmit Request Queue. This queue holds both IN and OUT requests in Host mode. • 8'h0: Non-periodic Transmit Request Queue is full • 8'h1: 1 location available • 8'h2: 2 locations available • n: n locations available ($0 \le n \le 8$) • Others: Reserved
15-0	NPTxFSpcAvail	Host	0	RO	Non-periodic TxFIFO Space Avail Indicates the amount of free space available in the Non-periodic TxFIFO. Values are in terms of 32-bit words. • 16'h0: Non-periodic TxFIFO is full • 16'h1: 1 word available • 16'h2: 2 words available • 16' hn: n words available (where $0 \le n \le$ 32,768) • 16'h8000: 32,768 words available • Others: Reserved

User ID Register (GUID) 0xF055003C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UserID[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UserID[15:0]															

This is a read/write register containing the User ID.

This register can be used in the following ways:

- To store the version or revision of your system
- To store hardware configurations
- As a scratch register

Bit	Name	Mode	Initial	Type	Description
31-0	UserID	Host and Device	0x1234567	R_W	User ID Application-programmable ID field

User HW Config1 Register (GHWCFG1) 0xF0550044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
epdir [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
epdir [15:0]															

Bit	Name	Mode	Initial	Type	Description
31-0	epdir	Host and Device	0	RO	Endpoint Direction Two bits per endpoint represent the direction. • 2'b00: BIDIR (IN and OUT) endpoint • 2'b01: IN endpoint • 2'b10: OUT endpoint • 2'b11: Reserved Bits [31:30]: Endpoint 15 direction Bits [29:28]: Endpoint 14 direction ... Bits [3:2]: Endpoint 1 direction Bits[1:0]: Endpoint 0 direction (always BIDIR)

User HW Config2 Register (GHWCFG2) 0xF0550048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TknQDepth					PTxQDepth		NPTxQDepth				DynFifoSizing	PerioSupport	NumHstChnl[3:2]	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NumHstChnl[1:0]		NumDevEps				FSPhyType		HSPhyType		SingPnt	OtgArch				

Bit	Name	Mode	Initial	Type	Description
31	-		0		Reserved
30:26	TknQDepth		5'b01000	RO	Device Mode IN Token Sequence Learning Queue Depth: Range: 0–30
25:24	PTxQDepth		2'b10	RO	Host Mode Periodic Request Queue Depth • 2'b00: 2 • 2'b01: 4 • 2'b10: 8 • Others: Reserved
23:22	NPTxQDepth		2'b10	RO	Non-periodic Request Queue Depth • 2'b00: 2 • 2'b01: 4 • 2'b10: 8 • Others: Reserved
21:20	-		0		Reserved
19	DynFifoSizing		1	RO	Dynamic FIFO Sizing Enabled • 1'b0: No • 1'b1: Yes
18	PerioSupport		1	RO	Periodic OUT Channels Supported in Host Mode • 1'b0: No • 1'b1: Yes
17:14	NumHstChnl		4'b1111	RO	Number of Host Channels Indicates the number of host channels supported by the core in Host mode. The range of this field is 0–15: 0 specifies 1 channel, 15 specifies 16 channels.
13:10	NumDevEps		4'b1111	RO	Number of Device Endpoints Indicates the number of device endpoints supported by the core in Device mode in addition to control endpoint 0. The range of this field is 1–15.
9:8	FSPhyType		2'b00	RO	Full-Speed PHY Interface Type • 2'b00: Full-speed interface not supported • 2'b01: Dedicated full-speed interface • 2'b10: FS pins shared with UTMI+ pins • 2'b11: FS pins shared with ULPI pins
7:6	HSPhyType		2'b11	RO	High-Speed PHY Interface Type • 2'b00: High-Speed interface not supported • 2'b01: UTMI+ • 2'b10: ULPI • 2'b11: UTMI+ and ULPI
5	SingPnt		1	RO	Point-to-Point • 1'b0: Multi-point application • 1'b1: Single-point application
4:3	OtgArch		2'b10	RO	Architecture • 2'b00: Slave-Only • 2'b01: External DMA • 2'b10: Internal DMA
2:0	OtgMode		3'b000	RO	Mode of Operation • 3'b000: HNP- and SRP-Capable OTG (Host & Device) • 3'b001: SRP-Capable OTG (Host & Device) • 3'b010: Non-HNP and Non-SRP Capable OTG (Host & Device)

					• 3’b011: SRP-Capable Device • 3’b100: Non-OTG Device • 3’b101: SRP-Capable Host • 3’b110: Non-OTG Host • Others: Reserved

User HW Config3 Register (GHWCFG3) 0xF055004C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DfifoDepth															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RstType	OptFeature			OtgEn	PktSizeWidth			XferSizeWidth			

Bit	Name	Mode	Initial	Type	Description
31-16	DfifoDepth		16'h8000	RO	DFIFO Depth This value is in terms of 32-bit words. • Minimum value is 32 • Maximum value is 4160
15-12	-		0		Reserved
11	RstType		0	RO	Reset Style for Clocked always Blocks in RTL • 1'b0: Asynchronous reset is used in the core • 1'b1: Synchronous reset is used in the core
10	OptFeature		0	RO	Optional Features Removed Indicates whether the User ID register, GPIO interface ports, and SOF toggle and counter ports were removed for gate count optimization by enabling Remove Optional Features during coreConsultant configuration. • 1'b0: No • 1'b1: Yes
9-8	-		0		Reserved
7	OtgEn		1	RO	OTG Function Enabled The application uses this bit to indicate core's OTG capabilities. • 1'b0: Not OTG capable • 1'b1: OTG Capable
6-4	PktSizeWidth		3'b110	RO	Width of Packet Size Counters • 3'b000: 4 bits • 3'b001: 5 bits • 3'b010: 6 bits • 3'b011: 7 bits • 3'b100: 8 bits • 3'b101: 9 bits • 3'b110: 10 bits • Others: Reserved
3-0	XferSizeWidth		4'b1001	RO	Width of Transfer Size Counters • 4'b0000: 11 bits • 4'b0001: 12 bits ... • 4'b1000: 19 bits • Others: Reserved

User HW Config4 Register (GHWCFG4) **0xF0550050**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		INEps				DedFifoMode	SessEndFltr	BValidFltr	AValidFltr	VBusValidFltr	IddgFltr	NumCtlEps			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PhyDataWidth										AhbFreq	EnablePwrOpt	NumDevPerioEps			

Bit	Name	Mode	Initial	Type	Description
31-30	-		0		Reserved
29-26	INEps		4'b111	RO	Endpoints Range 0 -15 • 0 : 1 IN Endpoint • 1 : 2 IN Endpoints • .... 15 : 16 IN Endpoints
25	DedFifoMode		1	RO	Enable Dedicated Transmit FIFO for device IN Endpoints • 1'b0 : Dedicated Transmit FIFO Operation not enabled. • 1'b1 : Dedicated Transmit FIFO Operation enabled.
24	SessEndFltr		1	RO	"session_end" Filter Enabled () • 1'b0: No filter • 1'b1: Filter
23	BValidFltr		1	RO	"b_valid" Filter Enabled • 1'b0: No filter • 1'b1: Filter
22	AValidFltr		1	RO	"a_valid" Filter Enabled • 1'b0: No filter • 1'b1: Filter
21	VBusValidFltr		1	RO	"vbus_valid" Filter Enabled • 1'b0: No filter • 1'b1: Filter
20	IddgFltr		1	RO	"iddig" Filter Enable • 1'b0: No filter • 1'b1: Filter
19-16	NumCtlEps		4'b0000	RO	Number of Device Mode Control Endpoints in Addition to Endpoint 0 Range: 0-15
15-14	PhyDataWidth		2'b10	RO	UTMI+ PHY/ULPI-to-Internal UTMI+ Wrapper Data Width When a ULPI PHY is used, an internal wrapper converts ULPI to UTMI+ . • 2'b00: 8 bits • 2'b01: 16 bits • 2'b10: 8/16 bits, software selectable • Others: Reserved
13-6	-		0		Reserved
5	AhbFreq		1	RO	Minimum AHB Frequency Less Than 60 MHz • 1'b0: No • 1'b1: Yes
4	EnablePwrOpt		1	RO	Enable Power Optimization • 1'b0: No • 1'b1: Yes
3-0	NumDevPerioEps		4'b0000	RO	Number of Device Mode Periodic IN Endpoints Range: 0–15

Host Periodic Transmit FIFO Size Register (HPTXFSIZ) **0xF0550100**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PTxFSize															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PTxFStAddr

This register holds the size and the memory start address of the Periodic TxFIFO.

Bit	Name	Mode	Initial	Type	Description
31-0	PTxFSize	Host	0	RO/R_W	Host Periodic TxFIFO Depth This value is in terms of 32-bit words. • Minimum value is 16 • Maximum value is 4160
15-0	PTxFStAddr	Host	0	RO/R_W	Host Periodic TxFIFO Start Address • OTG_RX_DFIFO_DEPTH + OTG_TX_HNPERIO_DFIFO_DEPTH

Device IN Endpoint Transmit Fifo Size Register: (DIEPTXFn) **0xF0550104+(FIFO_number-1)*0x4**

FIFO_number:1≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INEPnTxFDep															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPnTxFStAddr															

This register holds the size and the memory start address of IN endpoint TxFIFOs implemented in Device mode. Each FIFO holds the data for one IN endpoint. This register is repeated for instantiated IN endpoint FIFOs 1 to 15. For IN endpoint FIFO 0 use GNPTXFSIZ register for programming the size and memory start address.

Bit	Name	Mode	Initial	Type	Description
31-0	INEPnTxFDep	Device	0	RO/R_W	IN Endpoint TxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16. Maximum value is 4160.
15-0	INEPnTxFStAddr	Device	0	RO/R_W	IN Endpoint FIFOn Transmit RAM Start Address This field contains the memory start address for IN endpoint Transmit FIFOn (0<n< = 15). OTG_RX_DFIFO_DEPTH + SUM 0 to n – 1 (OTG_DINEP_TXFIFO_DEPTH_n) For example start address of IN endpoint FIFO 1 is OTG_RX_DFIFO_DEPTH + OTG_DINEP_TXFIFO_DEPTH_0 The start address of IN endpoint FIFO 2 is OTG_RX_DFIFO_DEPTH + OTG_DINEP_TXFIFO_DEPTH_0 + OTG_DINEP_TXFIFO_DEPTH_1

Host Configuration Register (HCFG) 0xF0550400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													FSLSSupp	FSLSPclkSel	

This register configures the core after power-on. Do not make changes to this register after initializing the host.

Bit	Name	Mode	Initial	Type	Description
31-3	-		0		Reserved
2	FSLSSupp	Host	0	R_W	FS- and LS-Only Support The application uses this bit to control the core's enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even if the connected device supports HS traffic. Do not make changes to this field after initial programming. • 1'b0: HS/FS/LS, based on the maximum speed supported by the connected device • 1'b1: FS/LS-only, even if the connected device can support HS
1-0	FSLSPclkSel	Host	0	R_W	FS/LS PHY Clock Select When the core is in FS Host mode • 2'b00: PHY clock is running at 30/60 MHz • 2'b01: PHY clock is running at 48 MHz • Others: Reserved When the core is in LS Host mode • 2'b00: PHY clock is running at 30/60 MHz. When the UTMI+/ULPI PHY Low Power mode is not selected, use 30/60 MHz. • 2'b01: PHY clock is running at 48 MHz. When the UTMI+ PHY Low Power mode is selected, use 48MHz if the PHY supplies a 48 MHz clock during LS mode. • 2'b10: PHY clock is running at 6 MHz. In USB 1.1 FS mode, use 6 MHz when the UTMI+ PHY Low Power mode is selected and the PHY supplies a 6 MHz clock during LS mode. If you select a 6 MHz clock during LS mode, you must do a soft reset. • 2'b11: Reserved

Host Frame Interval Register (HFIR) 0xF0550404

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FrInt															

This register stores the frame interval information for the current speed to which the otg core has enumerated.

Bit	Name	Mode	Initial	Type	Description
31-16	-		0		Reserved
15-0	FrInt	Host	60000	R_W	Frame Interval The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro-SOFs (HS) or Keep-Alive tokens (HS). This field contains the number of PHY clocks that constitute the required frame interval. The default value set in this field for a FS operation when the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHY clock specified in the FS/LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). Do not change the value of this field after the initial configuration. • 125 µs * (PHY clock frequency for HS) • 1 ms * (PHY clock frequency for FS/LS)

Host Frame Number/Frame Time Remaining Register (HFNUM) 0xF0550408

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FrInt															

This register indicates the current frame number. It also indicates the time remaining (interms of the number of PHY clocks) in the current (micro)frame.

Bit	Name	Mode	Initial	Type	Description
31-16	FrRem	Host	0	RO	Frame Time Remaining Indicates the amount of time remaining in the current microframe (HS) or frame (FS/LS), in terms of PHY clocks. This field decrements on each PHY clock. When it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on the USB.
15-0	FrNum	Host	16'h3FFF	RW	Frame Number This field increments when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF.

Host Periodic Transmit FIFO/Queue Status Register (HPTXSTS) 0xF0550410

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PTxQTop								PTxQSpcAvail							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PTxFSpcAvail															

This read-only register contains the free space information for the Periodic TxFIFO and the Periodic Transmit Request Queue.

Bit	Name	Mode	Initial	Type	Description
31-24	PTxQTop	Host	0	RO	Top of the Periodic Transmit Request Queue This indicates the entry in the Periodic Tx Request Queue that is currently being processes by the MAC. This register is used for debugging. • Bit [31]: Odd/Even (micro)frame - 1'b0: send in even (micro)frame - 1'b1: send in odd (micro)frame • Bits [30:27]: Channel/endpoint number • Bits [26:25]: Type - 2'b00: IN/OUT - 2'b01: Zero-length packet - 2'b10: CSPLIT - 2'b11: Disable channel command • Bit [24]: Terminate (last entry for the selected channel/endpoint)
23-16	PTxQSpcAvail	Host	0	RO	Periodic Transmit Request Queue Space Available Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. • 8'h0: Periodic Transmit Request Queue is full • 8'h1: 1 location available • 8'h2: 2 locations available • n: n locations available (0≤n≤8) • Others: Reserved
15-0	PTxFSpcAvail	Host	0	RW	Periodic Transmit Data FIFO Space Available Indicates the number of free locations available to be written to in the Periodic TxFIFO. Values are in terms of 32-bit words • 16'h0: Periodic TxFIFO is full • 16'h1: 1 word available • 16'h2: 2 words available • 16' hn: n words available (where 0≤n≤32,768) • 16'h8000: 32,768 words available • Others: Reserved

Host All Channels Interrupt Register (HAINT) 0xF0550414

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HAINT															

When a significant event occurs on a channel, the Host All Channels Interrupt register interrupts the application using the Host Channels Interrupt bit of the Core Interrupt register (GINTSTS.HChInt). This is shown in Figure 15.3. There is one interrupt bit per channel, up to a maximum of 16 bits. Bits in this register are set and cleared when the application sets and clears bits in the corresponding Host Channel-n Interrupt register.

Bit	Name	Mode	Initial	Type	Description
31-16	-		0		Reserved
15-0	HAINT	Host	0	RO	Channel Interrupts One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15

Host All Channels Interrupt Mask Register (HAINTMSK) 0xF0550418

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HAINTMsk															

The Host All Channel Interrupt Mask register works with the Host All Channel Interrupt register to interrupt the application when an event occurs on a channel. There is one interrupt mask bit per channel, up to a maximum of 16 bits.

- Mask interrupt: 1’b0
- Unmask interrupt: 1’b1

Bit	Name	Mode	Initial	Type	Description
31-16	-		0		Reserved
15-0	HAINTMsk	Host	0	R_W	Channel Interrupts Mask One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15

Host Port Control and Status Register (HPRT) 0xF0550440

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													PrtSpd		PrtTstCtl[3]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PrtTstCtl[2:0]			PrtPwr	PrtLnSts			PrtRst	PrtSusp	PrtRes	PrtOvrCurrChng	PrtOvrCurrAct	PrtEnChng	PrtEna	PrtConnDet	PrtConnSts

This register is available only in Host mode. Currently, the OTG Host supports only one port.

A single register holds USB port-related information such as USB reset, enable, suspend, resume, connect status, and test mode for each port. It is shown in Figure 4-2. The R_SS_WC bits in this register can trigger an interrupt to the application through the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). On a Port Interrupt, the application must read this register and clear the bit that caused the interrupt. For the R_SS_WC bits, the application must write a 1 to the bit to clear the interrupt.

Bit	Name	Mode	Initial	Type	Description
31-19	-		0		Reserved
18-17	PrtSpd	Host	0	RO	Port Speed () Indicates the speed of the device attached to this port. • 2'b00: High speed • 2'b01: Full speed • 2'b10: Low speed • 2'b11: Reserved
16-13	PrtTstCtl	Host	0	R_W	Port Test Control The application writes a nonzero value to this field to put the port into a Test mode, and the corresponding pattern is signaled on the port. • 4'b0000: Test mode disabled • 4'b0001: Test_J mode • 4'b0010: Test_K mode • 4'b0011: Test_SE0_NAK mode • 4'b0100: Test_Packet mode • 4'b0101: Test_Force_Enable • Others: Reserved
12	PrtPwr	Host	0	R_W_SC	Port Power The application uses this field to control power to this port, and the core clears this bit on an overcurrent condition. • 1'b0: Power off • 1'b1: Power on
11-10	PrtLnSts	Host	0	RO	Port Line Status Indicates the current logic level USB data lines • Bit [10]: Logic level of D– • Bit [11]: Logic level of D +
9	-		0		Reserved
8	PrtRst	Host	o	R_W	Port Reset When the application sets this bit, a reset sequence is started on this port. The application must time the reset period and clear this bit after the reset sequence is complete. • 1'b0: Port not in reset • 1'b1: Port in reset The application must leave this bit set for at least a minimum duration mentioned below to start a reset on the port. The application can leave it set for another 10 ms in addition to the required minimum duration, before clearing the bit, even though there is no maximum limit set by the USB standard.

					• High speed: 50 ms • Full speed/Low speed: 10 ms
7	PrtSusp	Host	0	R_WS_SC	Port Suspend The application sets this bit to put this port in Suspend mode. The core only stops sending SOFs when this is set. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the suspend input pin of the PHY. The read value of this bit reflects the current suspend status of the port. This bit is cleared by the core after a remote wakeup signal is detected or the application sets the Port Reset bit or Port Resume bit in this register or the Resume/Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.WkUpInt or GINTSTS.DisconnInt, respectively). • 1'b0: Port not in Suspend mode • 1'b1: Port in Suspend mode
6	PrtRes	Host	0	R_W_SS_SC	Port Resume The application sets this bit to drive resume signaling on the port. The core continues to drive the resume signal until the application clears this bit. If the core detects a USB remote wakeup sequence, as indicated by the Port Resume/Remote Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.WkUpInt), the core starts driving resume signaling without application intervention and clears this bit when it detects a disconnect condition. The read value of this bit indicates whether the core is currently driving resume signaling. • 1'b0: No resume driven • 1'b1: Resume driven
5	PrtOvrCurrChng	Host	0	R_SS_WC	Port Overcurrent Change The core sets this bit when the status of the Port Overcurrent Active bit (bit 4) in this register changes.
4	PrtOvrCurrAct	Host	0	RO	Port Overcurrent Active Indicates the overcurrent condition of the port. • 1'b0: No overcurrent condition • 1'b1: Overcurrent condition
3	PrtEnChng	Host	0	R_SS_WC	Port Enable/Disable Change The core sets this bit when the status of the Port Enable bit [2] of this register changes.
2	PrtEna	Host	0	R_SS_SC_WC	Port Enable A port is enabled only by the core after a reset sequence, and is disabled by an overcurrent condition, a disconnect condition, or by the application clearing this bit. The application cannot set this bit by a register write. It can only clear it to disable the port. This bit does not trigger any interrupt to the application. • 1'b0: Port disabled • 1'b1: Port enabled
1	PrtConnDet	Host	0	R_SS_WC	Port Connect Detected The core sets this bit when a device connection is detected to trigger an interrupt to the application using the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt).

					The application must write a 1 to this bit to clear the interrupt.
0	PrtConnSts	Host	0	RO	Port Connect Status • 0: No device is attached to the port. • 1: A device is attached to the port.

Host Channel-n Characteristics Register (HCCHARn) **0xF0550500+(Channel_number*0x20)**
Channel_number:0≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ChEna	ChDis	OddFrm	DevAddr							MC/ EC		EPType		LSpdDev	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EPDir	EPNum				MPS										

This register is available only in Host mode. Currently, the OTG Host supports only one port.

A single register holds USB port-related information such as USB reset, enable, suspend, resume, connect status, and test mode for each port. It is shown in Figure 4-2. The R_SS_WC bits in this register can trigger an interrupt to the application through the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). On a Port Interrupt, the application must read this register and clear the bit that caused the interrupt. For the R_SS_WC bits, the application must write a 1 to the bit to clear the interrupt.

Bit	Name	Mode	Initial	Type	Description
31	ChEna	Host	0	R_WS_SC	Channel Enable This field is set by the application and cleared by the OTG host. • 1'b0: Channel disabled • 1'b1: Channel enabled
30	ChDis	Host	0	R_WS_SC	Channel Disable The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.
29	OddFrm	Host	0	R_W	Odd Frame This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro)frame. This field is applicable for only periodic (isochronous and interrupt) transactions. • 1'b0: Even (micro)frame • 1'b1: Odd (micro)frame
28-22	DevAddr	Host	0	R_W	Device Address This field selects the specific device serving as the data source or sink.
21-20	MC/ EC	Host	0	R_W	Multi Count / Error Count When the Split Enable bit of the Host Channel-n Split Control register (HCSPLTn.SpltEna) is reset (1'b0), this field indicates to the host the number of transactions that must be executed per microframe for this endpoint. • 2'b00: Reserved This field yields undefined results. • 2'b01: 1 transaction • 2'b10: 2 transactions to be issued for this endpoint per microframe • 2'b11: 3 transactions to be issued for this endpoint per microframe When HCSPLTn.SpltEna is set (1'b1), this field indicates the number of immediate retries to be performed for a periodic split transactions on transaction errors. This field must be set to at least 2'b01.
19-18	EPType	Host	0	R_W	Endpoint Type Indicates the transfer type selected. • 2'b00: Control

					• 2'b01: Isochronous • 2'b10: Bulk • 2'b11: Interrupt
17	LSpdDev	Host	0	R_W	Low-Speed Device This field is set by the application to indicate that this channel is communicating to a low-speed device.
16	-		0		Reserved
15	EPDir	Host	0	R_W	Endpoint Direction Indicates whether the transaction is IN or OUT. • 1'b0: OUT • 1'b1: IN
14-11	EPNum	Host	0	R_W	Endpoint Number Indicates the endpoint number on the device serving as the data source or sink.
10-0	MPS	Host	0	R_W	Maximum Packet Size Indicates the maximum packet size of the associated endpoint.

Host Channel-n Split Control Register (HCSPLTn) **0xF0550504+(Channel_number*0x20)**

Channel_number:0≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SpltEna															CompSplt
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
XactPos		HubAddr							PrtAddr						

Bit	Name	Mode	Initial	Type	Description
31	SpltEna	Host	0	R_W	Split Enable The application sets this field to indicate that this channel is enabled to perform split transactions.
30-17	-		0		Reserved
16	CompSplt	Host	0	R_W	Do Complete Split The application sets this field to request the OTG host to perform a complete split transaction.
15-14	XactPos	Host	0	R_W	Transaction Position This field is used to determine whether to send all, first, middle, or last payloads with each OUT transaction. • 2'b11: All. This is the entire data payload is of this transaction (which is less than or equal to 188 bytes). • 2'b10: Begin. This is the first data payload of this transaction (which is larger than 188 bytes). • 2'b00: Mid. This is the middle payload of this transaction (which is larger than 188 bytes). • 2'b01: End. This is the last payload of this transaction (which is larger than 188 bytes).
13-7	HubAddr	Host	0	R_W	Hub Address This field holds the device address of the transaction translator's hub.
6-0	PrtAddr	Host	0	R_W	Port Address This field is the port number of the recipient transaction translator.

Host Channel-n Interrupt Register (HCINTn) 0xF0550508+(Channel_number*0x20)

Channel_number:0≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					DataTglErr	FrmOvrun	BblErr	XactErr	NYET	ACK	NAK	STALL	AHBErr	ChHltd	XferCompl

This register indicates the status of a channel with respect to USB- and AHB-related events. It is shown in Figure 15.3. The application must read this register when the Host Channels Interrupt bit of the Core Interrupt register (GINTSTS.HChInt) is set. Before the application can read this register, it must first read the Host All Channels Interrupt (HAINT) register to get the exact channel number for the Host Channel-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the HAINT and GINTSTS registers.

Bit	Name	Mode	Initial	Type	Description
31-11	-		0		Reserved
10	DataTglErr	Host	0	R_SS_WC	Data Toggle Error
9	FrmOvrun	Host	0	R_SS_WC	Frame Overrun
8	BblErr	Host	0	R_SS_WC	Babble Error
7	XactErr	Host	0	R_SS_WC	Transaction Error Indicates one of the following errors occurred on the USB. • CRC check failure • Timeout • Bit stuff error • False EOP
6	NYET	Host	0	R_SS_WC	NYET Response Received Interrupt
5	ACK	Host	0	R_SS_WC	ACK Response Received/Transmitted Interrupt
4	NAK	Host	0	R_SS_WC	NAK Response Received Interrupt
3	STALL	Host	0	R_SS_WC	STALL Response Received Interrupt
2	AHBErr	Host	0	R_SS_WC	AHB Error This is generated only in Internal DMA mode when there is an AHB error during AHB read/write. The application can read the corresponding channel's DMA address register to get the error address.
1	ChHltd	Host	0	R_SS_WC	Channel Halted Indicates the transfer completed abnormally either because of any USB transaction error or in response to disable request by the application.
0	XferCompl	Host	0	R_SS_WC	Transfer Completed Transfer completed normally without any errors.

Host Channel-n Interrupt Mask Register (HCINTMSKn) **0xF055050C+(Channel_number*0x20)**

Channel_number:0≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					DataTglErrMsk	FrmOvrunMsk	BblErrMsk	XactErrMsk	NyetMsk	AckMsk	NakMsk	StallMsk	AHBErrMsk	ChHltdMsk	XferComplMsk

This register reflects the mask for each channel status described in the previous section.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

Bit	Name	Mode	Initial	Type	Description
31-11	-		0		Reserved
10	DataTglErrMsk	Host	0	R_W	Data Toggle Error Mask
9	FrmOvrunMsk	Host	0	R_W	Frame Overrun Mask
8	BblErrMsk	Host	0	R_W	Babble Error Mask
7	XactErrMsk	Host	0	R_W	Transaction Error Mask
6	NyetMsk	Host	0	R_W	NYET Response Received Interrupt Mask
5	AckMsk	Host	0	R_W	ACK Response Received/Transmitted Interrupt Mask
4	NakMsk	Host	0	R_W	NAK Response Received Interrupt Mask
3	StallMsk	Host	0	R_W	STALL Response Received Interrupt Mask
2	AHBErrMsk	Host	0	R_W	AHB Error Mask
1	ChHltdMsk	Host	0	R_W	Channel Halted Mask
0	XferComplMsk	Host	0	R_W	Transfer Completed Mask

Host Channel-n Transfer Size Register (HCTSIZn) 0xF0550510+(Channel_number*0x20)

Channel_number:0≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DoPng	Pid		PktCnt										XferSize[18:16]		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
XferSize[15:0]															

Bit	Name	Mode	Initial	Type	Description
31	DoPng	Host	0	R_W	Do Ping () Setting this field to 1 directs the host to do PING protocol.
30-29	Pid	Host	0	R_W	PID The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer. • 2'b00: DATA0 • 2'b01: DATA2 • 2'b10: DATA1 • 2'b11: MDATA (non-control)/SETUP (control)
28-19	PktCnt	Host	0	R_W	Packet Count This field is programmed by the application with the expected number of packets to be transmitted (OUT) or received (IN). The host decreme ts this count on every successful transmission or reception of an OUT/IN packet. Once this count reaches zero, the application is interrupted to indicate normal completion.
18:0	XferSize	Host	0	R_W	Transfer Size For an OUT, this field is the number of data bytes the host sends during the transfer. For an IN, this field is the buffer size that the application has Reserved for the transfer. The application is expected to program this field as an integer multiple of the maximum packet size for IN transactions (periodic and non-periodic).

Host Channel-n DMA Address Register (HCDMAn) 0xF0550514+(Channel_number*0x20)

Channel_number:0≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DoPng	Pid		PktCnt										XferSize[18:16]		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
XferSize[15:0]															

This register is used by the OTG host in the internal DMA mode to maintain the current buffer pointer for IN/OUT transactions. The starting DMA address must be DWORD-aligned.

Bit	Name	Mode	Initial	Type	Description
31-0	DMAAddr	Host	0	R_W	DMA Address This field holds the start address in the external memory from which the data for the endpoint must be fetched or to which it must be stored. This register is incremented on every AHB transaction.

Device Configuration Register (DCFG) 0xF0550800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									EPMisCnt						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			PerFrInt		DevAddr								NZSts OUTH Shk	DevSpd	

This register configures the core in Device mode after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.

Bit	Name	Mode	Initial	Type	Description
31-23	-		0		Reserved
22-18	EPMisCnt	Device	0	R_W	IN Endpoint Mismatch Count This field is valid only in shared FIFO operation. The application programs this filed with a count that determines when the core generates an Endpoint Mismatch interrupt (GINTSTS.EPMis). The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or when the counter expires. The width of this counter depends on the depth of the Token Queue.
17-13	-		0		Reserved
12-11	PerFrInt	Device	0	R_W	Periodic Frame Interval Indicates the time within a (micro)frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro)frame is complete. • 2'b00: 80% of the (micro)frame interval • 2'b01: 85% • 2'b10: 90% • 2'b11: 95%
10-4	DevAddr	Device	0	R_W	Device Address The application must program this field after every SetAddress control command.
3	-		0		Reserved
2	NZStsOUTHShk	Device	0	R_W	Non-Zero-Length Status OUT Handshake The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage. • 1'b1: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. • 1'b0: Send the received OUT packet to the application (zerolength or nonzero-length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register.
1-0	DevSpd	Device	0	R_W	Device Speed Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. See "Device Initialization" on page 210 for details. • 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) • 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz)

					• 2'b10: Low speed (USB 1.1 transceiver clock is 6 MHz). If you select 6 MHz LS mode, you must do a soft reset. • 2'b11: Full speed (USB 1.1 transceiver clock is 48 MHz)

Device Control Register (DCTL) **0xF0550804**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				PWROnPrgDone	CGOUTNak	SGOUTNak	CGNPInNak	SGNPInNak	TstCtl			GOUTNakSts	GNPINNakSts	SftDiscon	RmtWkUpSig

Bit	Name	Mode	Initial	Type	Description
31-12	-		0		Reserved
11	PWROnPrgDone	Device	0	R_W	Power-On Programming Done The application uses this bit to indicate that register programming is completed after a wake-up from Power Down Mode.
10	CGOUTNak	Device	0	WO	Clear Global OUT NAK A write to this field clears the Global OUT NAK.
9	SGOUTNak	Device	0	WO	Set Global OUT NAK A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set the this bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register (GINTSTS.GOUTNakEff) is cleared.
8	CGNPInNak	Device	0	WO	Clear Global Non-periodic IN NAK A write to this field clears the Global Non-periodic IN NAK.
7	SGNPInNak	Device	0	WO	Set Global Non-periodic IN NAK A write to this field sets the Global Non-periodic IN NAK.The application uses this bit to send a NAK handshake on all nonperiodic IN endpoints. The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register (GINTSTS.GINNakEff) is cleared.
6-4	TstCtl	Device	0	R_W	Test Control • 3'b000: Test mode disabled • 3'b001: Test_J mode • 3'b010: Test_K mode • 3'b011: Test_SE0_NAK mode • 3'b100: Test_Packet mode • 3'b101: Test_Force_Enable • Others: Reserved
3	GOUTNakSts	Device	0	RO	Global OUT NAK Status • 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. • 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped.
2	GNPINNakSts	Device	0	RO	Global Non-periodic IN NAK Status • 1'b0: A handshake is sent out based on the data availability in the transmit FIFO. • 1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.
1	SftDiscon	Device	0	R_W	Soft Disconnect The application uses this bit to signal the otg core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears

					this bit. The minimum duration for which the core must keep this bit set is specified in Table below. • 1'b0: Normal operation. When this bit is cleared after a soft disconnect, the core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration. • 1'b1: The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host.
0	RmtWkUpSig	Device	0	R_W	Remote Wakeup Signaling When the application sets this bit, the core initiates remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1–15 ms after setting it.

Table below lists the minimum duration under various conditions for which the SoftDisconnect (SftDiscon) bit must be set for the USB host to detect a device disconnect. To accommodate clock jitter, it is recommended that the application add some extra delay to the specified minimum duration.

Table 15.3 Minimum Duration for Soft Disconnect

Operating Speed	Device Status	Minimum Duration
High Speed	Suspended	1 ms + 2.5 us
High Speed	Idle	3 ms + 2.5 us
High Speed	Not Idle or Suspended (Performing transactions)	125 us
Full speed/Low speed	Suspended	1 ms + 2.5 us
Full speed/Low speed	Idle	2.5 us
Full speed/Low speed	Not Idle or Suspended (Performing transactions)	2.5 us

Device Status Register (DSTS) 0xF0550808

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										SOFFN[13:8]					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOFFN[7:0]												ErrticErr	EnumSpd		SuspSts

This register indicates the status of the core with respect to USB-related events. It must be read on interrupts from Device All Interrupts (DAINT) register.

Bit	Name	Mode	Initial	Type	Description
31-22	-		0		Reserved
21-8	SOFFN	Device	0	RO	Frame or Microframe Number of the Received SOF When the core is operating at high speed, this field contains a microframe number. When the core is operating at full or low speed, this field contains a frame number.
7-4	-		0		Reserved
3	ErrticErr	Device	0	RO	Erratic Error The core sets this bit to report any erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted for at least 2 ms, due to PHY error) seen on the UTMI+ . Due to erratic errors, the OTG core goes into Suspended state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover.
2-1	EnumSpd	Device	2'b01	RO	Enumerated Speed Indicates the speed at which the OTG core has come up after speed detection through a chirp sequence. • 2'b00: High speed (PHY clock is running at 30 or 60 MHz) • 2'b01: Full speed (PHY clock is running at 30 or 60 MHz) • 2'b10: Low speed (PHY clock is running at 6 MHz) • 2'b11: Full speed (PHY clock is running at 48 MHz) Low speed is not supported for devices using a UTMI+ PHY.
0	SuspSts	Device	0	RO	Suspend Status In Device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspended state when there is no activity on the phy_line_state_i signal for an extended period of time. The core comes out of the suspend: • When there is any activity on the phy_line_state_i signal • When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig).

Device IN Endpoint Common Interrupt Mask Register (DIEPMSK) 0xF0550810

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

	TxfifoUndrnMsk		INEPNakEffMsk	INTknEPMisMsk	INTknTXFEmpMsk	TimeOUTMsk	AHBErrMsk	EPDisbldMsk	XferComplMsk

This register works with each of the Device IN Endpoint Interrupt (DIEPINTn) registers for all endpoints to generate an interrupt per IN endpoint. The IN endpoint interrupt for a specific status in the DIEPINTn register can be masked by writing to the corresponding bit in this register. Status bits are masked by default.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

Bit	Name	Mode	Initial	Type	Description
31-9	-		0		Reserved
8	TxfifoUndrnMsk	Device	0	R_W	Fifo Underrun Mask
7	-		0		Reserved
6	INEPNakEffMsk	Device	0	R_W	IN Endpoint NAK Effective Mask
5	INTknEPMisMsk	Device	0	R_W	IN Token received with EP Mismatch Mask
4	INTknTXFEmpMsk	Device	0	R_W	IN Token Received When TxFIFO Empty Mask
3	TimeOUTMsk	Device	0	R_W	Timeout Condition Mask (Non-isochronous endpoints)
2	AHBErrMsk	Device	0	R_W	AHB Error Mask
1	EPDisbldMsk	Device	0	R_W	Endpoint Disabled Interrupt Mask
0	XferComplMsk	Device	0	R_W	Transfer Completed Mask

Device OUT Endpoint Common Interrupt Mask Register (DOEPMSK) 0xF0550814

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							OutPkt ErrMsk		Back2 BackS ETup		OUTTk nEPdis Msk	SetUP Msk	AHBErr Msk	EPDisb ldMsk	XferCo mplMsk

This register works with each of the Device OUT Endpoint Interrupt (DOEPINTn) registers for all endpoints to generate an interrupt per OUT endpoint. The OUT endpoint interrupt for a specific status in the DOEPINTn register can be masked by writing into the corresponding bit in this register. Status bits are masked by default.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

Bit	Name	Mode	Initial	Type	Description
31-9	-		0		Reserved
8	OutPktErrMsk	Device	0	R_W	OUT Packet Error Mask
7	-		0		Reserved
6	Back2BackSETup	Device	0	R_W	Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only.
5	-		0		Reserved
4	OUTTknEPdisMsk	Device	0	R_W	OUT Token Received when Endpoint Disabled Mask Applies to control OUT endpoints only.
3	SetUPMsk	Device	0	R_W	SETUP Phase Done Mask Applies to control endpoints only.
2	AHBErrMsk	Device	0	R_W	AHB Error Mask
1	EPDisbldMsk	Device	0	R_W	Endpoint Disabled Interrupt Mask
0	XferComplMsk	Device	0	R_W	Transfer Completed Mask

Device All Endpoints Interrupt Register (DAINT) 0xF0550818

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OutEPInt															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
InEpInt															

When a significant event occurs on an endpoint, a Device All Endpoints Interrupt register interrupts the application using the Device OUT Endpoints Interrupt bit or Device IN Endpoints Interrupt bit of the Core Interrupt register (GINTSTS.OEPInt or GINTSTS.IEPInt, respectively). This is shown in Figure 4-2. There is one interrupt bit per endpoint, up to a maximum of 16 bits for OUT endpoints and 16 bits for IN endpoints. For a bidirectional endpoint, the corresponding IN and OUT interrupt bits are used. Bits in this register are set and cleared when the application sets and clears bits in the corresponding Device Endpointn Interrupt register (DIEPINTn/DOEPINTn).

Bit	Name	Mode	Initial	Type	Description
31-9	OutEPInt	Device	0	RO	OUT Endpoint Interrupt Bits One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15
0	InEpInt	Device	0	RO	IN Endpoint Interrupt Bits One bit per IN Endpoint: Bit 0 for IN endpoint 0, bit 15 for endpoint 15

Device All Endpoints Interrupt Mask Register (DAINTMSK) 0xF055081C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OutEPInt															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
InEpInt															

The Device Endpoint Interrupt Mask register works with the Device Endpoint Interrupt register to interrupt the application when an event occurs on a device endpoint. However, the Device All Endpoints Interrupt (DAINT) register bit corresponding to that interrupt is still set.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

Bit	Name	Mode	Initial	Type	Description
31-9	OutEPMsk	Device	0	RO	OUT Endpoint Interrupt Mask Bits One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15
0	InEpMsk	Device	0	RO	IN Endpoint Interrupt Mask Bits One bit per IN Endpoint: Bit 0 for IN endpoint 0, bit 15 for endpoint 15

Device VBUS Discharge Time Register (DVBUSDIS) 0xF0550828

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVBUSDis															

This register specifies the VBUS discharge time after VBUS pulsing during SRP.

Bit	Name	Mode	Initial	Type	Description
31-16	-		0		Reserved
15-0	DVBUSDis	Device	30 MHz: 16'h0B8F 60 MHz: 16'h17D7	R_W	Device VBUS Discharge Time Specifies the VBUS discharge time after VBUS pulsing during SRP. This value equals: VBUS discharge time in PHY clocks / 1, 024 The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width). Depending on your VBUS load, this value can need adjustment.

Device VBUS Pulsing Time Register (DVBUSPULSE) 0xF055082C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVBUSDis															

This register specifies the VBUS pulsing time during SRP.

Bit	Name	Mode	Initial	Type	Description
31-12	-		0		Reserved
11-0	DVBUSPulse	Device	30 MHz: 12'h2C6 60 MHz: 12'h5B8	R_W	Device VBUS Pulsing Time Specifies the VBUS pulsing time during SRP. This value equals: VBUS pulsing time in PHY clocks / 1, 024 The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width).

Device Threshold Control Register (DTHRCTL) **0xF0550830**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				ArbPrkEn		RxThrLen									RxThrEn
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					TxThrLen									ISOThrEn	NonISOThrEn

Thresholding is not supported in Slave mode and so this register should not be programmed in Slave mode. For threshold support, the AHB needs to be run at 60MHz or higher.

Bit	Name	Mode	Initial	Type	Description
31-28	-		0		Reserved
27	ArbPrkEn	Device	1	R_W	Arbiter Parking Enable This bit controls internal DMA arbiter parking for IN endpoints. When thresholding is enabled and this bit is set to one, then the arbiter parks on the IN endpoint for which there is a token received on the USB. This is done to avoid getting into underrun conditions. By default the parking is enabled.
26	-		0		Reserved
25-17	RxThrLen	Device	0	R_W	Receive Threshold Length This field specifies Receive thresholding size in DWORDS. This field also specifies the amount of data received on the USB before the core can start transmitting on the AHB. The threshold length has to be at least eight DWORDS. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).
16	RxThrEn	Device	0	R_W	Receive Threshold Enable When this bit is set, the core enables thresholding in the receive direction.
15-11	-		0		Reserved
10-2	TxThrLen	Device	0	R_W	Transmit Threshold Length This field specifies Transmit thresholding size in DWORDS. This field specifies the amount of data in bytes to be in the corresponding endpoint transmit FIFO, before the core can start transmit on the USB. The threshold length has to be at least eight DWORDS. This field controls both isochronous and non-isochronous IN endpoint thresholds. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).
1	ISOThrEn	Device	0	R_W	ISO IN Endpoints Threshold Enable. When this bit is set, the core enables thresholding for isochronous IN endpoints.
0	NonISOThrEn	Device	0	R_W	Non-ISO IN Endpoints Threshold Enable When this bit is set, the core enables thresholding for Non Isochronous IN endpoints.

Device IN Endpoint FIFO Empty Interrupt Mask Register: (DIEPEMPMSK) **0xF0550834**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
InEpTxfEmpMsk															

This register is used to control the IN endpoint FIFO empty interrupt generation (DIEPINTn.TxfEmp).

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

Bit	Name	Mode	Initial	Type	Description
31-16	-		0	RO	Reserved
15-0	InEpTxfEmpMsk	Device	0	R_W	IN EP Tx FIFO Empty Interrupt Mask Bits These bits acts as mask bits for DIEPINTn. TxFEmp interrupt One bit per IN Endpoint: Bit 0 for IN EP 0, bit 15 for IN EP 15

Device Control IN Endpoint 0 Control Register (DIEPCTL0) 0xF0550900

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEna	EPDis			SNAK	CNAK	TxFNum				Stall		EPType		NAKSts	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
USBA ctEP														MPS	

This section describes the Control IN Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1–15.

Bit	Name	Mode	Initial	Type	Description
31	EPEna	Device	0	R_WS_SC	Endpoint Enable Indicates that data is ready to be transmitted on the endpoint. The core clears this bit before setting any of the following interrupts on this endpoint: • Endpoint Disabled • Transfer Completed
30	EPDis	Device	0	R_WS_SC	Endpoint Disable The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.
29-28	-		0		Reserved
27	SNAK	Device	0	WO	Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint.
26	CNAK	Device	0	R_W	Clear NAK A write to this bit clears the NAK bit for the endpoint.
25-22	TxFNum	Device	0	R_W	TxFIFO Number This value is set to the FIFO number that is assigned to IN Endpoint 0. ********** Warning TxFIFO Number should be same as the Device IN Endpoint Number
21	Stall	Device	0	R_WS_SC	STALL Handshake The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Nonperiodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority.
20	-		0		Reserved
19-18	EPType	Device	0	RO	Endpoint Type Hardcoded to 00 for control.
17	NAKSts	Device	0	RO	NAK Status Indicates the following: • 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status • 1'b1: The core is transmitting NAK handshakes on this endpoint. When this bit is set, either by the application or core, the core stops transmitting data, even if there is data available in the TxFIFO. Irrespective of this bit's setting, the core

					always responds to SETUP data packets with an ACK handshake.
16	-		0		Reserved
15	USBActEP	Device	1	RO	USB Active Endpoint This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.
14-2	-		0		Reserved
1-0	MPS	Device	0	R_W	Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. • 2'b00: 64 bytes • 2'b01: 32 bytes • 2'b10: 16 bytes • 2'b11: 8 bytes

Device Control OUT Endpoint 0 Control Register (DOEPCTL0) 0xF0550B00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEna	EPDis			SNAK	CNAK					Stall	Snp	EPType		NAKSts	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
USBA ctEP														MPS	

This section describes the Control OUT Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1–15.

Bit	Name	Mode	Initial	Type	Description
31	EPEna	Device	0	R_WS_SC	Endpoint Enable Indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: • SETUP Phase Done • Endpoint Disabled • Transfer Completed Note: In DMA mode, this bit must be set for the core to transfer SETUP data packets into memory.
30	EPDis	Device	0	RO	Endpoint Disable The application cannot disable control OUT endpoint 0.
29-28	-		0		Reserved
27	SNAK	Device	0	WO	Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit on a Transfer Completed interrupt, or after a SETUP packet is received on the endpoint.
26	CNAK	Device	0	R_W	Clear NAK A write to this bit clears the NAK bit for the endpoint.
25-22	-		0		Reserved
21	Stall	Device	0	R_WS_SC	STALL Handshake The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
20	Snp	Device	0	R_W	Snoop Mode This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.
19-18	EPType	Device	0	RO	Endpoint Type Hardcoded to 00 for control.
17	NAKSts	Device	0	RO	NAK Status Indicates the following: • 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. • 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit, the core stops receiving data, even if there is space in the RxFIFO to accommodate the incoming packet.

					Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
16	-		0		Reserved
15	USBActEP	Device	1	RO	USB Active Endpoint This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.
14-2	-		0		Reserved
1-0	MPS	Device	0	RO	Maximum Packet Size The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. • 2'b00: 64 bytes • 2'b01: 32 bytes • 2'b10: 16 bytes • 2'b11: 8 bytes

Device Endpoint-n Control Register (DIEPCTLn/DOEPCTLn)

0xF0550900+(Endpoint_number*0x20) for IN endpoints
0xF0550B00+(Endpoint_number*0x20) for OUT endpoints
Endpoint_number:1≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEna	EPDis	SetD1PID/SetOddFr	SetD0PID/SetEvenFr	SNAK	CNAK	TxFNum				Stall	Snp	EPType		NAKSts	DPID/EO_FrNum
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
USBActEP					MPS										

The application uses this register to control the behavior of each logical endpoint other than endpoint 0.

Bit	Name	Mode	Initial	Type	Description
31	EPEna	Device	0	R_WS_SC	Endpoint Enable Applies to IN and OUT endpoints. For IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint. For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: • SETUP Phase Done (OUT only) • Endpoint Disabled • Transfer Completed Note: For control OUT endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.
30	EPDis	Device	0	R_WS_SC	Endpoint Disable Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.
29	SetD1PID	Device	0	WO	Set DATA1 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1.
	SetOddFr				Set Odd (micro)frame Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to odd (micro)frame.
28	SetD0PID	Device	0	WO	Set DATA0 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0.

	SetEvenFr				Set Even (micro)frame Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to even (micro)frame.
27	SNAK	Device	0	WO	Set NAK Applies to IN and OUT endpoints. A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.
26	CNAK	Device	0	WO	Clear NAK A write to this bit clears the NAK bit for the endpoint.
25-22	TxFNum	Device	0	R_W	TxFIFO Number These bits specify the FIFO number associated with this endpoint. Each active IN endpoint should be programmed to a separate FIFO number. This field is valid only for IN endpoints. ********** Warning TxFIFO Number should be same as the Device IN Endpoint Number
21	Stall	Device	0	R_W	STALL Handshake Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
20	Snp	Device	0	R_W	Snoop Mode Applies to OUT endpoints only. This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.
19-18	EPType	Device	0	R_W	Endpoint Type Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint. • 2'b00: Control • 2'b01: Isochronous • 2'b10: Bulk • 2'b11: Interrupt
17	NAKSts	Device	0	RO	NAK Status Applies to IN and OUT endpoints. Indicates the following: • 1'b0: The core is transmitting non-NAK

					handshakes based on the FIFO status. • 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: • The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. • For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. • For isochronous IN endpoints: The core sends out a zerolength data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
16	DPID	Device	0	RO	Endpoint Data PID Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. Applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. • 1'b0: DATA0 • 1'b1: DATA1
	EO_FrNum				Even/Odd (Micro)Frame Applies to isochronous IN and OUT endpoints only. Indicates the (micro)frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register. • 1'b0: Even (micro)frame • 1'b1: Odd (micro)frame
15	USBActEP	Device	0	R_W_SC	USB Active Endpoint Applies to IN and OUT endpoints. Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.
14-11	-		0		Reserved
10-0	MPS	Device	0	R_W	Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

Device Endpoint-n Interrupt Register (DIEPINTn/DOEPINTn)

0xF0550908+(Channel_number*0x20) for IN Endpoints
0xF0550B08+(Channel_number*0x20) for OUT Endpoints
Channel_number:0≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							TxfifoUndrn/ OutPktErr	TxFEmp	INEPNakEff /Back2BackSETup	INTknEPMis	INTknTXFEmp / OUTTknEPdis	TimeOUT /SetUp	AHBErr	EPDisbld	XferCompl

This register indicates the status of an endpoint with respect to USB- and AHB-related events. It is shown in Figure 15.3. The application must read this register when the OUT Endpoints Interrupt bit or IN Endpoints Interrupt bit of the Core Interrupt register (GINTSTS.OEPInt or GINTSTS.IEPInt, respectively) is set. Before the application can read this register, it must first read the Device All Endpoints Interrupt (DAINT) register to get the exact endpoint number for the Device Endpoint-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the DAINT and GINTSTS registers.

Bit	Name	Mode	Initial	Type	Description
31-9	-		0		Reserved
8	TxfifoUndrn	Device	0	R_SS_WC	Fifo Underrun Applies to IN endpoints Only This bit is valid only if thresholding is enabled. Core generates this interrupt when it see a transmit FIFO underrun condition for this endpoint.
	OutPktErr				OUT Packet Error Applies to OUT endpoints Only This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core sees an overflow or a CRC error for non-ISOC OUT packet.
7	TxFEmp	Device	1	RO	Transmit FIFO Empty This bit is valid only for IN Endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)).
6	INEPNakEff	Device	0	R_SS_WC	IN Endpoint NAK Effective Applies to periodic IN endpoints only. Indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit set (either by the application or by the core). This interrupt does not necessarily mean that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit.
	Back2BackSETup			R_W	Back-to-Back SETUP Packets Received Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, see page 15-379.
5	INTknEPMis	Device	0	R_SS_WC	IN Token Received with EP Mismatch Applies to non-periodic IN endpoints only.

					Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. For OUT endpoints, this bit is Reserved
4	INTknTXFEmp	Device	0	R_SS_WC	IN Token Received When TxFIFO is Empty Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received.
	OUTTknEPdis				OUT Token Received When Endpoint Disabled Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.
3	TimeOUT	Device	0	R_SS_WC	Timeout Condition Applies only to Control IN endpoints. Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint.
	SetUp				SETUP Phase Done Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.
2	AHBErr	Device	0	R_SS_WC	AHB Error Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address.
1	EPDisbld	Device	0	R_SS_WC	Endpoint Disabled Interrupt Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.
0	XferCompl	Device	0	R_SS_WC	Transfer Completed Interrupt Applies to IN and OUT endpoints. Indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.

Device Endpoint 0 Transfer Size Register (DIEPTSIZ0/DOEPTSIZ0)

0xF0550910 for IN Endpoints
0xF0550B10 for OUT Endpoints

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SUPCnt										PktCnt				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
									XferSize						

The application must modify this register before enabling endpoint 0. Once endpoint 0 is enabled using Endpoint Enable bit of the Device Control Endpoint 0 Control registers (DIEPCTL0.EPEna/DOEPCTL0.EPEna), the core modifies this register. The application can only read this register once the core has cleared the Endpoint Enable bit.

Nonzero endpoints use the registers for endpoints 1–15.

[Device IN Endpoint 0 Transfer Size Register: DIEPTSIZ0]

Bit	Name	Mode	Initial	Type	Description
31-21	-		0		Reserved
20-19	PktCnt	Device	0	R_W	Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for endpoint 0. This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO.
18-7	-		0		Reserved
6-0	XferSize	Device	0	R_W	Transfer Size Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet from the external memory is written to the TxFIFO.

[Device OUT Endpoint 0 Transfer Size Register: DOEPTSIZ0]

Bit	Name	Mode	Initial	Type	Description
31	-		0		Reserved
30-29	SUPCnt	Device	0	R_W	SETUP Packet Count This field specifies the number of back-to-back SETUP data packets the endpoint can receive. • 2'b01: 1 packet • 2'b10: 2 packets • 2'b11: 3 packets
28-20	-		0		Reserved
20-19	PktCnt	Device	0	R_W	Packet Count This field is decremented to zero after a packet is written into the RxFIFO.
18-7	-		0		Reserved
6-0	XferSize	Device	0	R_W	Transfer Size Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.

Device Endpoint-n Transfer Size Register (DIEPTSIZn/DOEPTSIZn)

0xF0550910+(Endpoint_number*0x20) for IN Endpoints
0xF0550B10+(Endpoint_number*0x20) for OUT Endpoints
Endpoint_number:1≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MC/ RxDPID/ SUPCnt		PktCnt										XferSize[18:16]		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
XferSize[15:0]															

The application must modify this register before enabling the endpoint. Once the endpoint is enabled using Endpoint Enable bit of the Device Endpoint-n Control registers (DIEPCTLn.EPEna/DOEPCTLn.EPEna), the core modifies this register. The application can only read this register once the core has cleared the Endpoint Enable bit.

This register is used only for endpoints other than Endpoint 0.

Bit	Name	Mode	Initial	Type	Description
31	-		0		Reserved
30-29	MC	Device	0	R_W	Multi Count Applies to IN endpoints only. For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints. • 2'b01: 1 packet • 2'b10: 2 packets • 2'b11: 3 packets
				RO	For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core should fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-n Control register (DIEPCTLn.NextEp).
	RxDPID			RO	Received Data PID Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint. • 2'b00: DATA0 • 2'b01: DATA2 • 2'b10: DATA1 • 2'b11: MDATA
	SUPCnt			R_W	SETUP Packet Count Applies to control OUT Endpoints only. This field specifies the number of back-to-back SETUP data packets the endpoint can receive. • 2'b01: 1 packet • 2'b10: 2 packets • 2'b11: 3 packets
28-19	PktCnt	Device	0	R_W	Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint. • IN Endpoints: This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO. • OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.
18-0	XferSize	Device	0	R_W	Transfer Size This field contains the transfer size in bytes for the current endpoint. The core only

					interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. • IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO. • OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.

Device Endpoint-n DMA Address Register (DIEPDMAn/DOEPDMAn)

0xF0550914+(Endpoint_number*0x20) for IN Endpoints
0xF0550B14+(Endpoint_number*0x20) for OUT Endpoints
Endpoint_number:0≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAAddr[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAAddr[15:0]															

Bit	Name	Mode	Initial	Type	Description
31	DMAAddr	Device	0	R_W	DMA Address Holds the start address of the external memory for storing or fetching endpoint data. This register is incremented on every AHB transaction. Application can give only a DWORD-aligned address. Note: For control endpoints, this address stores control OUT data packets as well as SETUP transaction data packets. If more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten.

Device IN Endpoint Transmit FIFO Status Register (DTXFSTSn)

0xF0550918+(Endpoint_number*0x20) for IN Endpoints

Endpoint_number:0≤n≤15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPTxFSpcAvail															

This read-only register contains the free space information for the Device IN endpoint TxFIFO.

Bit	Name	Mode	Initial	Type	Description
31-16	-		0		Reserved
15-0	INEPTxFSpcAvail	Device	0	R_W	IN Endpoint TxFIFO Space Avail Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. • 16'h0: Endpoint TxFIFO is full • 16'h1: 1 word available • 16'h2: 2 words available • 16' hn: n words available (where 0≤n≤ 32,768) • 16'h8000: 32,768 words available • Others: Reserved

Power and Clock Gating Control Register (PCGCCTL) 0xF0550E00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPTxFSpcAvail															

This register is available in Host and Device modes. The application can use this register to control the core's power-down and clock gating features. Because the CSR module is turned off during power-down, this registers is implemented inthe AHB Slave BIU module.

Bit	Name	Mode	Initial	Type	Description
31-5	-		0		Reserved
4	PhySuspended	Host and Device	0	RO	PHY Suspended. Indicates that the PHY has been suspended. After the application sets the Stop Pclk bit (bit 0), this bit is updated once the PHY is suspended. Because the UTMI+ PHY suspend is controlled through a port, the UTMI+ PHY is suspended immediately after Stop Pclk is set.
3	RstPdwnModule	Host and Device	0	R_W	Reset Power-Down Modules This bit is valid only in Partial Power-Down mode. The application sets this bit when the power is turned off. The application clears this bit after the power is turned on and the PHY clock is up.
2	PwrClmp	Host and Device	0	R_W	Power Clamp The application sets this bit before the power is turned off to clamp the signals between the poweron modules and the power-off modules. The application clears the bit to disable the clamping before the power is turned on.
1	GateHclk	Host and Device	0	R_W	Gate Hclk The application sets this bit to gate hclk to modules other than the AHB Slave and Master and wakeup logic when the USB is suspended or the session is not valid. The application clears this bit when the USB is resumed or a new session starts.
0	StopPclk	Host and Device	0	R_W	Stop Pclk The application sets this bit to stop the PHY clock (phy_clk) when the USB is suspended, the session is not valid, or the device is disconnected. The application clears this bit when the USB is resumed or a new session starts.

USB OTG Configuration Register (OTGCR) **0xF05F5000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												OTGAMAP		ENDS	ENDM

BIT	Name	RW	Reset	Description
3-2	OTGAMAP	R/W	2'b00	Fixed value of address[17:16] into USB OTG link slave interface
1	ENDS	R/W	0	Endian of slave interface of USB OTG link 0: little, 1: big
0	ENDM	R/W	0	Endian of master interface of USB OTG link 0: little, 1: big

15.3 Register Description for UTMI (USB PHY)

USB PHY Configuration Register0 (UPCR0) **0xF05F5028**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	PR	CM	RCS		RCD		SDI	FO	VBDS	DMPD	DPPD	TBSH	TBS	VBD	LBE

BIT	Name	RW	Reset	Description
15	MODE	R/W	0	Select control source over several input signals(OPMODE, TM, XCVRSEL, DPPD, DMPD, FO, txvalid, txvalidh, suspend) of USB PHY 0 The USBOTG host link signals controls the USB PHY 1: The value of UPCR0 and USB1.1 host signals control the USB PHY
14	PR	R/W	0	When asserted, this customer-specific signal resets the corresponding port's transmit and receive logic without disabling the clocks within the USB 2.0 nanoPHY. 0: The transmit and receive FSMs are operational, and the line_state logic becomes sequential after 11 PHYCLOCK cycles. 1: The transmit and receive finite state machines are reset, and the line_state logic combinationally reflects the state of the single-ended receivers.
13	CM	R/W	1	This signal controls the power-down signals in the XO, Bias, and PLL blocks when the USB 2.0 nanoPHY is suspended. 0: The XO, Bias, and PLL Blocks remains powered in suspend mode. 1: The XO, Bias, and PLL Blocks are powered down in suspend mode.
12-11	RCS	R/W	1	This signal selects the reference clock source for the PLL block. RCS: Common Block Power Down Control 11: The PLL uses CLKCORE as reference 10: The PLL uses CLKCORE as reference 01: The XO block uses an external clock supplied on the XO pin(default) 00: The XO block uses the clock from a crystal
10-9	RCD	R/W	0	This signal selects the reference clock source for the PLL block. RCD: Reference Clock Frequency Select 11: Reserved 10: 48 MHz 01: 24 MHz 00: 12 MHz

<table>
<tr><td>8</td><td>SDI</td><td>R/W</td><td>1</td><td>This test signal enables you to perform IDDQ testing by powering down all analog blocks.
<table><tr><th>SID</th><th>IDDQ Test Enable</th></tr><tr><td>1</td><td>The analog blocks are powered down</td></tr><tr><td>0</td><td>The analog blocks are not powered down.</td></tr></table></td></tr>
<tr><td>7</td><td>FO</td><td>R/W</td><td>0</td><td>This controller signal enables the UTMI or serial interface.
** It is effective only when UPCR0.MODE is set and you should not clear this bit when UPCR0.MODE is set
<table><tr><th>FO</th><th>UTMI/Serial Interface Select</th></tr><tr><td>1</td><td>The TXENABLEN, FSDATAEXT, and FSSE0EXT inputs drive USB 2.0 nanoPHY data output onto the D+ and D- lines</td></tr><tr><td>0</td><td>Data on the D+ and D- lines is transmitted and received through the UTMI.</td></tr></table></td></tr>
<tr><td>6</td><td>VBDS</td><td>R/W</td><td>0</td><td>This controller signal enables the UTMI or serial interface. This signal selects the VBUSVLDEXT input or the internal Session Valid comparator to indicate when the VBUS signal on the USB cable is valid
<table><tr><th>VBDS</th><th>External VBUS Valid Select</th></tr><tr><td>1</td><td>The VBUSVLDEXT input is used</td></tr><tr><td>0</td><td>The internal Session Valid comparator is used</td></tr></table></td></tr>
<tr><td>5</td><td>DMPD</td><td>R/W</td><td>1</td><td>This controller signal enables or disables the pull-down resistance on the D-line.
When an A/B device is acting as a host (Downstream-facing port), DPPULLDOWN and DMPULLDOWN are enabled. UTMI+/OTG-compliant controllers are not allowed to toggle DPPULLDOWN and DMPULLDOWN during normal operation. USB PHY Configuration Register2 (UPCR2).OPMODE = 2’b01(non-driving) has precedence over the DPPULLDOWN and DMPULLDOWN controls.
** It is effective only when UPCR0.MODE is set.
<table><tr><th>DMPD</th><th>D- Pull-Down Resistor Enable</th></tr><tr><td>1</td><td>The pull-down resistance on D- is enabled.</td></tr><tr><td>0</td><td>The pull-down resistance on D- is disabled.</td></tr></table></td></tr>
<tr><td>4</td><td>DPPD</td><td>R/W</td><td>1</td><td>This controller signal enables or disables the pull-down resistance on the D+line.
** It is effective only when UPCR0.MODE is set.
<table><tr><th>DPPD</th><th>D+ Pull-Down Resistor Enable</th></tr><tr><td>1</td><td>The pull-down resistance on D+ is enabled</td></tr><tr><td>0</td><td>The pull-down resistance on D+ is disabled</td></tr></table></td></tr>
<tr><td>3</td><td>TBSH</td><td>R/W</td><td>0</td><td>This controller signal enables or disables bits stuffing on DATAIN[7:0] when OPMODE[1:0] =2’b11
<table><tr><th>TBSH</th><th>Low-Byte Transmit Bit-Stuffing Enable</th></tr><tr><td>1</td><td>Bit stuffing is enabled</td></tr><tr><td>0</td><td>Bit stuffing is disabled.</td></tr></table></td></tr>
<tr><td>2</td><td>TBS</td><td>R/W</td><td>0</td><td>This controller signal enables or disables bits stuffing on DATAIN[7:0] when OPMODE[1:0] =2’b11
<table><tr><th>TBS</th><th>Low-Byte Transmit Bit-Stuffing Enable</th></tr><tr><td>1</td><td>Bit stuffing is enabled</td></tr><tr><td>0</td><td>Bit stuffing is disabled</td></tr></table></td></tr>
<tr><td>1</td><td>VBD</td><td>R/W</td><td>0</td><td>This signal is valid in device mode only when the VBUSVLDEXTSEL signal is high. VBUSVLDEXTSEL indicates whether the VBUS signal on the USB cable is valid. In addition, VBUSVLDEXTSEL enables the pull-up resistor on the D+ line.</td></tr>
</table>

				VBD / **External VBUS Valid Indicator** 1 / The VBUS signal is valid, and the pull-up resistor on D+ is enabled 0 / The VBUS signal is not valid, and the pull-up resistor on D+ is disabled.
0	LBE	R/W	0	This controller signal places the USB 2.0 nanoPHY in Loopback mode, which enables the receive and transmit logic concurrently. **LBE** / **Loopback Test Enable** 1 / During data transmission, the receive logic is enabled. 0 / During data transmission, the receive logic is disabled.

[VBUS Valid and End Interrupt]

VBUS is the USB power supply pin. When in device mode, an off-chip charge pump must provide power to VBUS pin. Alternatively VBUS can be set internally by VBUSVLDEXT and VBUSVLDEXTSEL in USB PHY Configuration Register 0(UPCR0). If VBUSVLDEXTSEL is set to 1, VBUSVLDEXT value can be supplied as VBUS data of USB PHY.

There are two interrupts related to VBUS and these interrupts, which indicate the state of VBUS, are generated from B_VALID and SESSEND signals. B_VALID signal is set to 1 when VBUS is 1 and SESSEND signal is set to 1 when VBUS is 0. Therefore, when VBUS value changes, one of two interrupts definitely occurs

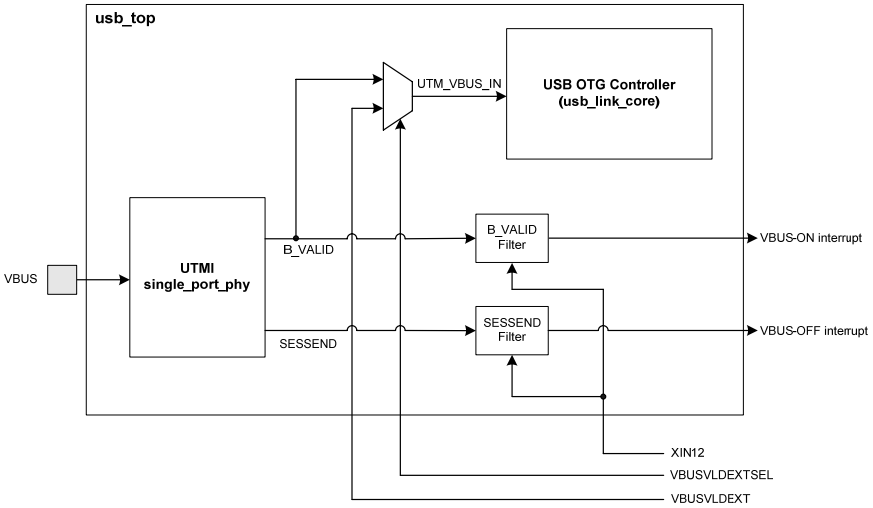


Figure 15.4 VBUS Control and Interrupts

USB PHY Configuration Register1 (UPCR1) 0xF05F502C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXFSLST					SQRXT				OTGT				CDT		

TXFSLST	[15:12]	FS/LS Pull-Up Resistance Adjustment
1111	-2.5%	
0111	Design default (default)	
0011	+2.5%	
0001	+5%	
0000	+7.5%	
Others	Reserved	

This bus adjusts the low- and full-speed pull-up resistance, based on nominal power, voltage, and temperature.

SQRXT	[10:8]	Squelch Threshold Tune
111	-20%	
110	-15%	
101	-10%	
100	-5%	
011	Design default (default)	
010	+5%	
001	+10%	
000	+15%	
Others	Reserved	

This bus adjusts the voltage level for the threshold used to detect valid high-speed data.

OTGT	[6:4]	VBUS Valid Threshold Adjustment
111	+9%	
110	+6%	
101	+3%	
100	Design default(default)	
011	-3%	
010	-6%	
001	-7%	
000	-12%	
Others	Reserved	

This bus adjusts the voltage level for the VBUS valid threshold.

CDT	[2:0]	Disconnect Threshold Adjustment
111	+6%	
110	+4.5%	
101	+3%	
100	+1.5%	
011	Design default (default)	
010	-1.5%	
001	-3%	
000	-4.5%	

This bus adjusts the voltage level for the threshold used to detect a disconnect event at the host.

USB PHY Configuration Register2 (UPCR2) 0xF05F5030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	TM	XCVRSEL			OPMODE		TXVRT					TXRT			TP

TM	[14]	USB Termination Select
1	Full Speed termination is enabled (default).	
0	High Speed termination is enabled.	

This controller signal selects FS or HS termination.

** It is effective only when UPCR0.MODE is set.

XCVRSEL	[13:12]	FS/LS Pull-Up Resistance Adjustment
11	Sends an LS packet on an FS bus or receives an Ls packet	
10	LS transceiver	
01	FS transceiver (default)	
00	HS transceiver	

This controller bus selects the HS, FS, or LS transceiver.

** It is effective only when UPCR0.MODE is set.

OPMODE	[10:9]	UTMI Operational Mode
11	Normal operation SYNC or EOP generation	
10	Disable bit stuffing and NRZI encoding	
01	Non driving	
00	Normal(default)	

This controller bus selects the UTMI operational mode.

** It is effective only when UPCR0.MODE is set.

TXVRT	[8:5]	HS DC Voltage Level Adjustment
1111	+8.75%	
1110	+7.5%	
1101	+6.25%	
1100	+5%	
1011	+3.75%	
1010	+2.5%	
1001	+1.25%	
1000	Design default (default)	
0111	-1.25%	
0110	-2.5%	
0101	-3.75%	
0100	-5%	
0011	-6.25%	
0010	-7.5%	
0001	-8.75%	
0000	-10%	

This bus adjusts the voltage to which the high speed DC level is tuned.

TXRT	[3:2]	HS Transmitter Rise/Fall Time Adjustment
01	-8%	
00	Design default (default)	
Others	Reserved	

This bus adjusts the rise/fall times the high speed waveform.

TP	[0]	HS transmitter Pre-Emphasis Enable
1	The HS Transmitter pre-emphasis is enabled	
0	The HS transmitter pre-emphasis is disabled. (default)	

This signal enables or disables the pre-emphasis for a J-K or K-J state transition in HS mode.

USB PHY Configuration Register3 (UPCR3) 0xF05F5034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCALE		OTG									TXHSXVT		TXF	ABE

SCALE	[14:13]	Scale down mode
0	This is enabled during simulation only.	

OTG	[12]	OTG Disable
1	The OTG block is powered down.	
0	The OTG block is not powered down(default)	

This controller signal powers down the OTG block, including the VBUS Valid comparator. If the application does not use OTG functionality, you can set this input high to save power.

TXHSXVT	[3:2]	Transmitter High Speed Crossover Adjustment
3	The crossover voltage is increased by 15mV	
2	The crossover voltage is increased by 30mV	
1	Default setting	
0	Reserved	

TXHSXVT adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode

TXF	[1]	TXF Mode
1	Synchronize txf empty signal	
0	Do not synchronize txf empty signal	

ABE	[0]	TX Jitter Adjustment
1	TX Jitter Fix enable (default)	
0	TX Jitter Fix disable	

15.4 Programming Model

15.4.1 Overview

This chapter describes the programming requirements for the otg core in Host and Device modes. Each significant programming feature of the otg core is discussed in a separate section.

15.4.2 Core Initialization

The application must perform the core initialization sequence following the configuration parameters the otg core. If the cable is connected during powerup, the Current Mode of Operation bit in the Core Interrupt register (GINTSTS.CurMod) reflects the mode. The otg core enters Host mode when an “A” plug is connected, or Device mode when a “B” plug is connected.

This section explains the initialization of the otg core after power-on. The application must follow the initialization sequence irrespective of Host or Device mode operation. All core global registers are initialized according to the core’s configuration.

1. Read the User Hardware Configuration registers (GHWCFG1, 2, 3, and 4) to find the configuration parameters selected for otg core.

2. Program the following fields in the Global AHB Configuration (GAHBCFG) register.

- DMA Mode bit
- AHB Burst Length field
- Global Interrupt Mask bit = 1
- RxFIFO Non-Empty (GINTSTS.RxFLvl) (applicable only when the core is operating in Slave mode)
- non-periodic TxFIFO Empty Level (can be enabled only when the core is operating in Slave mode as a host)
- Periodic TxFIFO Empty Level (can be enabled only when the core is operating in Slave mode)

3. Program the following fields in GUSBCFG register.

- HNP Capable bit
- SRP Capable bit
- External HS PHY
- UTMI+ Selection bit
- PHY Interface bit
- HS/FS TimeOUT Time-Out Calibration field

- USB Turnaround Time field

4. The software must unmask the following bits in the GINTMSK register.

- OTG Interrupt Mask
- Mode Mismatch Interrupt Mask

5. If the GUID register is selected for implementation, the software has the option of programming this register.

6. The software can read the GINTSTS.CurMod bit to determine whether the otg core is operating in Host or Device mode. The software the follows either the “Host Initialization” or “Device Initialization” sequence.

15.4.3 Host Initialization

To initialize the core as host, the application must perform the following steps.

1. Program GINTMSK.PrtInt to unmask.

2. Program the HCFG register to select full-speed host or high-speed host.

3. Program the HPRT.PrtPwr bit to 1’b1. This drives VBUS on the USB.

4. Wait for the HPRT0.PrtConnDet interrupt. This indicates that a device is connect to the port.

5. Program the HPRT.PrtRst bit to 1’b1. This starts the reset process.

6. Wait at least 10 ms for the reset process to complete.

7. Program the HPRT.PrtRst bit to 1’b0.

8. Wait for the HPRT.PrtEnChng interrupt.

9. Read the HPRT.PrtSpd field to get the enumerated speed.

10. Program the HFIR register with a value corresponding to the selected PHY clock[19].

11. Program the RXFSIZE register to select the size of the receive.

12. Program the NPTXFSIZE register to select the size and the start address of the Nonperiodic Transmit FIFO for non-periodic.

13. Program the HPTXFSIZ register to select the size and start address of the Periodic.

Transmit FIFO for periodic transactions.

To communicate with devices, the system software must initialize and enable at least one channel.

15.4.4 Device Initialization

The application must perform the following steps to initialize the core as a device on powerup or after a mode change from Host to Device.

1. Program the following fields in the DCFG register.

- Device Speed
- Non-Zero-Length Status OUT Handshake
- Periodic Frame Interval (when periodic endpoints are supported)

2. Program the Device threshold control register. This is required only if you are using DMA mode and you are planning to enable thresholding.

3. Program the GINTMSK register to unmask the following interrupts.

- USB Reset

[19] At this point, the host is up and running and the port register begins to report device disconnects, etc. The port is active with SOFs occurring down the enabled port.

- Enumeration Done
- Early Suspend
- USB Suspend
- SOF

4. Wait for the GINTSTS.USBReset interrupt, which indicates a reset has been detected on the USB and lasts for about 10 ms. On receiving this interrupt, the application must perform the steps listed in “Initialization on USB Reset”.

5. Wait for the GINTSTS.EnumerationDone interrupt. This interrupt indicates the end of reset on the USB. On receiving this interrupt, the application must read the DSTS register to determine the enumeration speed and perform the steps listed in “Initialization on Enumeration Completion”. At this point, the device is ready to accept SOF packets and perform control transfers on control endpoint 0.

15.5 Modes of Operation

The application can operate the core either in DMA mode, where the core fetches the data to be transmitted or updates the received data on the AHB, or in Slave mode, where the application initiates transfers for data fetch and store. The application cannot operate the core using DMA and Slave modes simultaneously.

15.5.1 DMA Mode

With an internal DMA option, the otg host uses the AHB Master interface for transmit packet data fetch (AHB to USB) and receive data update (USB to AHB). The AHB Master uses the programmed DMA address (HCDMAn register in Host mode and DIEPDMAn/DOEPDMAn register in Device mode) to access the data buffers.

15.5.1.1 Transfer-Level Operation

In DMA mode, the application is interrupted only after the programmed transfer size istransmitted or received (provided the otg core detects no NAK/NYET/Timeout/Error response in Host mode, or Timeout/CRC Error in Device mode). The application must handle all transaction errors. In Device mode, all the USB errors are handled by the core itself

15.5.1.2 Transaction-Level Operation

This mode is similar to transfer-level operation, with the programmed transfer size equal to one packet size (maximum size or short packet).

15.5.2 Slave Mode

In Slave mode, the application can operate the otg core either in transaction-level (packet-level) operation or in pipelined transaction-level operation.

15.5.2.1 Transaction-Level Operation

The application handles one data packet at a time per channel/endpoint in transaction-level operations. Based on the handshake response received on the USB, the application determines whether to retry the transaction or proceed with the next, until the end of the transfer. The application is interrupted on completion of every packet. The application performs transaction-level operations for a channel/endpoint for a transmission (host: OUT/device: IN) or reception (host: IN/device: OUT).

[Host Mode]

For an OUT transaction, the application enables the channel and writes the data packet into the corresponding (Periodic or Non-periodic) transmit FIFO. The otg core automatically writes the channel number into the corresponding (Periodic or Non-periodic) Request Queue, along with the last DWORD write of the packet.

For an IN transaction, the application enables the channel and the otg core automatically writes the channel number into the corresponding Request queue. The application must wait for the packet received interrupt, then empty the packet from the receive FIFO.

[Device Mode]

For an IN transaction, the application enables the endpoint, writes the data packet into the corresponding transmit FIFO, and waits for the packet completion interrupt from the core.

For an OUT transaction, the application enables the endpoint, waits for the packet received interrupt from the core, then

empties the packet from the receive FIFO.

Note

The application has to finish writing one complete packet before switching to a different channel/endpoint FIFO. Violating this rule results in an error.

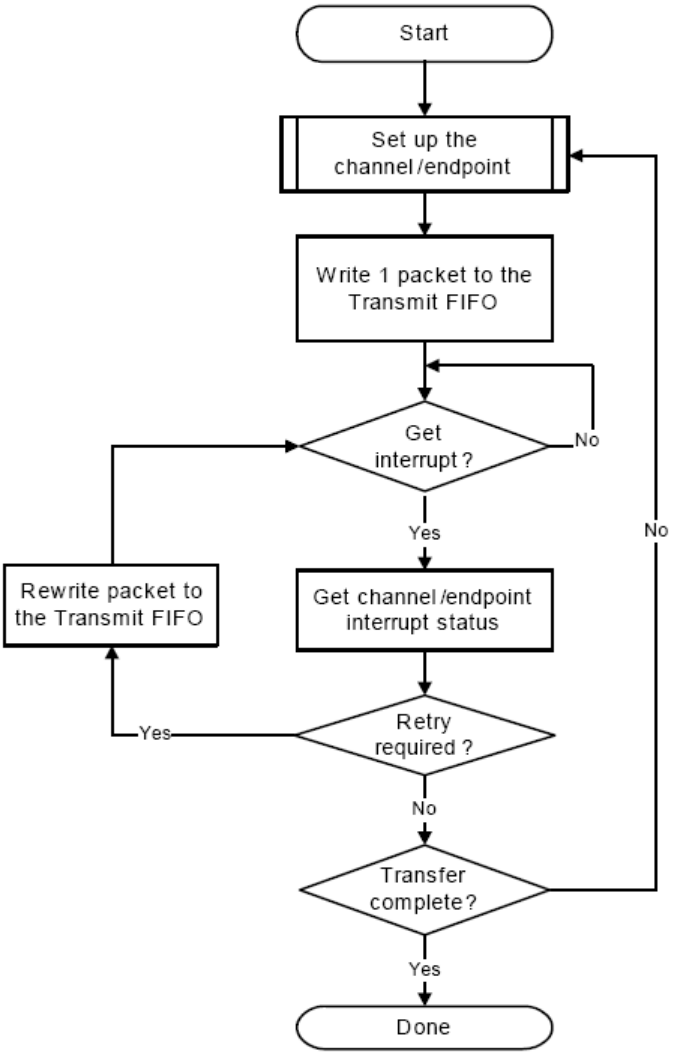


Figure 15.5 Transmit Transaction-Level Operation in Slave Mode

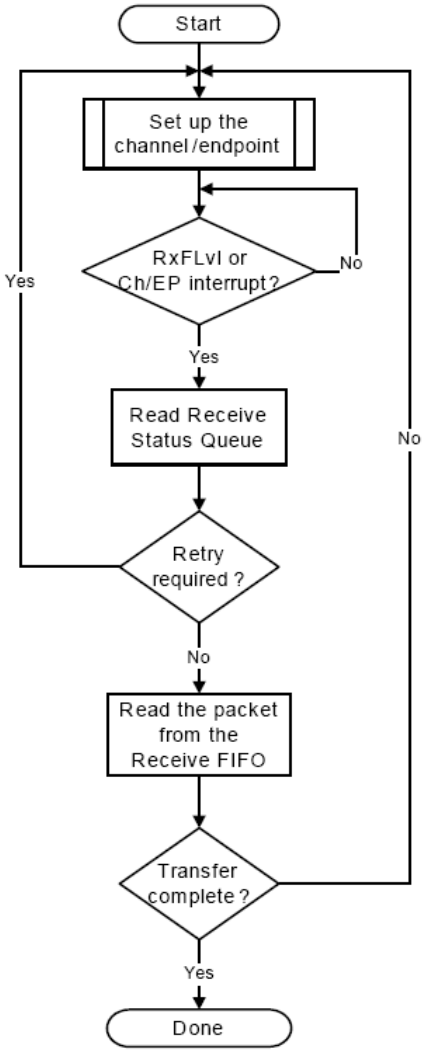


Figure 15.6 Receive Transaction-Level Operation in Slave Mode

15.5.2.2 Pipelined Transaction-Level Operation

The application can pipeline more than one transaction (IN or OUT) with pipelined transaction-level operation, which is analogous to Transfer mode in DMA mode. In pipelined transaction-level operation, the application can program the core to perform multiple transactions. The advantage of this mode of operation compared to transactionlevel operation is that the application is not interrupted on packet basis.

[Host Mode]

For an OUT transaction, the application sets up a transfer and enables the channel. The application can write multiple packets back-to-back for the same channel into the transmit FIFO, based on the space availability. It can also pipeline OUT transactions for multiple channels by writing into the HCHARn register, followed by a packet write to that channel. The core writes the channel number, along with the last DWORD write for the packet, into the Request queue and schedules transactions on the USB in the same order.
For an IN transaction, the application sets up a transfer and enables the channel, and the otg core writes the channel number into the Request queue. The application can schedule IN transactions on multiple channels, provided space is available in the Request queue. The core initiates an IN token on the USB only when there is enough space to receive at least of one maximum-packet-size packet of the channel in the top of the Request queue.

[Device Mode]

For an IN transaction, the application sets up a transfer and enables the endpoint. The application can write multiple packets back-to-back for the same endpoint into the transmit FIFO, based on available space. It can also pipeline IN transactions for multiple channels by writing into the DIEPCTLn register followed by a packet write to that endpoint. The core writes the endpoint number, along with the last DWORD write for the packet into the Request queue. The core transmits the data in the transmit FIFO when an IN token is received on the USB.

For an OUT transaction, the application sets up a transfer and enables the endpoint. The core receives the OUT data into the receive FIFO, when it has available space. As the packets are received into the FIFO, the application must empty data from it.

From this point on in this chapter, the terms “Pipelined Transaction mode” and “Transfe mode” are used interchangeably.

15.5.3 Thresholding in DMA Mode

The application can program the core to do FIFO thresholding when operating as a device in DMA mode. With threshold support, the core can be configured to operate with less than maximum packet size FIFOs for a particular endpoint. This results in a smaller FIFO requirement when compared to non-thresholding mode.
FIFO thresholding is supported only in DMA mode. FIFO thresholding is not supported when the core is operating as a host, even in DMA mode.

- The core allows both receive and transmit FIFO thresholding.
- Device Threshold Control Register bit DTHRCTL.RxThrRn must be set to enable receive thresholding. DTHRCTL.RxThrLen specifies the receive threshold size.
- Transmit uses separate Threshold Enable controls for isochronous and non-isochronous endpoints . Bits DTHRCTL.NonISOThrEn and DTHRCTL.ISOThrEn specify these Threshold Enable controls.
- The register filed DTHRCTL.TxThrLen specifies the transmit threshold length and is common for isochronous and non-isochronous endpoints. The minimum threshold length supported by the core is four DWORDs.
- Threshold enable controls cannot be changed randomly. The application can set or reset the threshold enable bits only after ensuring that the core is not programmed to do any transfers (FIFOs are flushed, NAK bits are set, all endpoints are disabled).
- One of the limitation of thresholding mode is in ping protocol, and could violate PING protocol. A PING token will be responded with an ACK handshake and the following OUT token could result in a NAK handshake. This behavior is a result of receive fifo overflow, and cannot be avoided in thresholding mode. This scenario will not occur ifthere are no overflows.

When transmit thresholding is enabled, the core starts transmitting data on the USB for a particular endpoint when there is threshold amount of data available in the corresponding transmit FIFO.

When receive thresholding is enabled, the core starts transferring data from the receive FIFO to the system memory as soon as there is threshold amount of data available in the receive FIFO. Any underrun or overflow conditions are handled by the core internally.

15.6 Host Programming Model

15.6.1 Channel Initialization

The application must initialize one or more channels before it can communicate with connected devices. To initialize and enable a channel, the application must perform the following steps.

1. Program the GINTMSK register to unmask the following:

- Channel Interrupt
- Non-periodic Transmit FIFO Empty for OUT transactions (applicable for Slave mode that operates in pipelined transaction-level with the Packet Count field programmed with more than one).
- Non-periodic Transmit FIFO Half-Empty for OUT transactions (applicable for Slave mode that operates in pipelined transaction-level with the Packet Count field programmed with more than one).

2. Program the HAINTMSK register to unmask the selected channels' interrupts.

3. Program the HCINTMSK register to unmask the transaction-related interrupts of interest given in the Host Channel Interrupt register.

4. Program the selected channel's HCTSIZn register with the total transfer size, in bytes, and the expected number of packets, including short packets. The application must program the PID field with the initial data PID (to be used on the first OUT transaction or to be expected from the first IN transaction).

5. Program the selected channels' HCSPLTn register(s) with the hub and port addresses (split transactions only).

6. Program the selected channels' HCDMAn register(s) with the buffer start address.

7. Program the HCCHARn register of the selected channel with the device's endpoint characteristics, such as type, speed, direction, and so forth. (The channel can be enabled by setting the Channel Enable bit to 1'b1 only when the application is ready to transmit or receive any packet).

Repeat steps 1–7 for other channels.

15.6.2 Halting a Channel

The application can disable any channel by programming the HCCHARn register with the HCCHARn.ChDis and HCCHARn.ChEna bits set to 1'b1. This enables the OTG host to flush the posted requests (if any) and generates a Channel Halted interrupt. The application must wait for the HCINTn.ChHltd interrupt before reallocating the channel for other transactions. The OTG host does not interrupt the transaction that has been already started on USB.
In Slave mode operation, before disabling a channel, the application must ensure that there is at least one free space available in the Non-periodic Request Queue (when disabling a nonperiodic channel) or the Periodic Request Queue (when disabling a periodic channel). The application can simply flush the posted requests when the Request queue is full (before disabling the channel), by programming the HCCHARn register with the HCCHARn.ChDis bit set to 1'b1, and the HCCHARn.ChEna bit reset to 1'b0.

To disable a channel in DMA mode operation, the application need not check for space in the Request queue. The OTG host checks for space in which to write the Disable request on the disabled channel's turn during arbitration. Meanwhile, all posted requests are dropped from the Request queue when the HCCHARn.ChDis bit is set to 1'b1.

The application is expected to disable a channel on any of the following conditions:

1. When a HCINTn.XferCompl interrupt is received during a non-periodic IN transfer or
high-bandwidth interrupt IN transfer (Slave mode only)

2. When a HCINTn.STALL, HCINTn.XactErr, HCINTn.BblErr, or HCINTn.DataTglErr interrupt is received for an IN or OUT channel (Slave mode only). For high-bandwidth interrupt INs in Slave mode, once the application has received a DataTglErr interrupt it must disable the channel and wait for a Channel Halted interrupt. The application must be able to receive other interrupts (DataTglErr, Nak, Data, XactErr, BabbleErr) for the same channel before receiving the halt.

3. When a GINTSTS.DisconnInt (Disconnect Device) interrupt is received. (The application is expected to disable all enabled channels in Slave and DMA modes)

4. When the application aborts a transfer before normal completion (Slave and DMA modes).

15.6.3 Ping Protocol

When the OTG host operates in high speed, the application must initiate the ping protocol when communicating with high-speed bulk or control (Data and Status stage) OUT endpoints. The application must initiate the ping protocol when it receives a NAK/NYET/XactErr interrupt. When the otg host receives one of the above responses, it does not continue any transaction for a specific endpoint, drops all posted or fetched OUT requests (from the Request queue), and flushes the corresponding data (from the transmit FIFO).

In Slave mode, the application can send a ping token either by setting the HCTSIZn.DoPng bit before enabling the channel or by just writing the HCTSIZn register with DoPng bit set when the channel is already enabled. This enables the otg host to write a ping request entry to the Request queue. The application must wait for the response to the ping token (a NAK, ACK, or XactErr interrupt) before continuing the transaction or sending another ping token. The application can continue the data transaction only after receiving an ACK from the OUT endpoint for the requested ping. The channel-specific interrupt service routine for the ping protocol in Slave mode is shown in table below.

In DMA mode operation, the application can start a ping protocol transfer by setting the HCTSIZn.DoPng bit before enabling the channel. The otg host continues sending ping tokens until receiving an ACK, then switches automatically to the data transaction.

Table 15.4 Interrupt Service Routine for Ping Protocol in Slave Mode

Ping Protocol for High Speed Bulk/Control OUT Endpoints

Unmask (ACK/NAK/XactErr/ChHltd/STALL)
if (ACK)
{
Reset Error Count
Re-initialize Channel (Do data transactions)
{
else if (NAK)
{
Reset Error Count
Send Ping
}
else if (STALL)
{
Disable Channel
}
else if (XactErr)
{
Increment Error Count
if (Error_count < 3)
{
Send Ping
}
else
{
Disable Channel
}
}
else if (ChHltd)
{
De-allocate Channel
}

15.6.4 Sending a Zero-Length Packet

To send a zero-length data packet, the application must initialize an OUT channel as follows.

1. Program the HCTSIZn register of the selected channel with a correct PID, XferSize = 0, and PktCnt = 1.

2. Program the HCCHARn register of the selected channel with ChEna = 1 and the device's endpoint characteristics, such as type, speed, and direction.

The application must treat a zero-length data packet as a separate transfer, and cannot combine it with a non-zero-length

transfer.

15.6.5 Operational Model

The application must initialize a channel as described in “Channel Initialization” before communicating to the connected device. This section explains the sequence of operation to be performed for different types of USB transactions.

15.6.5.1 Writing the Transmit FIFO in Slave Mode

The figure below shows the flow diagram for writing to the transmit FIFO in Slave mode. The otg host automatically writes an entry (OUT request) to the Periodic/Non-periodic Request Queue, along with the last DWORD write of a packet. The application must ensure that at least one free space is available in the Periodic/Non-periodic Request Queue before starting to write to the transmit FIFO. The application must always write to the transmit FIFO in DWORDs. If the packet size is non-DWORD aligned, the application must use padding. The otg host determines the actual packet size based on the programmed maximum packet size and transfer size.

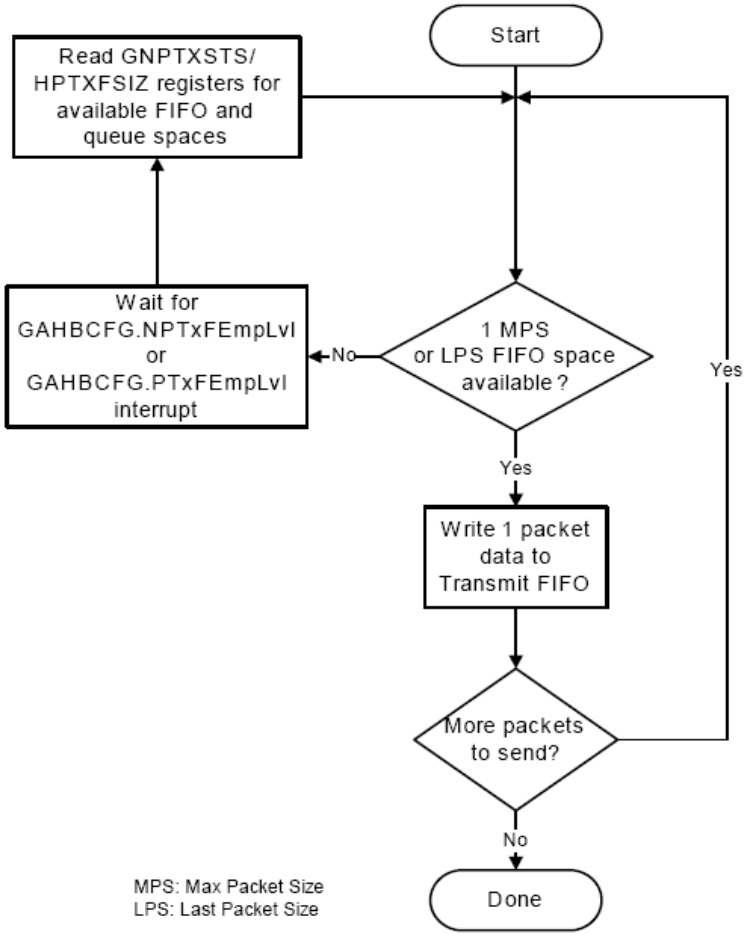


Figure 15.7 Transmit FIFO Write Task in Slave Mode

15.6.5.2 Reading the Receive FIFO in Slave Mode

The figure below shows the flow diagram for reading the receive FIFO in Slave mode. The application must ignore all packet statuses other than IN Data Packet (4’b0010).

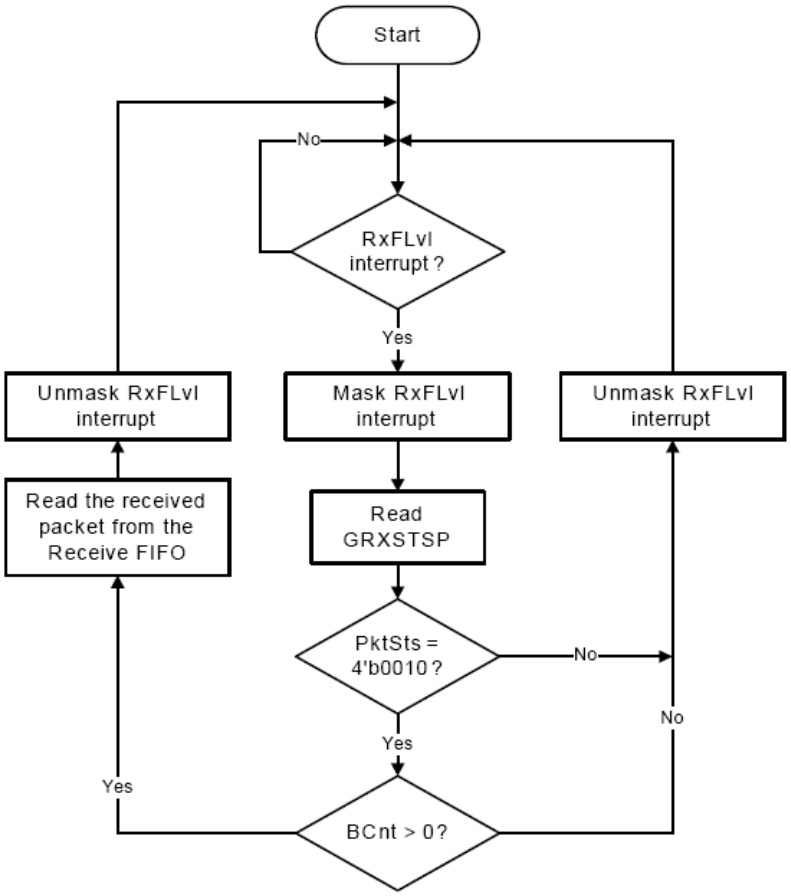


Figure 15.8 Receive FIFO Read Task in Slave Mode

15.6.5.3 Bulk and Control OUT/SETUP Transactions in Slave Mode

A typical bulk or control OUT/SETUP pipelined transaction-level operation in Slave mode is shown in figure below. See channel 1 (ch_1). Two bulk OUT packets are transmitted. A control SETUP transaction operates the same way but has only one packet. The assumptions are:

- The application is attempting to send two maximum-packet-size packets (transfer size = 1, 024 bytes).
- The Non-periodic Transmit FIFO can hold two packets (1 KB for HS or 128 KB for FS).
- The Non-periodic Request Queue depth = 4.

[Normal Bulk and Control OUT/SETUP Operations]

The sequence of operations in figure below(channel 1) is as follows:

1. Initialize channel 1 as explained in “Channel Initialization”.

2. Write the first packet for channel 1.

3. Along with the last DWORD write, the core writes an entry to the Non-periodic Request Queue.

4. As soon as the non-periodic queue becomes non-empty, the core attempts to send an OUT token in the current frame/microframe.

5. Write the second (last) packet for channel 1.

6. The core generates the XferCompl interrupt as soon as the last transaction is completed successfully.

7. In response to the XferCompl interrupt, de-allocate the channel for other transfers.

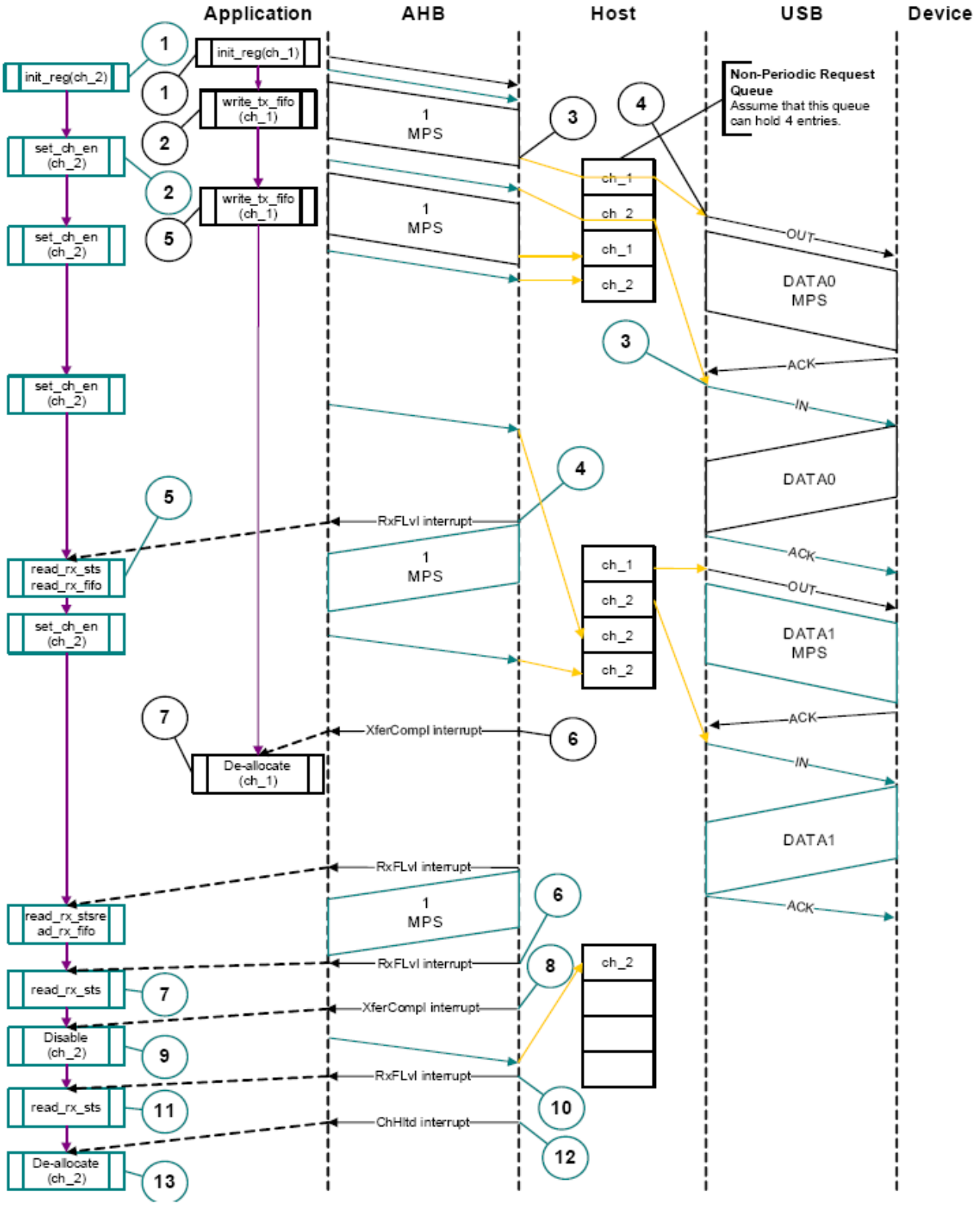


Figure 15.9 Normal Bulk/Control OUT/SETUP and Bulk/Control IN Transactions in Slave Mode

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for bulk and control OUT/SETUP transactions in Slave mode is shown in table below.

Table 15.5 Interrupt Service Routine for Bulk/Control OUT/SETUP and Bulk/Control IN Transactions in Slave Mode

Bulk/Control OUT/SETUP (Non-Split)	Bulk/Control IN (Non-Split)

Unmask (NAK/XactErr/NYET/STALL/XferCompl)
if (XferCompl)
{
Reset Error Count
Mask ACK
De-allocate Channel
}
else if (STALL)
{
Transfer Done = 1
Unmask ChHltd
Disable Channel
}
else if (NAK or XactErr or NYET)
{
Rewind Buffer Pointers
Unmask ChHltd
Disable Channel
if (XactErr)
{
Increment Error Count
Unmask ACK
}
else
{
Reset Error Count
}
}
else if (ChHltd)
{
Mask ChHltd
if (Transfer Done or (Error_count == 3))
{
De-allocate Channel
}
else
{
Re-initialize Channel (Do ping protocol for HS)
}
}
else if (ACK)
{
Reset Error Count
Mask ACK
}
	Unmask (XactErr/XferCompl/BblErr/STALL/DataTglErr)
	if (XferCompl)
	{
	Reset Error Count
	Unmask ChHltd
	Disable Channel
	Reset Error Count
	Mask ACK
	}
	else if (XactErr or BblErr or STALL)
	{
	Unmask ChHltd
	Disable Channel
	if (XactErr)
	{
	Increment Error Count
	Unmask ACK
	}
	}
	else if (ChHltd)
	{
	Mask ChHltd
	if (Transfer Done or (Error_count == 3))
	{
	De-allocate Channel
	}
	else
	{
	Re-initialize Channel
	}
	}
	else if (ACK)
	{
	Reset Error Count
	Mask ACK
	}
	else if (DataTglErr)
	{
	Reset Error Count
	}

1. The application is expected to write the data packets into the transmit FIFO as and when the space is available in the transmit FIFO and the Request queue. The application can make use of GINTSTS.NPTxFEmp interrupt to find the transmit FIFO space.	1. The application is expected to write the requests as and when the Request queue space is available and until the XferCompl interrupt is received. 2. The application must clear and never modify the DoPing bit after enabling the channel and until the XferCompl or ChHltd interrupt is received. The core uses the DoPing bit to flush the excessive IN requests after receiving the last or short packet.

15.6.5.4 Bulk and Control IN Transactions in Slave Mode

A typical bulk or control IN pipelined transaction-level operation in Slave mode is shown in figure above. See channel 2 (ch_2). The assumptions are:

- The application is attempting to receive two maximum-packet-size packets (transfer size = 1, 024 bytes).
- The receive FIFO can contain at least one maximum-packet-size packet and two status DWORDs per packet (520 bytes for HS or 72 bytes for FS).
- The Non-periodic Request Queue depth = 4.

[Normal Bulk and Control IN Operations]

The sequence of operations in figure above (channel 2) is as follows:

1. Initialize channel 2 as explained in “Channel Initialization”.

2. Set the HCCHAR2.ChEna bit to write an IN request to the Non-periodic Request Queue.

3. The core attempts to send an IN token after completing the current OUT transaction.

4. The core generates an RxFLvl interrupt as soon as the received packet is written to the receive FIFO.

5. In response to the RxFLvl interrupt, mask the RxFLvl interrupt and read the received packet status to determine the number of bytes received, then read the receive FIFO accordingly. Following this, unmask the RxFLvl interrupt.

6. The core generates the RxFLvl interrupt for the transfer completion status entry in the receive FIFO.

7. The application must read and ignore the receive packet status when the receive packet status is not an IN data packet (GRXSTSR.PktSts != 4’b0010).

8. The core generates the XferCompl interrupt as soon as the receive packet status is read.

9. In response to the XferCompl interrupt, disable the channel (as explained in “Halting a Channel”) and stop writing the HCCHAR2 register for further requests. The core writes a channel disable request to the Non-periodic Request Queue as soon as the HCCHAR2 register is written.

10. The core generates the RxFLvl interrupt as soon as the halt status is written to the receive FIFO.

11. Read and ignore the receive packet status.

12. The core generates a ChHltd interrupt as soon as the halt status is popped from the receive FIFO.

13. In response to the ChHltd interrupt, de-allocate the channel for other transfers.

Note

For Bulk/Control IN transfer, the application is expected to write the requests when the Request queue space is available, and until the XferCompl interrupt is received.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for bulk and control IN transactions in Slave mode is shown in the table above.

15.6.5.5 Bulk and Control OUT/SETUP Transactions IN DMA Mode

A typical bulk or control OUT/SETUP operation in DMA mode is shown in figure below. See channel 1 (ch_1). Two bulk OUT packets are transmitted. A control SETUP transaction operates the same way but has only one packet. The assumptions are:

- The application is attempting to send two maximum-packet-size packets (transfer size = 1, 024 bytes).
- The Non-periodic Transmit FIFO can hold two packets (1 KB for HS or 128 bytes for FS).
- The Non-periodic Request Queue depth = 4.

[Normal Bulk and Control OUT/SETUP Operations]

The sequence of operations in Figure 15.9 (channel 1) is as follows:

1. Initialize and enable channel 1 as explained in “Channel Initialization”.

2. The otg host starts fetching the first packet as soon as the channel is enabled. For internal DMA mode, the otg host uses the programmed DMA address to fetch the packet.

3. After fetching the last DWORD of the second (last) packet, the otg host masks channel 1 internally for further arbitration.

4. The otg host generates a ChHltd interrupt as soon as the last packet is received.

5. In response to the ChHltd interrupt, de-allocate the channel for other transfers.

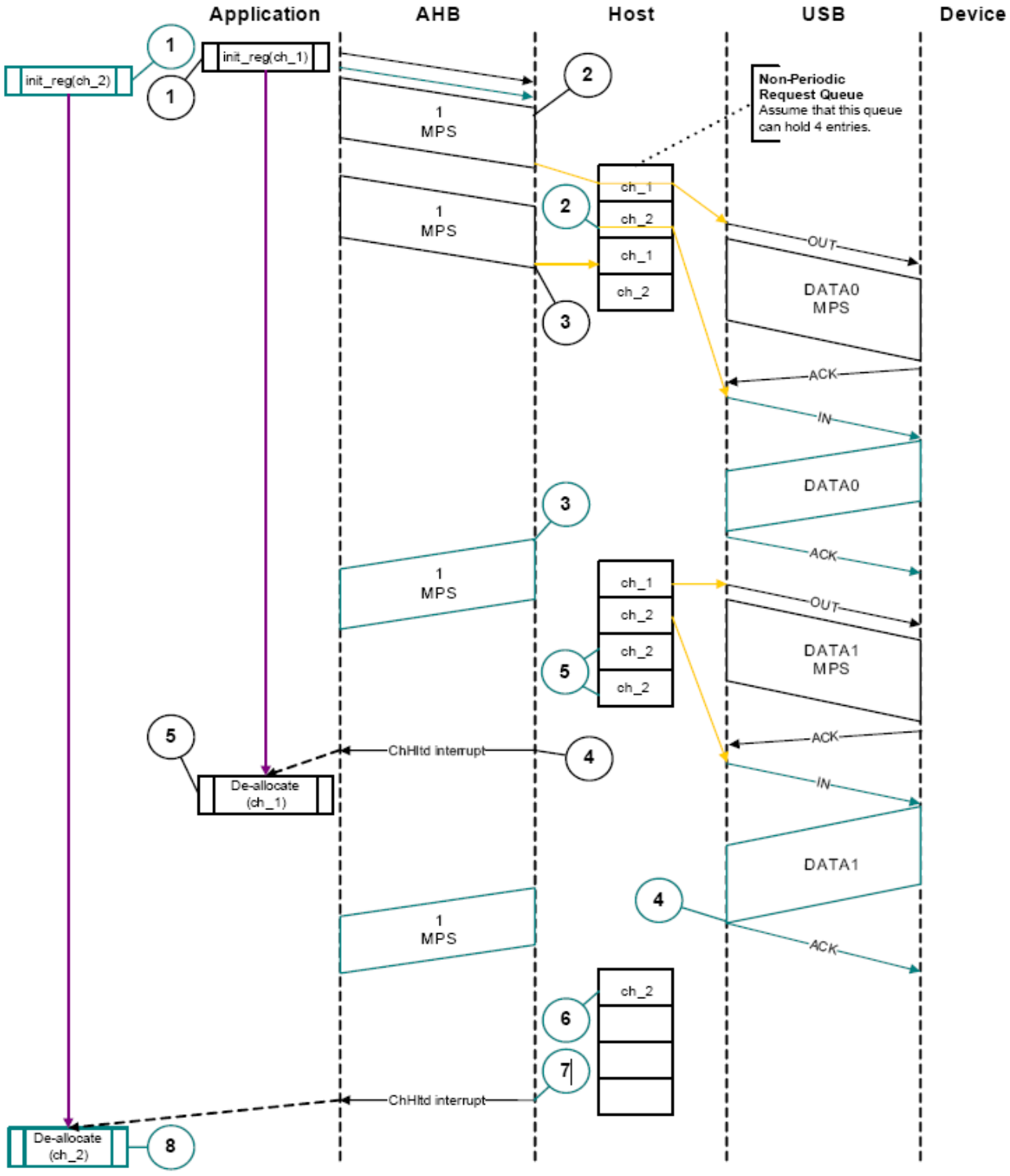


Figure 15.10 Normal Bulk/Control OUT/SETUP and Bulk/Control IN Transactions in DMA Mode

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for bulk and control OUT/SETUP transactions in DMA mode is shown in the table below.

Table 15.6 Interrupt Service Routines for Bulk/Control OUT/SETUP and Bulk/Control IN Transactions in DMA Mode

Bulk/Control OUT/SETUP (Non-Split)	Bulk/Control IN (Non-Split)

Unmask (ChHltd)
if (ChHltd)
{
if (XferCompl or STALL)
{
Reset Error Count
Mask ACK
De-allocate Channel
}
else if (NAK or XactErr or NYET)
{
Rewind Buffer Pointers
if (XactErr)
{
if (Error_count == 2)
{
De-allocate Channel
}
else
{
Increment Error Count
Re-initialize Channel
}
}
else
{
Reset Error Count
Mask ACK
}
Re-initialize Channel
}
}
else if (ACK)
{
Reset Error Count
Mask ACK
}
	Unmask (ChHltd)
	if (ChHltd)
	{
	if (XferCompl or STALL or BblErr)
	{
	Reset Error Count
	Mask ACK
	De-allocate Channel
	}
	else if (XactErr)
	{
	if (Error_count == 2)
	{
	De-allocate Channel
	}
	else
	{
	Unmask ACK
	Unmask NAK
	Unmask DataTglErr
	Increment Error Count
	Re-initialize Channel
	}
	}
	}
	else if (ACK or NAK or DataTglErr)
	{
	Reset Error Count
	Mask ACK
	Mask NAK
	Mask DataTglErr
	}

1. As soon as the channel is enabled, the core attempts to fetch and write data packets, in multiples of the maximum packet size, to the transmit FIFO when space is available in the transmit FIFO and the Request queue. The core stops fetching as soon as the last packet is fetched. 2. While continuing the transfer to a high-speed device, the application must set the DoPing bit before enabling the channel if the previous transaction ended with NAK, NYET, or XacrErr response. In this case, the core starts with the ping protocol, then automatically switches to Data Transfer mode.	1. The application must clear and never modify the DoPing bit after enabling the channel and until the ChHltd interrupt is received. The core uses the DoPing bit to flush the excessive IN requests after receiving the last or short packet.

15.6.5.6 Bulk and Control IN Transactions in DMA Mode

A typical bulk or control IN operation in DMA mode is shown in Figure 5-6. See channel 2 (ch_2). The assumptions are:

- The application is attempting to send two maximum-packet-size packets (transfer size = 1, 024 bytes).
- The application is attempting to receive two maximum-packet-size packets (transfer size = 1, 024 bytes).
- The receive FIFO can hold at least one maximum-packet-size packet and two status DWORDs per packet (520 bytes for HS or 72 bytes for FS).
- The Non-periodic Request Queue depth = 4.

[Normal Bulk and Control IN Operations]

The sequence of operations in the figure above (channel 2) is as follows:

1. Initialize and enable channel 2 as explained in “Channel Initialization”.

2. The otg host writes an IN request to the Request queue as soon as channel 2 receives the grant from the arbiter. (Arbitration is performed in a round-robin fashion, with fairness.).

3. The otg host starts writing the received data to the system memory as soon as the last byte is received with no errors.

4. When the last packet is received, the otg host sets an internal flag to remove any extra IN requests from the Request queue.

5. The otg host flushes the extra requests.

6. The final request to disable channel 2 is written to the Request queue. At this point, channel 2 is internally masked for further arbitration.

7. The otg host generates the ChHltd interrupt as soon as the disable request comes to the top of the queue.

8. In response to the ChHltd interrupt, de-allocate the channel for other transfers.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for bulk and control IN transactions in DMA mode is shown in the table above.

15.6.5.7 Control Transactions in Slave Mode

Setup, Data, and Status stages of a control transfer must be performed as three separate transfers. Setup- Data- or Status-stage OUT transactions are performed similarly to the bulk OUT transactions explained in “Bulk and Control OUT/SETUP Transactions in Slave Mode”. Data- or Status-stage IN transactions are performed similarly to the bulk IN transactions explained in “Bulk and Control IN Transactions in Slave Mode”.

For all three stages, the application is expected to set the HCCHAR1.EPType field to Control. During the Setup stage, the application is expected to set the HCTSIZ1.PID field to SETUP.

15.6.5.8 Control Transactions in DMA Mode

Setup, Data, and Status stages of a control transfer must be performed as three separate transfers. Setup- and Data- or Status-stage OUT transactions are performed similarly to the bulk OUT transactions explained in “Bulk and Control OUT/SETUP Transactions in DMA Mode”. Data- or Status-stage IN transactions are performed similarly to the bulk IN transactions explained in “Bulk and Control IN Transactions in DMA Mode”.

For all three stages, the application is expected to set the HCCHAR1.EPType field to Control. During the Setup stage, the application is expected to set the HCTSIZ1.PID field to SETUP.

15.6.5.9 Interrupt OUT Transactions in Slave Mode

A typical interrupt OUT operation in Slave mode is shown in the figure below. See channel 1 (ch_1). The assumptions are:

- The application is attempting to send one packet in every frame/microframe (up to 1 maximum packet size), starting with the odd frame/microframe (transfer size = 1, 024 bytes).
- The Periodic Transmit FIFO can hold one packet (1 KB for HS or FS).
- Periodic Request Queue depth = 4.

[Normal Interrupt OUT Operation]

The sequence of operations in the figure below (channel 1) is as follows:

1. Initialize and enable channel 1 as explained in “Channel Initialization”. The application must set the HCCHAR1.OddFrm bit.

2. Write the first packet for channel 1. For a high-bandwidth interrupt transfer, the application must write the subsequent packets up to MC (maximum number of packets to be transmitted in the next frame/microframe times before switching to another channel.

3. Along with the last DWORD write of each packet, the otg host writes an entry to the Periodic Request Queue.

4. The otg host attempts to send an OUT token in the next (odd) frame/microframe.

5. The otg host generates an XferCompl interrupt as soon as the last packet is transmitted successfully.

6. In response to the XferCompl interrupt, reinitialize the channel for the next transfer.

[Handling Non-ACK Responses]

The table below shows the channel-specific ISR for an interrupt OUT transaction in Slave mode.

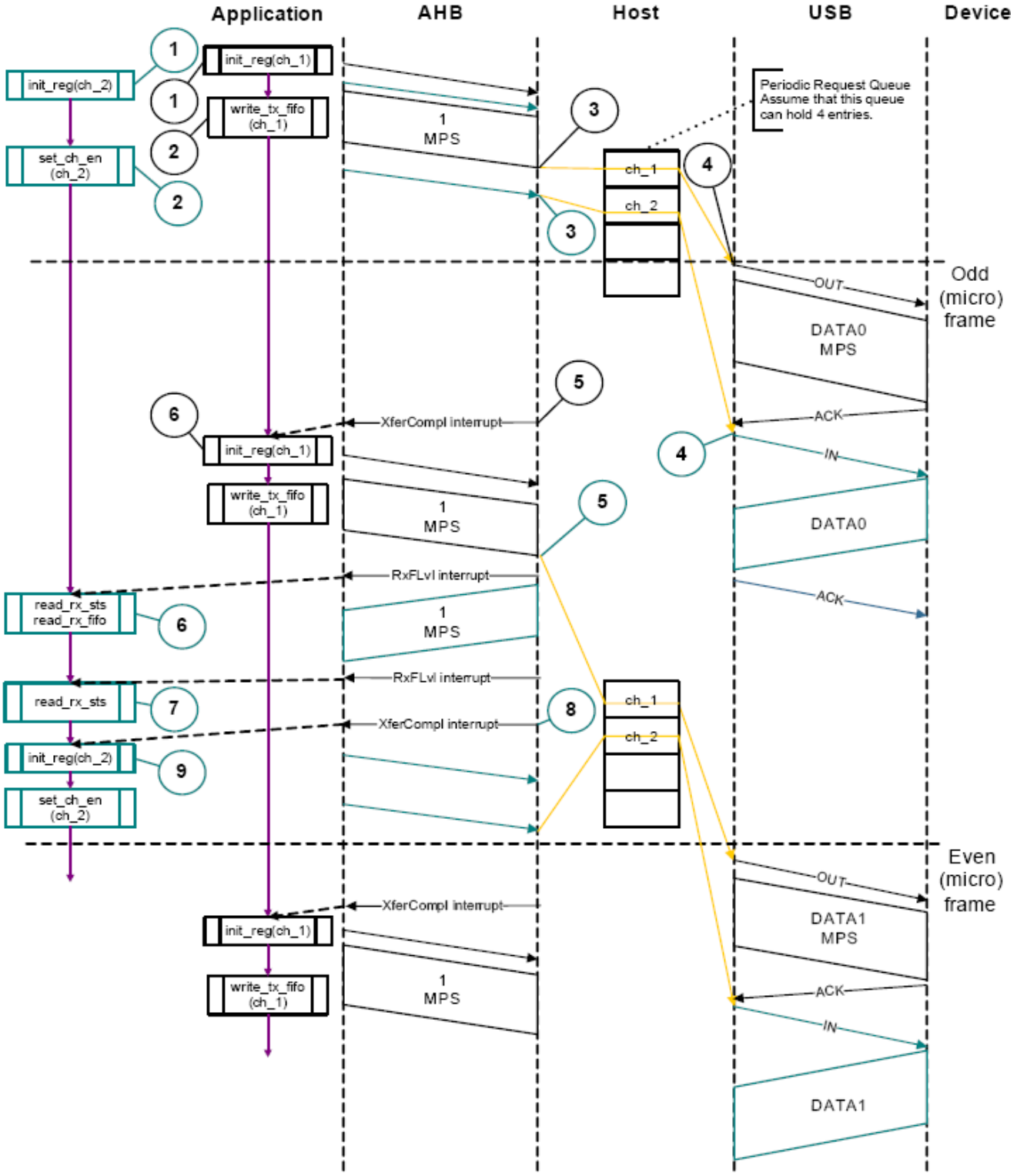


Figure 15.11 Normal Interrupt OUT/IN Transactions in Slave Mode

Table 15.7 Interrupt Service Routine for Interrupt OUT/IN Transactions in Slave Mode

Interrupt OUT (Non-Split)	Interrupt IN (Non-Split)

Unmask
(NAK/XactErr/STALL/XferCompl/FrmOvrun)
if (XferCompl)
{
Reset Error Count
Mask ACK
De-allocate Channel
}
	Unmask
	(NAK/XactErr/XferCompl/BblErr/STALL/FrmOvrun/DataTglErr)
	if (XferCompl)
	{
	Reset Error Count
	Mask ACK
	if (HCTSIZn.PktCnt == 0)
	{

else if (STALL or FrmOvrun)
{
Mask ACK
Unmask ChHltd
Disable Channel
if (STALL)
{
Transfer Done = 1
}
}
else if (NAK or XactErr)
{
Rewind Buffer Pointers
Reset Error Count
Mask ACK
Unmask ChHltd
Disable Channel
}
else if (ChHltd)
{
Mask ChHltd
if (Transfer Done or (Error_count == 3))
{
De-allocate Channel
}
else
{
Re-initialize Channel (in next b_interval - 1 uF/F)
}
}
else if (ACK)
{
Reset Error Count
Mask ACK
}
	De-allocate Channel
	}
	else
	{
	Transfer Done = 1
	Unmask ChHltd
	Disable Channel
	}
	}
	else if (STALL or FrmOvrun or NAK or DataTglErr or BblErr)
	{
	Mask ACK
	Unmask ChHltd
	Disable Channel
	if (STALL or BblErr)
	{
	Reset Error Count
	Transfer Done = 1
	}
	else if (!FrmOvrun)
	{
	Reset Error Count
	}
	}
	else if (XactErr)
	{
	Increment Error Count
	Unmask ACK
	Unmask ChHltd
	Disable Channel
	}
	else if (ChHltd)
	{
	Mask ChHltd
	if (Transfer Done or (Error_count == 3))
	{
	De-allocate Channel
	}
	else
	{
	Re-initialize Channel (in next b_interval - 1 uF/F)
	}
	}
	else if (ACK)
	{
	Reset Error Count
	Mask ACK
	}

1. The application is expected to write the data packets into the transmit FIFO when the space is available in the transmit FIFO and the Request queue up to the count specified in the MC field before switching to another channel. The application uses the GINTSTS.NPTxFEmp interrupt to find the transmit FIFO space.	1. The application is expected to write the requests for the same channel when the Request queue space is available up to the count specified in the MC field before switching to another channel (if any).

15.6.5.10 Interrupt IN Transactions in Slave Mode

A typical interrupt IN operation in Slave mode is shown in the figure above. See channel 2 (ch_2). The assumptions are:

- The application is attempting to receive one packet (up to 1 maximum packet size) in every frame/microframe, starting with odd. (transfer size = 1, 024 bytes).
- The receive FIFO can hold at least one maximum-packet-size packet and two status DWORDs per packet (1, 032 bytes for HS or 1, 031 bytes for FS).
- Periodic Request Queue depth = 4.

[Normal Interrupt IN Operation]

The sequence of operations in the figure above (channel 2) is as follows:

1. Initialize channel 2 as explained in “Channel Initialization”. The application must set the HCCHAR2.OddFrm bit.

2. Set the HCCHAR2.ChEna bit to write an IN request to the Periodic Request Queue. For a high-bandwidth interrupt transfer, the application must write the HCCHAR2 register MC (maximum number of expected packets in the next frame/microframe) times before switching to another channel.

3. The otg host writes an IN request to the Periodic Request Queue for each HCCHAR2 register write with a ChEna bit set.

4. The otg host attempts to send an IN token in the next (odd) frame/microframe.

5. As soon as the IN packet is received and written to the receive FIFO, the otg host generates an RxFLvl interrupt.

6. In response to the RxFLvl interrupt, read the received packet status to determine the number of bytes received, then read the receive FIFO accordingly. The application must mask the RxFLvl interrupt before reading the receive FIFO, and unmask after reading the entire packet.

7. The core generates the RxFLvl interrupt for the transfer completion status entry in the receive FIFO. The application must read and ignore the receive packet status when the receive packet status is not an IN data packet (GRXSTSR.PktSts != 4’b0010).

8. The core generates an XferCompl interrupt as soon as the receive packet status is read.

9. In response to the XferCompl interrupt, read the HCTSIZ2.PktCnt field. If HCTSIZ2.PktCnt != 0, disable the channel (as explained in “Halting a Channel”) before re-initializing the channel for the next transfer, if any). If HCTSIZ2.PktCnt == 0, reinitialize the channel for the next transfer. This time, the application must reset the HCCHAR2.OddFrm bit.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for an interrupt IN transaction in Slave mode is shown in the Table above.

15.6.5.11 Interrupt OUT Transactions in DMA Mode

A typical interrupt OUT operation in DMA mode is shown in figure below. See channel 1 (ch_1). The assumptions are:

- The application is attempting to transmit one packet in every frame/microframe (up to 1 maximum packet size of 1, 024 bytes).
- The Periodic Transmit FIFO can hold one packet (1 KB for HS/FS).
- Periodic Request Queue depth = 4.

[Normal Interrupt OUT Operation]

1. Initialize and enable channel 1 as explained in “Channel Initialization”.

2. The otg host starts fetching the first packet as soon the channel is enabled and writes the OUT request along with the last DWORD fetch. In high-bandwidth transfers, the otg host continues fetching the next packet (up to the value specified in the MC field) before switching to the next channel.

3. The otg host attempts to send the OUT token in the beginning of the next odd frame/microframe.

4. After successfully transmitting the packet, the otg host generates a ChHltd interrupt.

5. In response to the ChHltd interrupt, reinitialize the channel for the next transfer.

[Handling Non-ACK Responses]

Table below shows the channel-specific ISR for an interrupt OUT transaction in DMA mode.

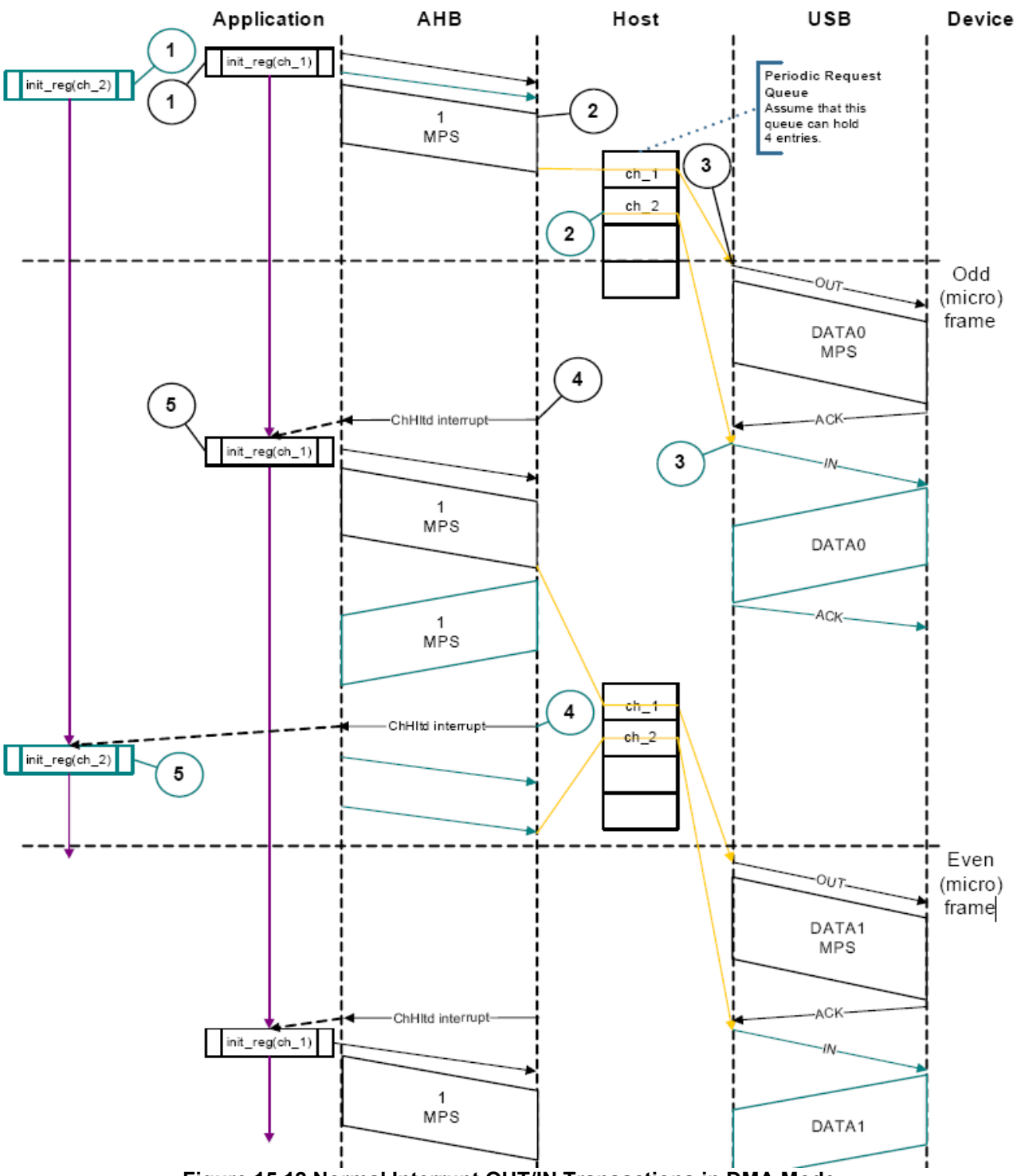


Figure 15.12 Normal Interrupt OUT/IN Transactions in DMA Mode

Table 15.8 Interrupt Service Routine for Interrupt OUT/IN Transactions in DMA Mode

Interrupt OUT (Non-Split)	Interrupt IN (Non-Split)

Unmask (ChHltd)
if (ChHltd)
{
if (XferCompl)
{
Reset Error Count
Mask ACK
if (Transfer Done)
{
De-allocate Channel
}
	Unmask (ChHltd)
	if (ChHltd)
	{
	if (XferCompl)
	{
	Reset Error Count
	Mask ACK
	if (Transfer Done)
	{
	De-allocate Channel
	}

else
{
Re-initialize Channel (in next b_interval - 1 uF/F)
}
}
else if (STALL)
{
Transfer Done = 1
Reset Error Count
Mask ACK
De-allocate Channel
}
else if (NAK or FrmOvrun)
{
Mask ACK
Rewind Buffer Pointers
Re-initialize Channel (in next b_interval - 1 uF/F)
if (NAK)
{
Reset Error Count
}
}
else if (XactErr)
{
if (Error_count == 2)
{
De-allocate Channel
}
else
{
Increment Error Count
Rewind Buffer Pointers
Unmask ACK
Re-initialize Channel (in next b_interval - 1 uF/F)
}
}
}
else if (ACK)
{
Reset Error Count
Mask ACK
}
	else
	{
	Re-initialize Channel (in next b_interval - 1 uF/F)
	}
	}
	else if (STALL or BblErr)
	{
	Reset Error Count
	Mask ACK
	De-allocate Channel
	}
	else if (NAK or DataTglErr or FrmOvrun)
	{
	Mask ACK
	Re-initialize Channel (in next b_interval - 1 uF/F)
	if (DataTglErr or NAK)
	{
	Reset Error Count
	}
	}
	else if (XactErr)
	{
	if (Error_count == 2)
	{
	De-allocate Channel
	}
	else
	{
	Increment Error Count
	Unmask ACK
	Re-initialize Channel (in next b_interval - 1 uF/F)
	}
	}
	}
	else if (ACK)
	{
	Reset Error Count
	Mask ACK
	}

1. As soon as the channel is enabled, the core attempts to fetch and write data packets, in maximum packet size multiples, to the transmit FIFO when the space is available in the transmit FIFO and the Request queue. The core stops fetching as soon as the last packet is fetched (the number of packets is determined by the MC field of the HCCHARn register).	1. As soon as the channel is enabled, the core attempts to write the requests into the Request queue when the space is available up to the count specified in the MC field.

15.6.5.12 Interrupt IN Transactions in DMA Mode

A typical isochronous IN operation in DMA mode is shown in the figure above. See channel 2 (ch_2). The assumptions are:

- The application is attempting to receive one packet in every frame/microframe (up to 1 maximum packet size of 1, 024 bytes).
- The receive FIFO can hold at least one maximum-packet-size packet and two status DWORDs per packet (1, 032 bytes for HS/FS).
- Periodic Request Queue depth = 4.

[Normal Interrupt IN Operation]

The sequence of operations in the figure above (channel 2) is as follows:

1. Initialize and enable channel 2 as explained in “Channel Initialization”.

2. The otg host writes an IN request to the Request queue as soon as the channel 2 gets the grant from the arbiter (round-robin with fairness). In high-bandwidth transfers, the otg host writes consecutive writes up to MC times.

3. The otg host attempts to send an IN token at the beginning of the next (odd) frame/microframe.

4. As soon the packet is received and written to the receive FIFO, the otg host generates a ChHltd interrupt.

5. In response to the ChHltd interrupt, reinitialize the channel for the next transfer.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for Interrupt IN transaction in DMA mode is shown in the table above.

15.6.5.13 Isochronous OUT Transactions in Slave Mode

A typical isochronous OUT operation in Slave mode is shown in figure below. See channel 1 (ch_1). The assumptions are: The application is attempting to send one packet every frame/microframe (up to 1 maximum packet size), starting with an odd frame/microframe. (transfer size = 1, 024 bytes).

- The Periodic Transmit FIFO can hold one packet (1 KB for HS/FS).
- Periodic Request Queue depth = 4.

[Normal Isochronous OUT Operation]

The sequence of operations in the figure below (channel 1) is as follows:

1. Initialize and enable channel 1 as explained in “Channel Initialization”. The application must set the HCCHAR1.OddFrm bit.

2. Write the first packet for channel 1. For a high-bandwidth isochronous transfer, the application must write the subsequent packets up to MC (maximum number of packets to be transmitted in the next frame/microframe) times before switching to another channel.

3. Along with the last DWORD write of each packet, the otg host writes an entry to the Periodic Request Queue.

4. The otg host attempts to send the OUT token in the next frame/microframe (odd).

5. The otg host generates the XferCompl interrupt as soon as the last packet is transmitted successfully.

6. In response to the XferCompl interrupt, reinitialize the channel for the next transfer.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for an isochronous OUT transaction in Slave mode is shown in table below.

Table 15.9 Interrupt Service Routine for Isochronous OUT/IN Transactions in Slave Mode

Isochronous OUT (Non-Split)	Isochronous IN (Non-Split)

Unmask (FrmOvrun/XferCompl)
if (XferCompl)
{
De-allocate Channel
}
else if (FrmOvrun)
{
Unmask ChHltd
Disable Channel
}
else if (ChHltd)
{
Mask ChHltd
De-allocate Channel
}
	Unmask (XactErr/XferCompl/FrmOvrun/BblErr)
	if (XferCompl or FrmOvrun)
	{
	if (XferCompl and (HCTSIZn.PktCnt == 0))
	{
	Reset Error Count
	De-allocate Channel
	}
	else
	{
	Unmask ChHltd
	Disable Channel
	}
	}
	else if (XactErr or BblErr)
	{
	Increment Error Count

	Unmask ChHltd
	Disable Channel
	}
	else if (ChHltd)
	{
	Mask ChHltd
	if (Transfer Done or (Error_count == 3))
	{
	De-allocate Channel
	}
	else
	{
	Re-initialize Channel
	}
	}

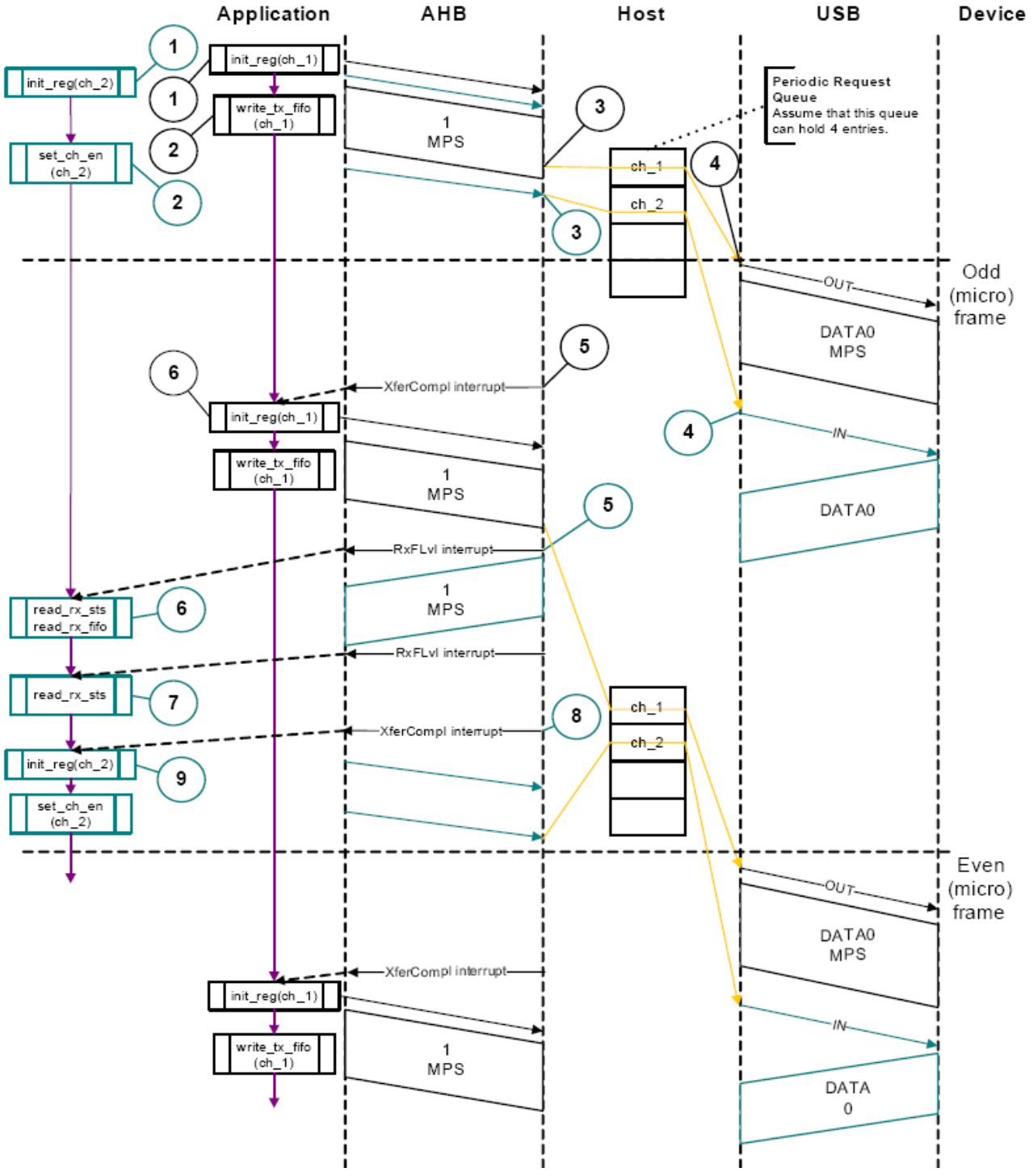


Figure 15.13 Normal Isochronous OUT/IN Transactions in Slave Mode

15.6.5.14 Isochronous IN Transactions in DMA Mode

A typical isochronous IN operation in DMA mode is shown in figure below. See channel 2 (ch_2). The assumptions are:

- The application is attempting to receive one packet in every frame/microframe (up to 1 maximum packet size of 1, 024 bytes).
- The receive FIFO can hold at least one maximum-packet-size packet and two status DWORDS per packet (1, 032 bytes for HS/FS).
- Periodic Request Queue depth = 4.

[Normal Isochronous IN Operation]

The sequence of operations in the figure above(channel 2) is as follows:

1. Initialize and enable channel 2 as explained in “Channel Initialization”.

2. The otg host writes an IN request to the Request queue as soon as the channel 2 gets the grant from the arbiter (round-robin with fairness). In high-bandwidth transfers, the otg host performs consecutive writes up to MC times.

3. The otg host attempts to send an IN token at the beginning of the next (odd) frame/microframe.

4. As soon the packet is received and written to the receive FIFO, the OTG host generates a ChHltd interrupt.

5. In response to the ChHltd interrupt, reinitialize the channel for the next transfer.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for an isochronous IN transaction in DMA

mode is shown in the table above.

15.6.5.15 Bulk and Control OUT/SETUP Split Transactions in Slave Mode

A typical bulk or control SETUP/OUT operation in Slave mode is shown in figure below. See channel 1 (ch_1). Two bulk OUT packets are transmitted. A control SETUP transaction operates the same way but has only one packet. The assumptions are:

- The application is attempting to transmit one packet.

[Normal Bulk and Control OUT/SETUP Split Operations]

The sequence of operations in figure below (channel 1) is as follows:

1. Initialize and enable channel 1 as explained in “Channel Initialization”.

2. Write the packet for channel 1. Along with the last DWORD write of the packet, the otg host writes an entry to the Periodic Request Queue.

3. The otg host sends an OUT token.

4. The otg host generates an ACK interrupt as soon as the start split transaction completes successfully.

5. In response to the ACK interrupt, set the HCSPLT1.ComplSplt to send the complete split.

6. The otg host sends out the complete split transaction.

7. The otg host generates the XferCompl interrupt after successfully completing the complete split transaction.

8. In response to XferCompl interrupt, de-allocate the channel.

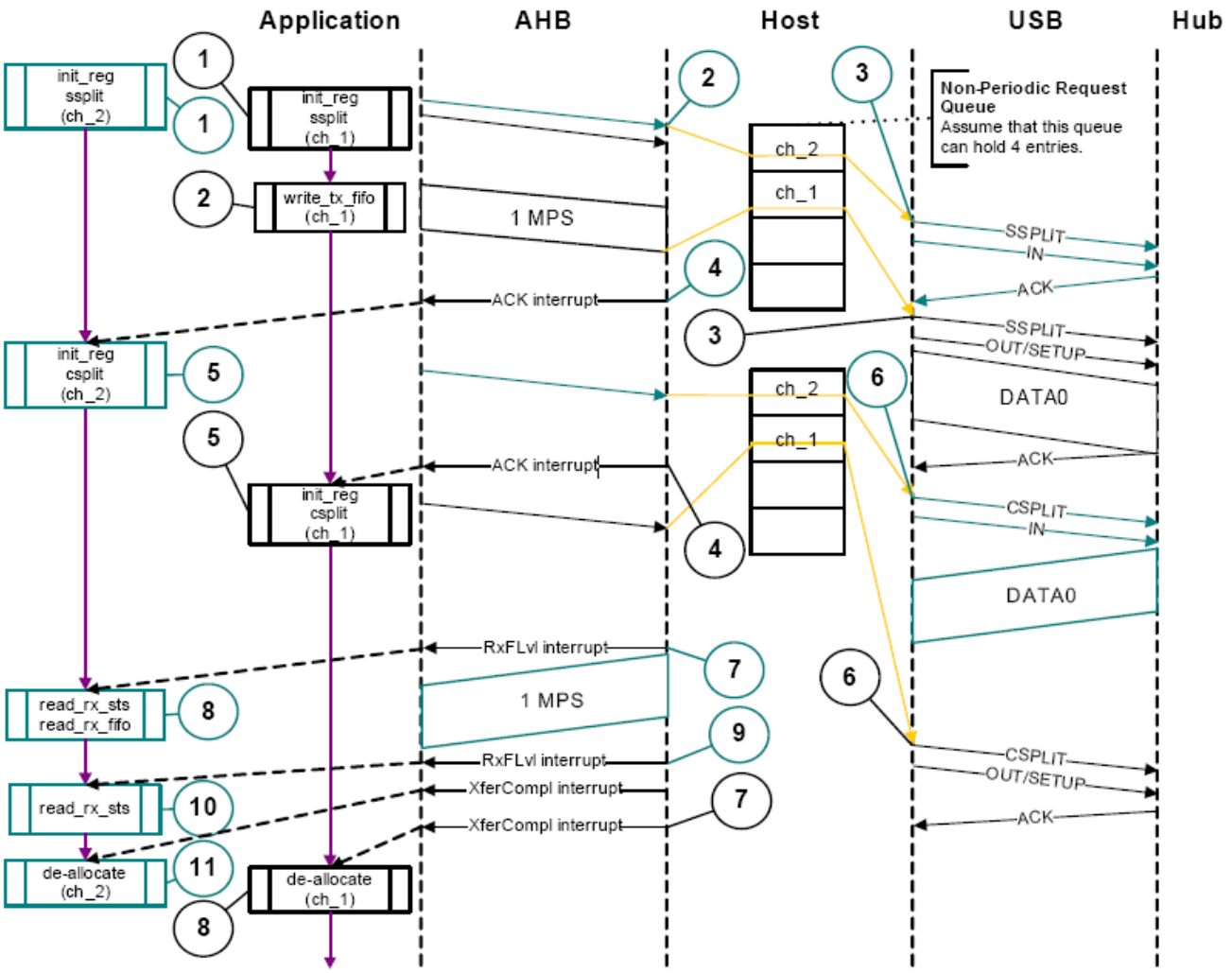


Figure 15.14 Normal Bulk/Control OUT/SETUP and Bulk/Control IN Split Transactions in Slave Mode

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for bulk and control OUT/SETUP split transactions in Slave mode is shown in table below.

Table 15.10 Interrupt Service Routine for Bulk/Control OUT/SETUP and Bulk/Control IN Split Transactions in Slave Mode

	Bulk/Control OUT/SETUP (Split)	Bulk/Control IN (Split)

Start Split	if (ACK)
	{
	Reset Error Count
	Do Complete Split
	}
	else if (NAK)
	{
	Rewind Buffer Pointers
	Retry Start Split
	}
	else if (XactErr)
	{
	Rewind Buffer Pointers
	Increment Error Count
	if (Error_count < 3)
	{
	Retry Start Split
	}
	else
	{
		Unmask (ACK/NAK/XactErr/DataTglErr)
		if (ACK)
		{
		Reset Error Count
		Do Complete Split
		}
		else if (NAK)
		{
		Retry Start Split
		}
		else if (XactErr/DataTglErr)
		{
		Increment Error Count
		if (Error_count < 3)
		{
		Retry Start Split
		}
		else
		{
		Unmask ChHltd

	Unmask ChHltd
	Disable Channel
	}
	}
	else if (ChHltd)
	{
	Mask ChHltd
	De-allocate Channel
	}
		Disable Channel
		}
		}
		else if (ChHltd)
		{
		Mask ChHltd
		De-allocate Channel
		}

Complete Split	Unmask (NAK/XactErr/NYET/STALL/XferCompl)
	if (XferCompl)
	{
	De-allocate Channel
	}
	else if (NAK)
	{
	Rewind Buffer Pointers
	Retry Start Split
	}
	else if (NYET)
	{
	Retry Complete Split
	}
	else if (STALL)
	{
	Unmask ChHltd
	Disable Channel
	}
	else if (XactErr)
	{
	Rewind Buffer Pointers
	Increment Error Count
	if (Error_count < 3)
	{
	Retry Start Split
	}
	else
	{
	Unmask ChHltd
	Disable Channel
	}
	}
	else if (ChHltd)
	{
	Mask ChHltd
	De-allocate Channel
	}
		Unmask (NAK/XactErr/NYET/STALL/XferCompl)
		if (XferCompl)
		{
		De-allocate Channel
		}
		else if (NAK)
		{
		Retry Start Split
		}
		else if (NYET)
		{
		Retry Complete Split
		}
		else if (STALL or BblErr)
		{
		Unmask ChHltd
		Disable Channel
		}
		else if (XactErr)
		{
		Increment Error Count
		if (Error_count < 3)
		{
		Retry Start Split
		}
		else
		{
		Unmask ChHltd
		Disable Channel
		}
		}
		else if (ChHltd)
		{
		Mask ChHltd
		De-allocate Channel
		}

15.6.5.16 Bulk and Control IN Split Transactions in Slave Mode

A typical bulk or control IN operation in Slave mode is shown in the figure above (see channel 2 [ch_2). The assumptions are:

- The application is attempting to receive one packet.

[Normal Bulk and Control IN Split Operations]

The sequence of operations in the figure above (channel 2) is as follows:

1. Initialize and enable channel 2 as explained in “Channel Initialization”.

2. The otg host writes the Start Split request to the Periodic Request Queue as soon as the HCCHAR2 register is written.

3. The otg host sends the Start Split IN token.

4. As soon as the IN token is transmitted, the otg host generates an ACK interrupt.

5. In response to the ACK interrupt, set the HCSPLT2.ComplSplt bit to send the complete split token.

6. The otg host sends the complete split token.

7. As soon as the received packet is written to the receive FIFO, the otg host generates the RxFLvl interrupt.

8. In response to the RxFLvl interrupt, read the received packet status to determine the number of bytes received, then read the receive FIFO accordingly. The application must mask the RxFLvl interrupt before reading the receive FIFO and unmask it after reading the entire packet.

9. The core generates the RxFLvl interrupt for the transfer completion status entry in the receive FIFO.

10. The application must read the receive packet status and, when the receive packet status is not an IN data packet (GRXSTSR.PktSts != 4'b0010), ignore it.

11. The core generates the XferCompl interrupt as soon as the receive packet status is read. In response to the XferCompl interrupt, read the HCTSIZ2.PktCnt field. If HCTSIZ2.PktCnt != 0, disable the channel (as explained in "Halting a Channel") before re-initializing the channel for the next transfer, if any.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for bulk and control IN split transactions inSlave mode is shown in the table above.

15.6.5.17 Bulk and Control OUT/SETUP Split Transactions in DMA Mode

A typical bulk or control OUT/SETUP operation in DMA mode is shown in figure below. See channel 1 (ch_1). Two bulk OUT packets are transmitted. A control SETUP transaction operates the same way but has only one packet. It is assumed that the application is attempting to transmit one packet.

[Normal Bulk and Control OUT/SETUP Split Operations]

The sequence of operations in figure below (channel 1) is as follows:

Initialize and enable channel 1 for start split as explained in "Channel Initialization".
The host starts fetching the first packet as soon the channel is enabled and writes the OUT request along with the last DWORD fetch.

After successfully transmitting start split, the OTG host generates the ChHltd interrupt.

In response to the ChHltd interrupt, set the HCSPLT1.ComplSplt bit to send the complete split.

After successfully transmitting complete split, the OTG host generates the ChHltd interrupt.

In response to the ChHltd interrupt, de-allocate the channel.

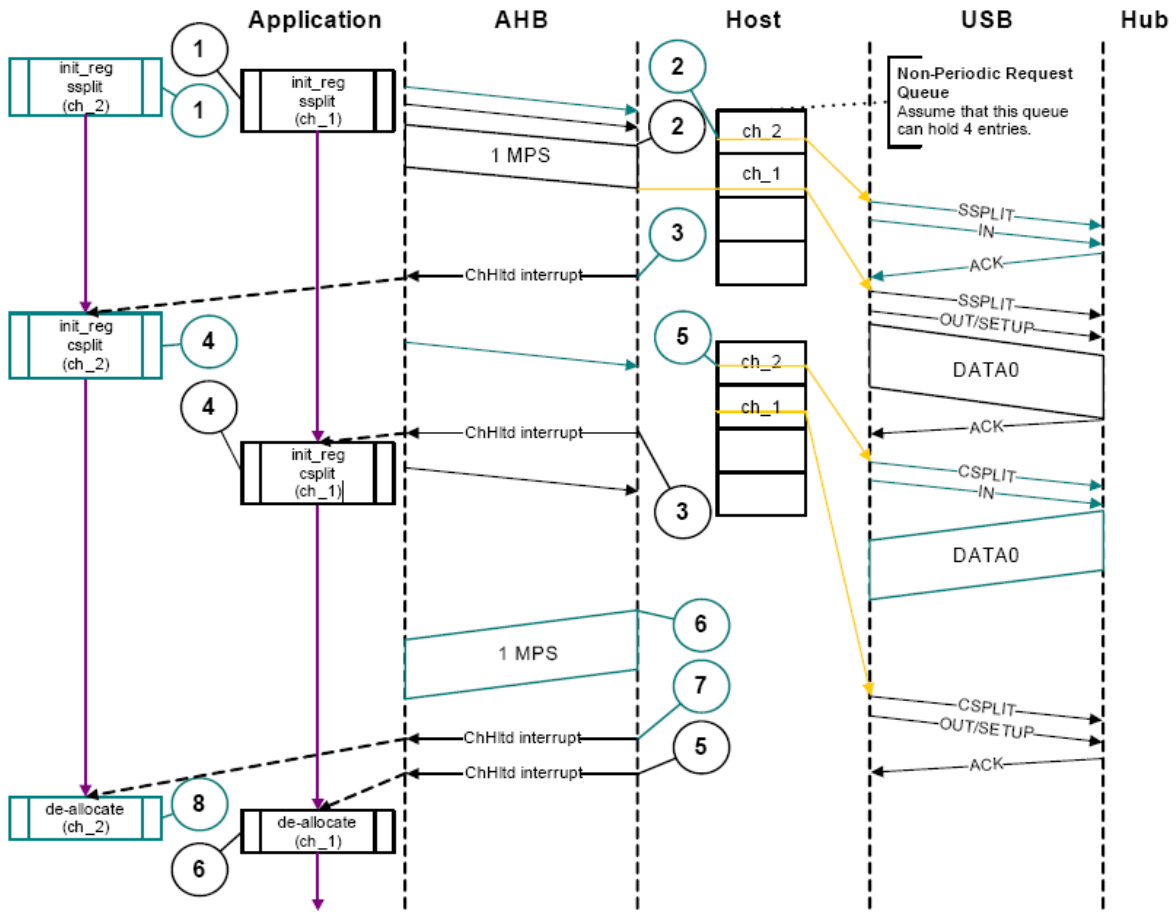


Figure 15.15 Normal Bulk/Control OUT/SETUP and Bulk/Control IN Split Transactions in DMA Mode

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for bulk and control OUT/SETUP split transactions in DMA mode is shown in table below.

Table 15.11 Interrupt Service Routine for Bulk/Control OUT/SETUP and Bulk/Control IN Split Transactions in DMA Mode

	Bulk/Control OUT/SETUP (Split)	Bulk/Control IN (Split)

Start Split	if (ChHltd)
	{
	if (ACK)
	{
	Reset Error Count
	Do Complete Split
	}
	else if (NAK)
	{
	Rewind Buffer Pointers
	Retry Start Split
	}
	else if (XactErr)
	{
	Rewind Buffer Pointers
	Increment Error Count
	if (error_count < 3)
	{
	Retry Start Split
	}
	else
	{
	De-allocate Channel
	}
		Unmask (ChHltd)
		if (ChHltd)
		{
		if (ACK)
		{
		Reset Error Count
		Do Complete Split
		}
		else if (NAK)
		{
		Retry Start Split
		}
		else if (XactErr)
		{
		Increment Error Count
		if (error_count < 3)
		{
		Retry Start Split
		}
		else
		{
		De-allocate Channel
		}
		}

	} }	}

Complete Split	Unmask (ChHltd)
	if (ChHltd)
	{
	if (XferCompl)
	{
	De-allocate Channel
	}
	else if (NAK)
	{
	Rewind Buffer Pointers
	Retry Start Split
	}
	else if (NYET)
	{
	Retry Complete Split
	}
	else if (STALL)
	{
	De-allocate Channel
	}
	else if (XactErr)
	{
	Rewind Buffer Pointers
	Increment Error Count
	if (error_count < 3)
	{
	Retry Start Split
	}
	else
	{
	De-allocate Channel
	}
	}
	}
		Unmask (ChHltd)
		if (ChHltd)
		{
		if (XferCompl)
		{
		De-allocate Channel
		}
		else if (NAK)
		{
		Retry Start Split
		}
		else if (NYET)
		{
		Retry Complete Split
		}
		else if (STALL/BblErr)
		{
		De-allocate Channel
		}
		else if (XactErr)
		{
		Increment Error Count
		if (error_count < 3)
		{
		Retry Start Split
		}
		else
		{
		De-allocate Channel
		}
		}
		}

15.6.5.18 Bulk/Control IN Split Transactions in DMA Mode

A typical bulk or control IN operation in DMA mode is shown in the figure above. See channel 1 (ch_1). The assumptions are:

- The application is attempting to receive one packet.

[Normal Bulk and Control IN Split Operations]

The sequence of operations in the figure above (channel 2) is as follows:

1. Initialize and enable channel 2 as explained in “Channel Initialization”.

2. The otg host writes the start split request to the non-periodic request after getting the grant from the arbiter. The otg host masks the channel 2 internally for the arbitration after writing the request.

3. As soon as the IN token is transmitted, the otg host generates the ChHltd interrupt.

4. In response to the ChHltd interrupt, set the HCSPLT2.ComplSplt bit and re-enable the channel to send the complete split token. This unmasks channel 2 for arbitration.

5. The otg host writes the complete split request to the non-periodic request after receiving the grant from the arbiter.

6. The otg host starts writing the packet to the system memory after receiving the packet successfully.

7. As soon as the received packet is written to the system memory, the otg host generates a ChHltd interrupt.

8. In response to the ChHltd interrupt, de-allocate the channel.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for bulk and control IN split transactions in DMA mode is shown in the table above.

15.6.5.19 Interrupt OUT Split Transactions in Slave Mode

A typical interrupt OUT split operation in Slave mode is shown in figure below. See channel 1 (ch_1). The assumptions are:

- The application is attempting to transmit one maximum-packet-size packet in an odd microframe.

[Normal Interrupt OUT Split Operation]

The sequence of operations in figure below (channel 1) is as follows:

1. Initialize and enable channel 1 as explained in “Channel Initialization”. The application must set the HCCHAR1.OddFrm bit.

2. Write the packet for channel 1.

3. Along with the last DWORD write of the packet, the otg host writes an entry to the Periodic Request Queue.

4. The otg host attempts to send an OUT token in the next odd microframe.

5. The otg host generates an ACK interrupt as soon as the packet is transmitted successfully.

6. In response to the ACK interrupt, set the HCSPLT1.ComplSplt to send the complete split.

7. The otg host generates the XferCompl interrupt after successfully completing the complete split transaction.

8. In response to the XferCompl interrupt, de-allocate the channel.

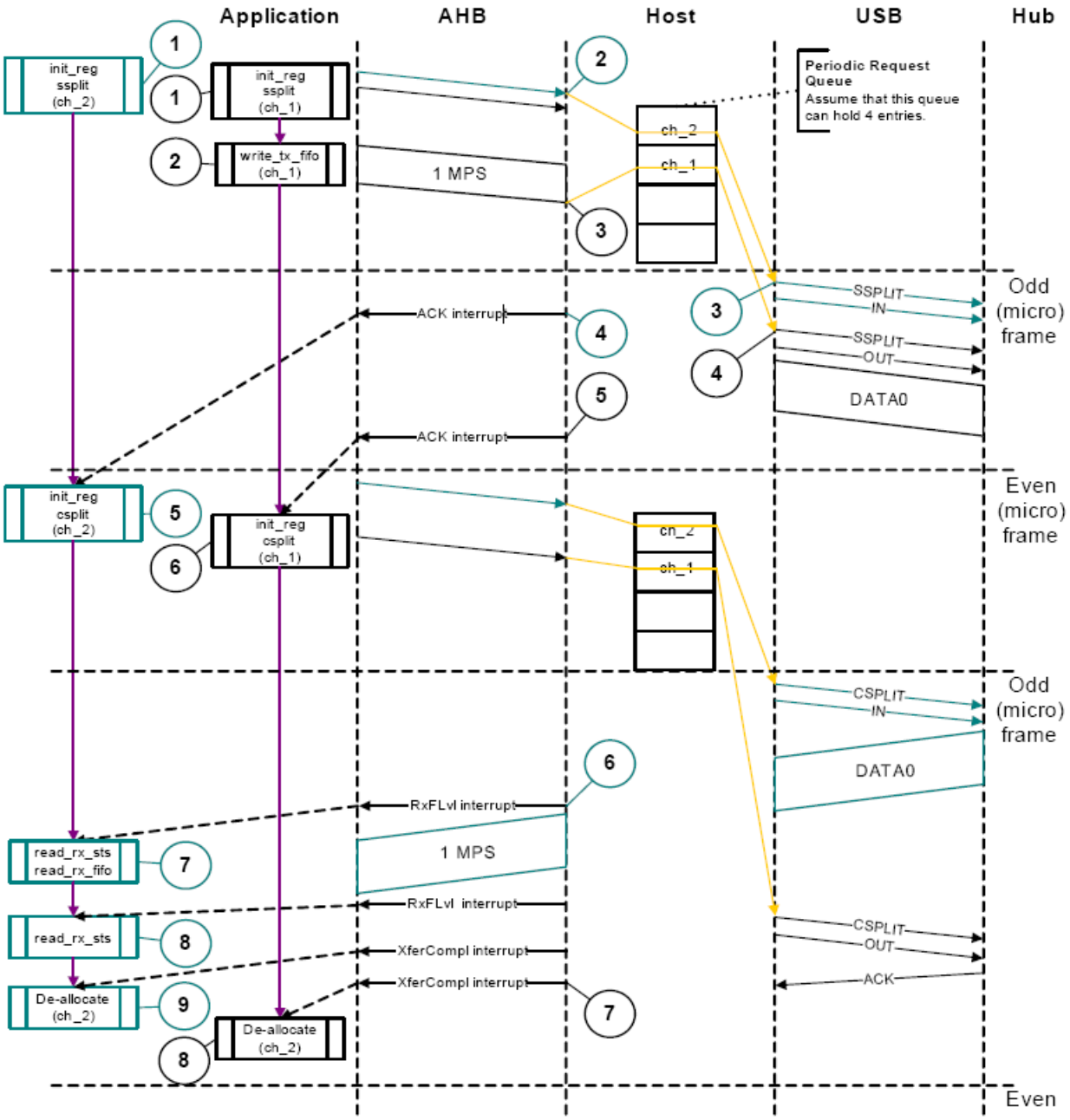


Figure 15.16 Normal Interrupt OUT/IN Split Transactions in Slave Mode

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for interrupt OUT and IN split transactions in Slave mode is shown in table below.

	Interrupt OUT (Split)	Interrupt IN (Split)

Start Split	Unmask (ACK/FrmOvrun)
	if (ACK)
	{
	Do Complete Split
	}
	else if (FrmOvrun)
	{
	Rewind Buffer Pointers
	Unmask ChHltd
	Disable Channel
	(Retry Start Split (in next b_interval - 1 uF))
		Unmask (ACK/FrmOvrun/
		DataTglErr)
		if (ACK)
		{
		Do Complete Split
		}
		else if (FrmOvrun/DataTglErr)
		{
		Rewind Buffer Pointers
		Unmask ChHltd
		Disable Channel

	}
	else if (ChHltd)
	{
	Mask ChHltd
	De-allocate Channel
	}
		(Retry Start Split (in next b_interval - 1 uF))
		}
		else if (ChHltd)
		{
		Mask ChHltd
		De-allocate Channel
		}

Complete Split	Unmask (NAK/XactErr/NYET/STALL/XferCompl/FrmOvrun)
	if (XferCompl)
	{
	De-allocate Channel
	}
	else if (NAK)
	{
	Unmask ChHltd
	Disable Channel
	(Retry Start Split (in next b_interval - 1 uF))
	}
	else if (NYET)
	{
	Retry Complete Split
	}
	else if (STALL or FrmOvrun)
	{
	Unmask ChHltd
	Disable Channel
	}
	else if (XactErr)
	{
	Rewind Buffer Pointers
	Unmask ChHltd
	Disable Channel
	(If (HCCHARn.EC == 3), Retry Start Split (in next b_interval - 1 uF))
	}
	else if (ChHltd)
	{
	Mask ChHltd
	De-allocate Channel
	}
		Unmask (NAK/XactErr/NYET/STALL/XferCompl/FrmOvrun/BblErr)
		if (XferCompl)
		{
		De-allocate Channel
		}
		else if (NAK)
		{
		Unmask ChHltd
		Disable Channel
		(Retry Start Split (in next b_interval - 1 uF))
		}
		else if (NYET)
		{
		Retry Complete Split
		}
		else if (STALL or FrmOvrun or BblErr)
		{
		Unmask ChHltd
		Disable Channel
		}
		else if (XactErr)
		{
		Rewind Buffer Pointers
		Unmask ChHltd
		Disable Channel
		(If (HCCHARn.EC == 3), Retry Start Split (in next b_interval - 1 uF))
		}
		else if (ChHltd)
		{
		Mask ChHltd
		De-allocate Channel
		}

Note

The otg host tracks the error count in EC field for periodic split transactions. If the EC field matches the original programmed error count after XactErr interrupt, the application must treat the XactErr as “ERR response received.” The EC field indicates the number of immediate retries that the OTG host has performed before generating the XactErr interrupt.

15.6.5.20 Interrupt IN Split Transactions in Slave Mode

A typical interrupt IN split operation in Slave mode is shown in the figure above. See channel 2 (ch_2). The assumptions are:

- The application is attempting to receive one maximum-packet-size packet in an odd microframe.
- The receive FIFO can hold at least one maximum-packet-size packet and two status DWORDs per packet.

[Normal Interrupt IN Split Operation]

The sequence of operations in the figure above (channel 2) is as follows:

1. Initialize and enable channel 2 as explained in “Channel Initialization”. The application must set the HCCHAR2.OddFrm bit.

2. The otg host writes the start split request to the Periodic Request Queue as soon as the HCCHAR2 register is written.

3. The otg host attempts to send a start split IN token in the next odd microframe.

4. As soon as the IN packet is transmitted, the otg host generates an ACK interrupt.

5. In response to the ACK interrupt, set the HCSPLT2.CompSplt bit in the next microframe to send the complete split token in the next odd microframe.

6. As soon as the received packet is written to the receive FIFO, the otg host generates the RxFLvl interrupt.

7. In response to the RxFLvl interrupt, read the received packet status to determine the number of bytes received, then read the receive FIFO accordingly. The application must mask the RxFLvl interrupt before reading the receive FIFO and unmask after reading the entire packet.

8. The core generates the RxFLvl interrupt for the transfer completion status entry in the receive FIFO. The application must read the receive packet status and, if the receive packet status is not an IN data packet (GRXSTSR.PktSts != 4'b0010), ignore it.

9. The core generates the XferCompl interrupt as soon as the receive packet status is read. In response to the XferCompl interrupt, read the HCTSIZ2.PktCnt field. If HCTSIZ2.PktCnt != 0, disable the channel (as explained in "Halting a Channel" on page 216) before re-initializing the channel for the next transfer, if any. If HCTSIZ2.PktCnt == 0, reinitialize the channel for the next transfer. This time, the application must reset the HCCHAR2.OddFrm bit.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for an interrupt IN split transaction in Slave mode is shown in the table above.

15.6.5.21 Interrupt OUT Split Transactions in DMA Mode

A typical interrupt OUT split operation in DMA mode is shown in figure below (see channel 1 [ch_1]). It is assumed that the application is attempting to transmit one packet (1 maximum packet size) in an odd microframe.

[Normal Interrupt OUT Split Operation]

The sequence of operations in figure below (channel 1) is as follows:

1. Initialize and enable channel 1 for start split as explained in "Channel Initialization". The application must set the HCCHAR1.OddFrm bit.

2. The otg host starts reading the packet.

3. The otg host attempts to send the start split transaction.

4. After successfully transmitting the start split, the otg host generates the ChHltd interrupt.

5. In response to the ChHltd interrupt, set the HCSPLT1.ComplSplt bit to send the complete split.

6. After successfully completing the complete split transaction, the otg host generates the ChHltd interrupt.

7. In response to ChHltd interrupt, de-allocate the channel.

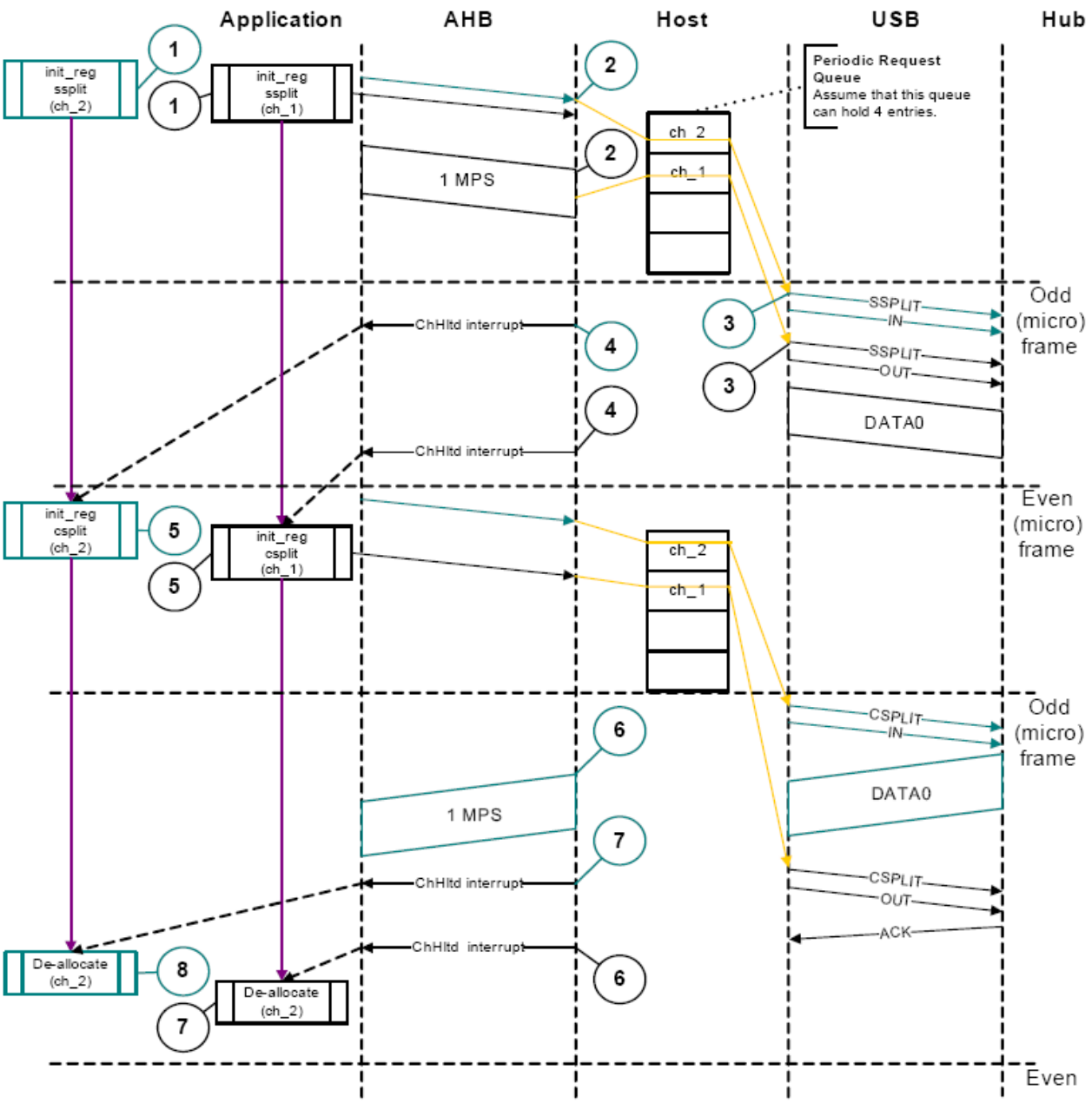


Figure 15.17 Normal Interrupt OUT/IN Split Transactions in DMA Mode

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for interrupt OUT split transaction in DMA mode is shown in table below.

Table 15.12 Interrupt Service Routine for Interrupt OUT/IN Split Transactions in DMA Mode

	Interrupt OUT (Split)	Interrupt IN (Split)

Start Split	Unmask (ChHltd)
	if (ChHltd)
	{
	if (ACK)
	{
	Do Complete Split
	}
	else if (FrmOvrun)
	{
	Rewind Buffer Pointers
	Retry Start Split (in next b_interval - 1 uF)
	}
		Unmask (ChHltd)
		if (ChHltd)
		{
		if (ACK)
		{
		Do Complete Split
		}
		else if (FrmOvrun)
		{
		Rewind Buffer Pointers
		Retry Start Split (in next b_interval - 1 uF)
		}

	}	}

Complete Split	Unmask (ChHltd)
	if (ChHltd)
	{
	if (XferCompl)
	{
	De-allocate Channel
	}
	else if (NAK)
	{
	Retry Start Split (in next b_interval - 1 uF)
	}
	else if (NYET)
	{
	Retry Complete Split
	}
	else if (STALL or FrmOvrun)
	{
	De-allocate Channel
	}
	else if (XactErr)
	{
	Rewind Buffer Pointers
	if (HCCHARn.EC == 3) // ERR response received
	{
	Retry Start Split (in next b_interval - 1 uF)
	}
	else
	{
	De-allocate Channel
	}
	}
	}
		Unmask (ChHltd)
		if (ChHltd)
		{
		if (XferCompl)
		{
		De-allocate Channel
		}
		else if (NAK)
		{
		Retry Start Split (in next b_interval - 1 uF)
		}
		else if (NYET)
		{
		Retry Complete Split
		}
		else if (STALL or FrmOvrun or BblErr)
		{
		De-allocate Channel
		}
		else if (XactErr)
		{
		Rewind Buffer Pointers
		if (HCCHARn.EC == 3) // ERR response received
		{
		Retry Start Split (in next b_interval - 1 uF)
		}
		else
		{
		De-allocate Channel
		}
		}
		}

Note

The otg host tracks the error count in the EC field for periodic split transactions. If the EC field matches the original programmed error count after XactErr interrupt, the application must treat the XactErr as an ERR Response Received. The EC field indicates the number of immediate retries the OTG host has performed before generating the XactErr interrupt.

15.6.5.22 Interrupt IN Split Transactions in DMA Mode

A typical interrupt IN split operation in DMA mode is shown in the figure above. See channel 2 (ch_2). The assumptions are:

- The application is attempting to receive one maximum-packet-size packet in an odd microframe.
- The receive FIFO can hold at least one maximum-packet-size packet and two status DWORDs per packet.

[Normal Interrupt IN Split Operation]

The sequence of operations in the figure above (channel 2 {ch_2]) is as follows:

1. Initialize and enable channel 2 for start split as explained in “Channel Initialization”.

2. The otg host writes an IN request to the Request queue as soon as channel 2 receives the grant from the arbiter.

3. The otg host attempts to send the start split IN token in the beginning of the next odd microframe.

4. The otg host generates the ChHltd interrupt after successfully transmitting the start split IN token.

5. In response to the ChHltd interrupt, set the HCSPLT2.ComplSplt bit to send the complete split.

6. As soon the packet is received successfully, the otg host starts writing the data to the system memory.

7. The otg host generates the ChHltd interrupt after transferring the received data to the system memory.

8. In response to the ChHltd interrupt, de-allocate or reinitialize the channel for the next start split.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for interrupt IN split transaction in DMA mode is shown in the table above .

15.6.5.23 Isochronous OUT Split Transactions in Slave Mode

A typical isochronous OUT split operation in Slave mode is shown in the figure below. See channel 1 (ch_1). The assumptions are:

- The application is attempting to transmit a 376-byte packet in an odd microframe.

[Normal Isochronous OUT Split Operation]

The sequence of operations in the figure below (channel 1 [ch_1]) is as follows:

1. Initialize and enable channel 1 for start split (begin) as explained in “Channel Initialization”. The application must set the HCCHAR1.OddFrm bit. Program the MPS field with 188 bytes.

2. Write the packet for channel 1.

3. Along with the last DWORD write of the packet, the otg host writes an entry to the Periodic Request Queue.

4. The otg host attempts to send an OUT token in the next odd microframe.

5. The otg host generates an ACK interrupt as soon as the packet is transmitted successfully.

6. In response to the ACK interrupt, reinitialize the registers to send the start split (end).

7. The otg host generates an ACK interrupt after successfully completing the start split (end) transaction.

8. In response to the ACK interrupt, de-allocate the channel.

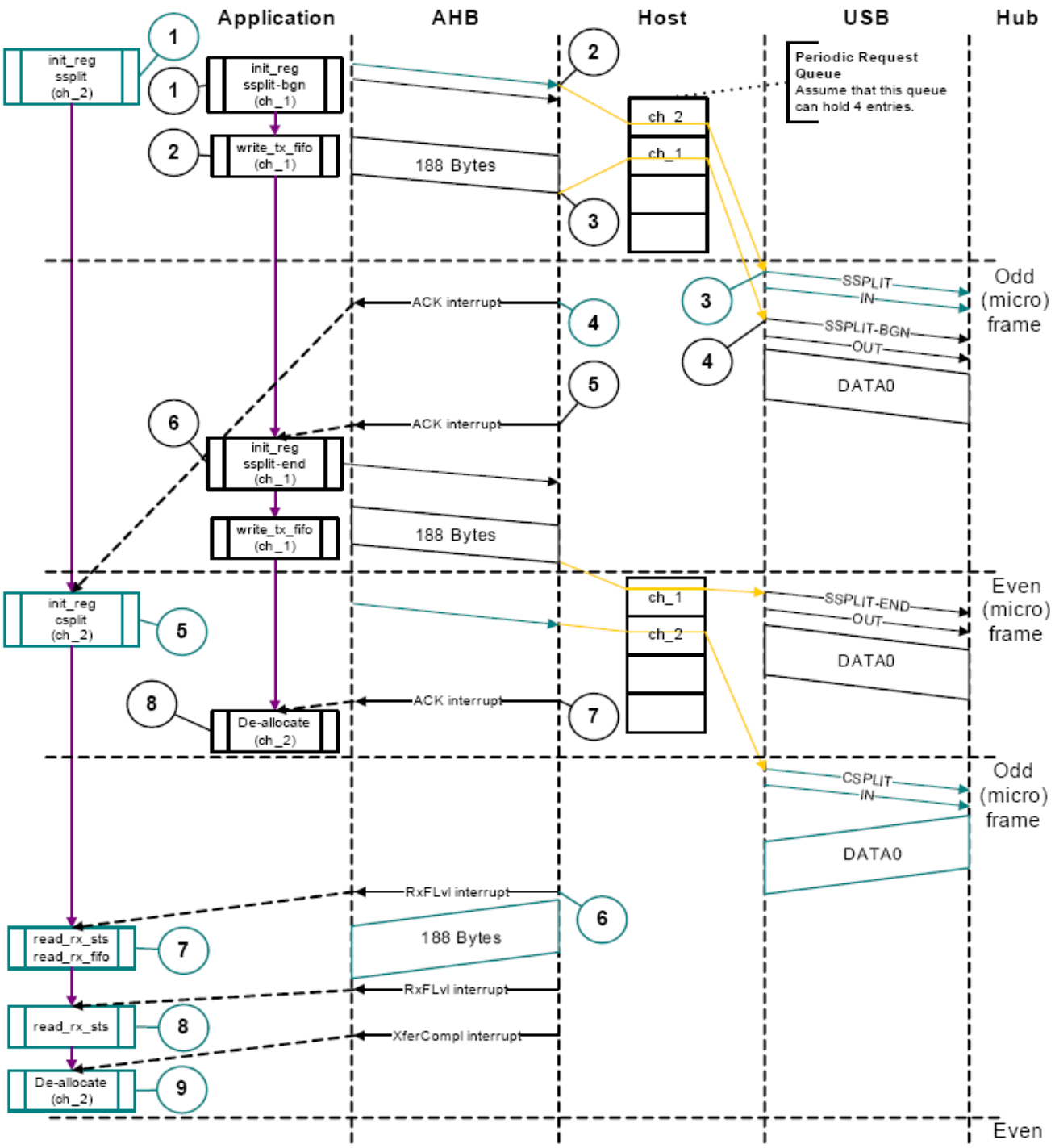


Figure 15.18 Normal Isochronous OUT/IN Split Transactions in Slave Mode

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for isochronous OUT split transaction in Slave mode is shown in table below.

Table 15.13 Interrupt Service Routine for Isochronous OUT/IN Split Transactions in Slave Mode

	Isochronous OUT (Split)	Isochronous IN (Split)

Start Split	Unmask (XferCompl)
	if (XferCompl)
	{
	Do Next Start Split (in next b_interval - 1 uF)
	}
	else if (FrmOvrun)
		Unmask (ACK)
		if (ACK)
		{
		Do Complete Split
		}
		else if (FrmOvrun)

	{
	Do Next Transaction in next frame.
	}
		{
		Do Next Transaction in next frame.
		}

Complete Split	Not Applicable	Unmask (XactErr/NYET/STALL/XferCompl/FrmOvrun/BblErr)
		if (XferCompl)
		{
		De-allocate Channel
		}
		else if (NYET)
		{
		Do Next Complete Split
		}
		else if (STALL or FrmOvrun or BblErr)
		{
		Unmask ChHltd
		Disable Channel
		}
		else if (XactErr)
		{
		Rewind Buffer Pointers
		if (HCCHARn.EC == 3) // ERR response received
		{
		Record ERR error
		Do Next Start Split (in next frame)
		}
		else
		{
		Unmask ChHltd
		Disable Channel
		}
		}
		else if (ChHltd)
		{
		Mask ChHltd
		De-allocate Channel
		}

Note

The otg host keeps track of error count in EC field for periodic split transactions. If the EC field matches the original programmed error count after XactErr interrupt, the application must treat the XactErr as “ERR response received”. The EC field indicates the number of immediate retries that the otg host has performed before generating the XactErr interrupt.

15.6.5.24 Isochronous IN Split Transactions in Slave Mode

A typical isochronous IN split operation in Slave mode is shown in the figure above. See channel 2 (ch_2). The assumptions are:

- The application is attempting to receive one maximum-packet-size packet in an odd microframe.
- The receive FIFO can hold at least one maximum-packet-size packet and two status DWORDs per packet.

[Normal Isochronous IN Split Operation]

The sequence of operations in the figure above (channel 2) is as follows:

1. Initialize and enable channel 2 for start split as explained in “Channel Initialization”. The application must set the HCCHAR2.OddFrm bit.

2. The otg host writes the start split request to the Periodic Request Queue as soon as the HCCHAR2 register is written.

3. The otg host attempts to send the start split IN token in the next odd microframe.

4. As soon as the IN packet is transmitted, the otg host generates an ACK interrupt.

5. In response to the ACK interrupt, set the Do Complete Split bit (HCSPLT2.CompSplt) in the next microframe to send the complete split token in the next odd microframe.

6. As soon as the received packet is written to the receive FIFO, the otg host generates the RxFLvl interrupt.

7. In response to the RxFLvl interrupt, read the received packet status to determine the number of bytes received, then read the receive FIFO accordingly. The application must mask the RxFLvl interrupt before reading the receive FIFO and unmask it after reading the entire packet.

8. The core generates the RxFLvl interrupt for the transfer completion status entry in the receive FIFO. The application must read the receive packet status and if the receive packet status is not an IN data packet (GRXSTSR.PktSts != 4'b0010), ignore it.

9. The core generates the XferCompl interrupt as soon as the receive packet status is read. In response to the XferCompl interrupt, read the HCTSIZ2.PktCnt field. If HCTSIZ2.PktCnt != 0, disable the channel (as explained in "Halting a Channel" on page 216) before re-initializing the channel for the next transfer, if any. If HCTSIZ2.PktCnt == 0, reinitialize the channel for the next transfer. This time, the application must reset the HCCHAR2.OddFrm bit.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for isochronous IN split transaction in Slave mode is shown in the table above.

15.6.5.25 Isochronous OUT Split Transactions in DMA Mode

A typical isochronous OUT split operation in DMA mode is shown in figure below. See channel 1 (ch_1). The assumption is that the application is attempting to transmit a 376-byte packet in an odd microframe.

[Normal Isochronous OUT Split Operation]

The sequence of operations in figure below (channel 1) is as follows:

1. Initialize and enable channel 1 for start split (begin) as explained in "Channel Initialization". The application must set the HCCHAR1.OddFrm bit. Program the MPS field with 188 bytes.

2. The otg host starts reading the packet.

3. After successfully transmitting the start split (begin), the otg host generates the ChHltd interrupt.

4. In response to the ChHltd interrupt, reinitialize the registers to send the start split (end).

5. After successfully transmitting the start split (end), the _otg host generates a ChHltd interrupt.

6. In response to the ChHltd interrupt, de-allocate the channel.

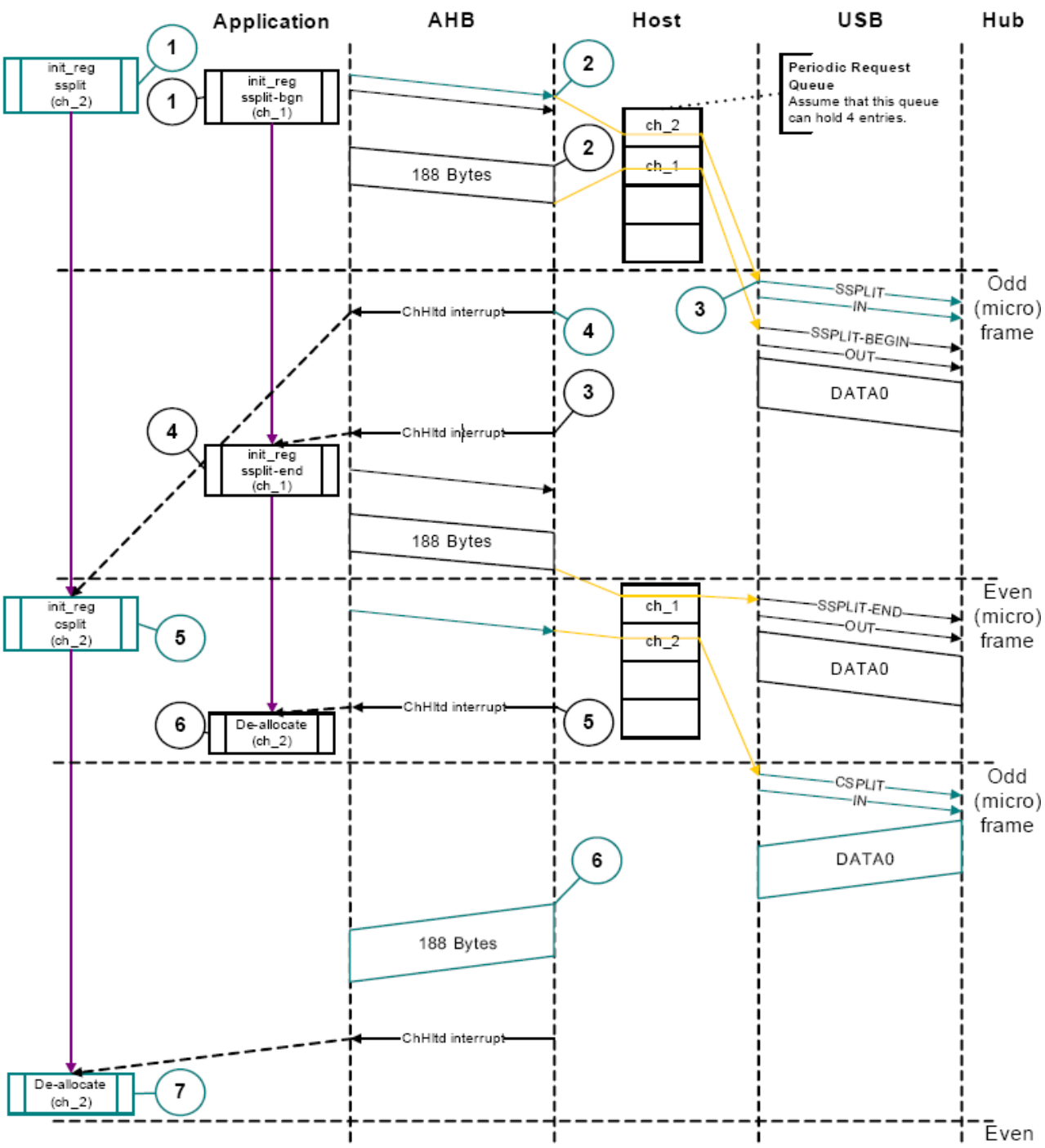


Figure 15.19 Normal Isochronous OUT/IN Split Transactions in DMA Mode

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for an isochronous OUT split transaction in DMA mode is shown in table below.

	Isochronous OUT (Split)	Isochronous IN (Split)

Start Split	Unmask (ChHltd)
	if (ChHltd)
	{
	if (ACK)
	{
	Do Next Start Split (in next b_interval – 1 uF)
	}
	else if (FrmOvrun)
		Unmask (ChHltd)
		if (ChHltd)
		{
		if (ACK)
		{
		Do Complete Split
		}
		else if (FrmOvrun)

	{
	Do Next Transaction in next frame.
	}
	}
		{
		Rewind Buffer Pointers
		Retry Start Split (in next b_interval – 1 uF)
		}
		}

Complete Split	Not Applicable	Unmask (ChHltd)
		if (ChHltd)
		{
		if (XferCompl)
		{
		De-allocate Channel
		}
		else if (NAK)
		{
		Retry Start Split (in next b_interval – 1 uF)
		}
		else if (NYET)
		{
		Do Next Complete Split
		}
		else if (STALL or FrmOvrun or BblErr)
		{
		De-allocate Channel
		}
		else if (XactErr)
		{
		Rewind Buffer Pointers
		if (HCCHARn.EC = = 3) // ERR response received
		{
		Record ERR error
		Do Next Start Split (in next frame)
		}
		else
		{
		De-allocate Channel
		}
		}
		}

Note

The otg host keeps track of error count in EC field for periodic split transactions. If the EC field matches the original programmed error count after XactErr interrupt, the application must treat the XactErr as “ERR response received”. The EC field indicates the number of immediate retries that the OTG host has performed before generating the XactErr interrupt.

15.6.5.26 Isochronous IN Split Transactions in DMA Mode

A typical isochronous IN split operation in DMA mode is shown in the figure above. See channel 2 (ch_2). The assumptions are:

- The application is attempting to receive one maximum-packet-size packet in an odd microframe.
- The receive FIFO can hold at least one maximum-packet-size packet and two status DWORDs per packet.

[Normal Isochronous IN Split Operation]

The sequence of operations in the figure above (channel 2) is as follows:

1. Initialize and enable channel 2 for start split as explained in “Channel Initialization”.

2. The otg host writes an IN request to the Request queue as soon as channel 2 receives the grant from the arbiter.

3. The otg host attempts to send the start split IN token in the beginning of the next odd microframe.

4. The otg host generates the ChHltd interrupt after successfully transmitting the start split IN token.

5. In response to the ChHltd interrupt, set the HCSPLT2.ComplSplt bit to send the complete split.

6. As soon the packet is received successfully, the otg host starts writing the data to the system memory.

7. The otg host generates the ChHltd interrupt after transferring the received data to the system memory. In response to the ChHltd interrupt, de-allocate the channel or reinitialize the channel for the next start split.

[Handling Non-ACK Responses]

The channel-specific interrupt service routine for isochronous IN split transaction in DMA mode is shown in the table above.

15.6.6 Selecting the Queue Depth

Choose the Periodic and Non-periodic Request Queue depths carefully to match the number of periodic/non-periodic endpoints accessed.

The Non-periodic Request Queue depth affects the performance of non-periodic transfers. The deeper the queue (along with sufficient FIFO size), the more often the core is able to pipeline non-periodic transfers. If the queue size is small, the core is able to put in new requests only when the queue space is freed up.

The core's Periodic Request Queue depth is critical to performing periodic transfers as scheduled. Select the periodic queue depth, based on the number of periodic transfers scheduled in a microframe. In Slave mode, however, the application must also take into account the disable entry that must be put into the queue. So, if there are two non-high bandwidth periodic endpoints, the Periodic Request Queue depth must be at least 4. If at least one high-bandwidth endpoint supported, the queue depth must be 8. If the Periodic Request Queue depth is smaller than the periodic transfers scheduled in a microframe, a frame overrun condition results.

15.6.7 Handling Babble Conditions

The otg handles two cases of babble: packet babble and port babble. Packet babble occurs if the device sends more data than the maximum packet size for the channel. Port babble occurs if the core continues to receive data from the device at EOF2 (the end of frame 2, which is very close to SOF).

When OTG detects a packet babble, it stops writing data into the Rx buffer and waits for the end of packet (EOP). When it detects an EOP, it flushes already-written data in the Rx buffer and generates a Babble interrupt to the application.

When OTG detects a port babble, it flushes the RxFIFO and disables the port. The core then generates a Port Disabled Interrupt (GINTSTS.PrtInt, HPRT.PrtEnChng). On receiving this interrupt, the application must determine that this is not due to an overcurrent condition (another cause of the Port Disabled interrupt) by checking HPRT.PrtOvrCurrAct, then perform a soft reset. The core does not send any more tokens after it has detected a port babble condition.

15.7 Device Programming Model

15.7.1 Endpoint Initialization

15.7.1.1 Initialization on USB Reset

1. Set the NAK bit for all OUT endpoints

- DOEPCTLn.SNAK = 1 (for all OUT endpoints)

2. Unmask the following interrupt bits

- DAINTMSK.INEP0 = 1 (control 0 IN endpoint)
- DAINTMSK.OUTEP0 = 1 (control 0 OUT endpoint)
- DOEPMSK.SETUP = 1
- DOEPMSK.XferCompl = 1
- DIEPMSK.XferCompl = 1
- DIEPMSK.TimeOut = 1

3. To transmit or receive data, the device must initialize more registers as specified in "Device DMA/Slave Mode Initialization".

4. Set up the Data FIFO RAM for each of the FIFOs

- Program the GRXFSIZ Register, to be able to receive control OUT data and setup data. If thresholding is not enabled, at a minimum, this must be equal to 1 max packet size of control endpoint 0 + 2 DWORDs (for the status of the control OUT data packet) + 10 DWORDs (for setup packets). If thresholding is enabled, at a minimum, this must be equal to 2* (Rx_threshold_length/4 +1)+ 2 DWORDs (for the status of the control OUT data packet) + 10 DWORDs (for setup packets)
- Program the GNPTXFSIZ Register in Shared FIFO operation or dedicated FIFO size register (depending on the FIFO number chosen) in Dedicated FIFO operation, to be able to transmit control IN data. At a minimum, this must be equal to 1 max packet size of control endpoint 0.If thresholding is enabled, this can be programmed to less than one max packet size.

5. Program the following fields in the endpoint-specific registers for control OUT endpoint 0 to receive a SETUP packet

- DOEPTSIZ0.SetUP Count = 3 (to receive up to 3 back-to-back SETUP packets)
- In DMA mode, DOEPDMA0 register with a memory address to store any SETUP packets received

At this point, all initialization required to receive SETUP packets is done, except for enabling control OUT endpoint 0 in DMA mode.

15.7.1.2 Initialization on Enumeration Completion

1. On the Enumeration Done interrupt (GINTSTS.EnumDone, read the DSTS register to determine the enumeration speed.

2. Program the DIEPCTL0.MPS field to set the maximum packet size. This step configures control endpoint 0. The maximum packet size for a control endpoint depends on the enumeration speed.

3. In DMA mode, program the DOEPCTL0 register to enable control OUT endpoint 0, in order to receive a SETUP packet

- DOEPCTL0.EPEna = 1

At this point, the device is ready to receive SOF packets and is configured to perform control transfers on control endpoint 0.

15.7.1.3 Initialization on SetAddress Command

This section describes what the application must do when it receives a SetAddress command in a SETUP packet.

1. Program the DCFG register with the device address received in the SetAddress command

2. Program the core to send out a status IN packet.

15.7.1.4 Initialization on SetConfiguration/SetInterface Command

This section describes what the application must do when it receives a SetConfiguration or SetInterface command in a SETUP packet.

1. When a SetConfiguration command is received, the application must program the endpoint registers to configure them with the characteristics of the valid endpoints in the new configuration.

2. When a SetInterface command is received, the application must program the endpoint registers of the endpoints affected by this command.

3. Some endpoints that were active in the prior configuration or alternate setting are not valid in the new configuration or alternate setting. These invalid endpoints must be deactivated.

4. For details on a particular endpoint's activation or deactivation, see "Endpoint Activation" and "Endpoint Deactivation".

5. Unmask the interrupt for each active endpoint and mask the interrupts for all inactive endpoints in the DAINTMSK register.

6. Set up the Data FIFO RAM for each FIFO (only if Dynamic FIFO Sizing is enabled). See "Data FIFO RAM Allocation" for more detail.

7. After all required endpoints are configured, the application must program the core to send a status IN packet.
At this point, the device core is configured to receive and transmit any type of data packet.

15.7.1.5 Endpoint Activation

This section describes the steps required to activate a device endpoint or to configure an existing device endpoint to a new type.

1. Program the characteristics of the required endpoint into the following fields of theDIEPCTLn register (for IN or bidirectional endpoints) or the DOEPCTLn register (for OUT or bidirectional endpoints).

- Maximum Packet Size
- USB Active Endpoint = 1
- Endpoint Start Data Toggle (for interrupt and bulk endpoints)
- Endpoint Type
- TxFIFO Number1

2. Once the endpoint is activated, the core starts decoding the tokens addressed to that endpoint and sends out a valid handshake for each valid token received for the endpoint.

15.7.1.6 Endpoint Deactivation

This section describes the steps required to deactivate an existing endpoint.

1. In the endpoint to be deactivated, clear the USB Active Endpoint bit in the DIEPCTLn register (for IN or bidirectional endpoints) or the DOEPCTLn register (for OUT or bidirectional endpoints).

2. Once the endpoint is deactivated, the core ignores tokens addressed to that endpoint, resulting in a timeout on the USB.

15.7.1.7 Device DMA/Slave Mode Initialization

The application must meed the following conditions to set up the device core to handle traffic.

- In Slave mode, GINTMSK.NPTxFEmpMsk, and GINTMSK.RxFLvlMsk must be unset.
- In DMA mode, the aforementioned interrupts must be masked.

15.7.2 Operational Model

15.7.2.1 SETUP and OUT Data Transfers

This section describes the internal data flow and application-level operations during data OUT transfers and SETUP transactions.

[Packet Read in Slave Mode]

This section describes how to read packets (OUT data and SETUP packets) from the receive FIFO in Slave mode.

1. On catching a GINTSTS.RxFLvl interrupt, the application must read the Receive Status Pop register (GRXSTSP).

2. The application can mask the GINTSTS.RxFLvl interrupt by writing to GINTMSK.RxFLvl = 1'b0, until it has read the packet from the receive FIFO.

3. If the received packet's byte count is not 0, the byte count amount of data is popped from the receive Data FIFO and stored in memory. If the received packet byte count is 0, no data is popped from the Receive Data FIFO.

4. The receive FIFO's packet status readout indicates one of the following.

- Global OUT NAK Pattern: PktSts = Global OUT NAK, BCnt = 11'h000, EPNum = Dont Care (4'h0), DPID = Dont Care (2'b00). This data indicates that the global OUT NAK bit has taken effect.
- SETUP Packet Pattern: PktSts = SETUP, BCnt = 11'h008, EPNum = Control EPNum, DPID = D0. This data indicates that a SETUP packet for the specified endpoint is now available for reading from the receive FIFO.
- Setup Stage Done Pattern: PktSts = Setup Stage Done, BCnt = 11'h0, EPNum = Control EP Num, DPID = Don't Care (2'b00). This data indicates that the Setup stage for the specified endpoint has completed and the Data stage has started. After this entry is popped from the receive FIFO, the core asserts a Setup interrupt on the specified control OUT endpoint.
- Data OUT Packet Pattern: PktSts = DataOUT, BCnt = size of the Received data OUT packet ($0 \leq$ BCnt $\leq$ 1, 024), EPNum = EPNum on which the packet was received, DPID = Actual Data PID.
- Data Transfer Completed Pattern: PktSts = Data OUT Transfer Done, BCnt = 11'h0, EPNum = OUT EP Num on which the data transfer is complete, DPID = Dont Care (2'b00). This data indicates that a OUT data transfer for the specified OUT endpoint has completed. After this entry is popped from the receive FIFO, the core asserts a Transfer Completed interrupt on the specified OUT endpoint.

The encoding for the PktSts is listed in "Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP)".

5. After the data payload is popped from the receive FIFO, the GINTSTS.RxFLvl interrupt must be unmasked.

6. Steps 1–5 are repeated every time the application detects assertion of the interrupt line due to GINTSTS.RxFLvl. Reading an empty receive FIFO can result in undefined core behavior.

Below figure provides a flow chart of the above procedure.

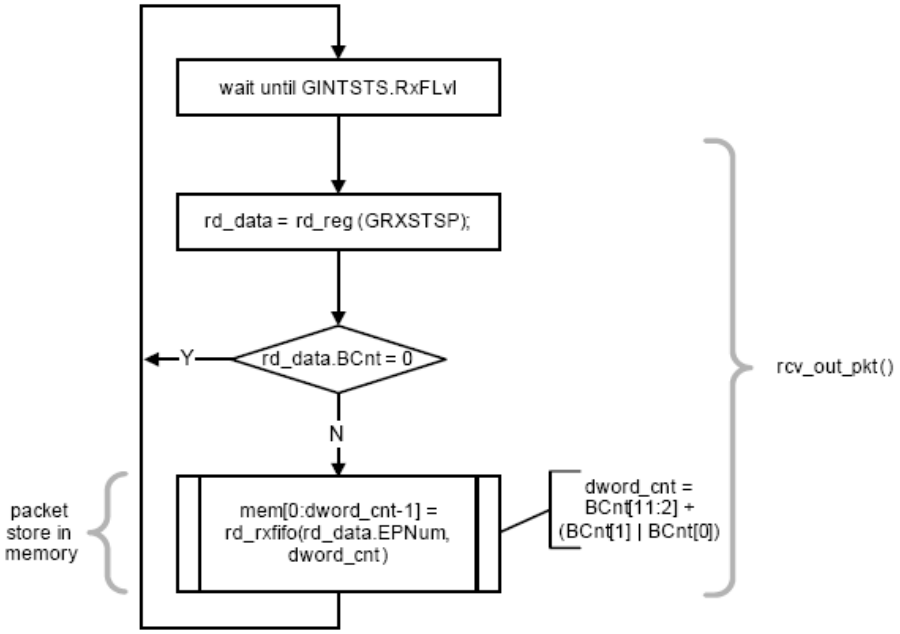


Figure 15.20 Receive FIFO Packet Read in Slave Mode

[SETUP Transactions]

This section describes how the core handles SETUP packets and the application's sequence for handling SETUP transactions.

Application Requirements

1. To receive a SETUP packet, the DOEPTSIZn.SUPCnt field in a control OUT endpoint must be programmed to a non-zero value. When the application programs the SUPCnt field to a non-zero value, the core receives SETUP packets and writes them to the receive FIFO, irrespective of the DOEPCTLn.NAK status and DOEPCTLn.EPEna bit setting. The SUPCnt field is decremented every time the control endpoint receives a SETUP packet. If the SUPCnt field is not programmed to a proper value before receiving a SETUP packet, the core still receives the SETUP packet and decrements the SUPCnt field, but the application possibly is not be able to determine the correct number of SETUP packets received in the Setup stage of a control transfer.

- DOEPTSIZn.SUPCnt = 3

2. In DMA mode, the OUT endpoint must also be enabled, to transfer the received SETUP packet data from the internal receive FIFO to the external memory.

- DOEPCTLn.EPEna = 1'b1

3. The application must always allocate some extra space in the Receive Data FIFO, to be able to receive up to three SETUP packets on a control endpoint.

- The space to be Reserved is (4 * n) + 6 DWORDs, where n is the number of control endpoints supported by the device. Three DWORDs are required for the first SETUP packet, 1 DWORD is required for the Setup Stage Done DWORD, and 6 DWORDs are required to store two extra SETUP packets among all control endpoints.
- 3 DWORDs per SETUP packet are required to store 8 bytes of SETUP data and 4 bytes of SETUP status (Setup Packet Pattern). The core reserves this space in the receive data.
- FIFO to write SETUP data only, and never uses this space for data packets.

4. In Slave mode, the application must read the 2 DWORDs of the SETUP packet from the receive FIFO. In DMA mode, the core writes the 2 DWORDs of SETUP data to the memory.

5. The application must read and discard the Setup Stage Done DWORD from the receive FIFO.

Internal Data Flow

1. When a SETUP packet is received, the core writes the received data to the receive FIFO, without checking for available space in the receive FIFO and irrespective of the endpoint's NAK and Stall bit settings.

- The core internally sets the IN NAK and OUT NAK bits for the control IN/OUT endpoints on which the SETUP packet was received.

2. For every SETUP packet received on the USB, 3 DWORDs of data is written to the receive FIFO, and the SUPCnt field is decremented by 1.

- The first DWORD contains control information used internally by the core
- The second DWORD contains the first 4 bytes of the SETUP command
- The third DWORD contains the last 4 bytes of the SETUP command

3. When the Setup stage changes to a Data IN/OUT stage, the core writes an entry (Setup Stage Done DWORD) to the receive FIFO, indicating the completion of the Setup stage.

4. On the AHB side, SETUP packets are emptied either by the DMA or the application. In DMA mode, the SETUP packets (2 DWORDs) are written to the memory location programmed in the DOEPDMAn register, only if the endpoint is enabled. If the endpoint is not enabled, the data remains in the receive FIFO until the enable bit is set.

5. When either the DMA or the application pops the Setup Stage Done DWORD from the receive FIFO, the core interrupts the application with a DOEPINTn.SETUP interrupt, indicating it can process the received SETUP packet.

- The core clears the endpoint enable bit for control OUT endpoints.

Application Programming Sequence

1. Program the DOEPTSIZn register.

- DOEPTSIZn.SUPCnt = 3

2. In DMA mode, program the DOEPDMAn register and DOEPCTLn register with the endpoint characteristics and set the Endpoint Enable bit (DOEPCTLn.EPEna).

- Endpoint Enable = 1

3. In Slave mode, wait for the GINTSTS.RxFLvl interrupt and empty the data packets from the receive FIFO, as explained in "Packet Read in Slave Mode". This step can be repeated many times.

4. Assertion of the DOEPINTn.SETUP interrupt marks a successful completion of the SETUP Data Transfer.

- On this interrupt, the application must read the DOEPTSIZn register to determine the number of SETUP packets received and process the last received SETUP packet.
- In DMA mode, the application must also determine if the interrupt bit DOEPINTn.Back2BackSETup is set. This bit is set if the core has received more than three back-to-back SETUP packets. If this is the case, the application must ignore the DOEPTSIZn.SUPCnt value and use the DOEPDMAn directly to read out the last SETUP packet received. DOEPDMAn-8 provides the pointer to the last valid SETUP data.

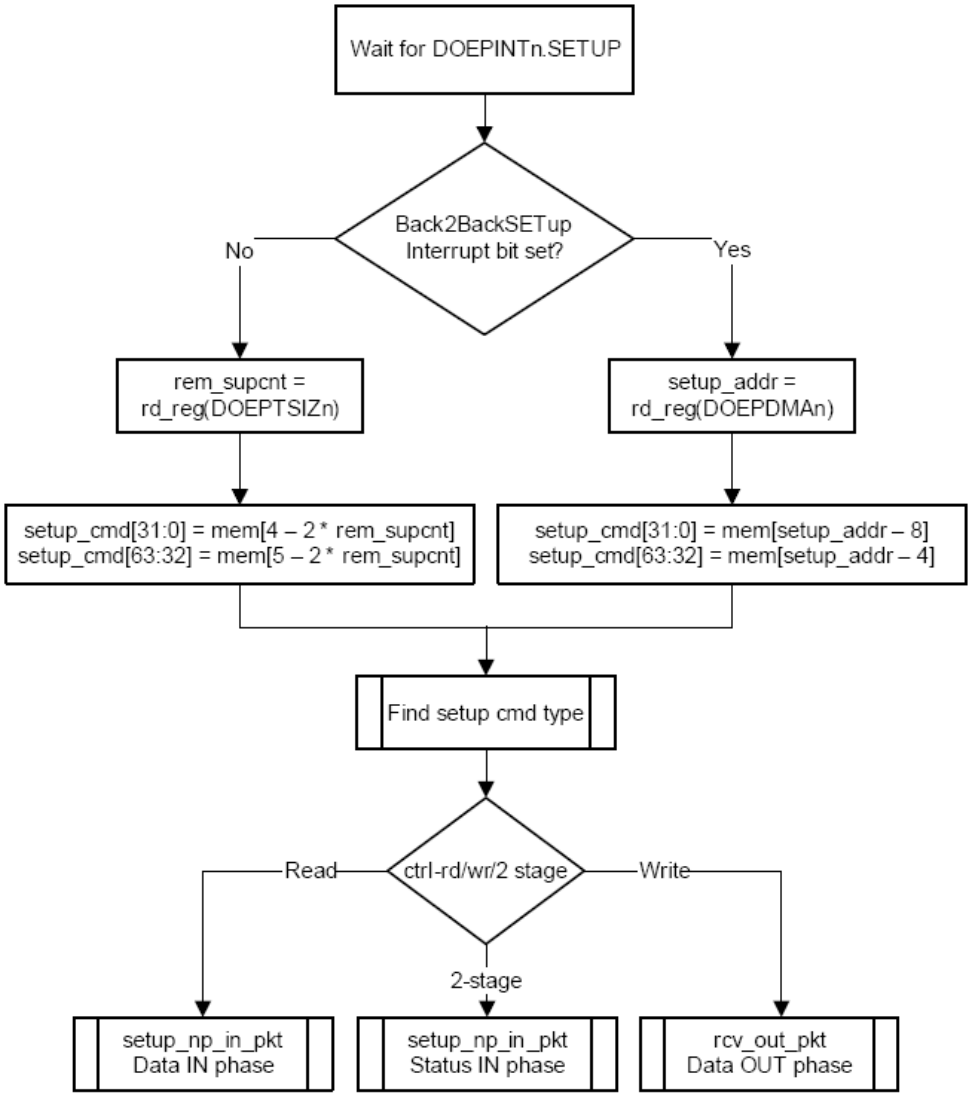


Figure 15.21 Processing a SETUP Packet

[Handling More Than Three Back-to-Back SETUP Packets]

Per the USB 2.0 specification, normally, during a SETUP packet error, a host does not send more than three back-to-back SETUP packets to the same endpoint. However, the USB 2.0 specification does not limit the number of back-to-back SETUP packets a host can send to the same endpoint. When this condition occurs, the OTG core generates an interrupt (DOEPINTn.Back2BackSETup). In DMA mode, the core also rewinds the DMA address for that endpoint (DOEPDMAn) and overwrites the first SETUP packet in system memory with the fourth, second with the fifth, and so on. If the Back2BackSETup interrupt is asserted, the application must read the OUT endpoint DMA register (DOEPDMAn) to determine the final SETUP data in system memory.

In DMA mode, the application can mask the Back2BackSETup interrupt, but after receiving the DOEPINT.SETUP interrupt, the application can read the DOEPINT.Back2BackSETup interrupt bit. In Slave mode, the application can use the GINTSTS.RxFLvl interrupt to read out the SETUP packets from the FIFO whenever the core receives the SETUP packet.

[Setting the Global OUT NAK]

Internal Data Flow

1. When the application sets the Global OUT NAK (DCTL.SGOUTNak), the core stops writing data, except SETUP packets, to the receive FIFO. Irrespective of the space availability in the receive FIFO, non-isochronous OUT tokens receive a NAK handshake response, and the core ignores isochronous OUT data packets

2. The core writes the Global OUT NAK pattern to the receive FIFO. The application must reserve enough receive FIFO space to write this data pattern. See “Data FIFO RAM Allocation”.

3. When either the core (in DMA mode) or the application (in Slave mode) pops the Global OUT NAK pattern DWORD from the receive FIFO, the core sets the GINTSTS.GOUTNakEff interrupt.

4. Once the application detects this interrupt, it can assume that the core is in Global OUT NAK mode. The application can clear this interrupt by clearing the DCTL.SGOUTNak bit.

Application Programming Sequence

1. To stop receiving any kind of data in the receive FIFO, the application must set the Global OUT NAK bit by programming the following field.

- DCTL.SGOUTNak = 1'b1

2. Wait for the assertion of the interrupt GINTSTS.GOUTNakEff. When asserted, this interrupt indicates that the core has stopped receiving any type of data except SETUP packets.

3. The application can receive valid OUT packets after it has set DCTL.SGOUTNak and before the core asserts the GINTSTS.GOUTNakEff interrupt.

4. The application can temporarily mask this interrupt by writing to the GINTMSK.GINNakEffMsk bit.

- GINTMSK.GINNakEffMsk = 1'b0

5. Whenever the application is ready to exit the Global OUT NAK mode, it must clear the DCTL.SGOUTNak bit. This also clears the GINTSTS.GOUTNakEff interrupt.

- DCTL.CGOUTNak = 1'b1

6. If the application has masked this interrupt earlier, it must be unmasked as follows:

- GINTMSK.GINNakEffMsk = 1'b1

[Disabling an OUT Endpoint]

The application must use this sequence to disable an OUT endpoint that it has enabled.

Application Programming Sequence

1. Before disabling any OUT endpoint, the application must enable Global OUT NAK mode in the core, as described in "Setting the Global OUT NAK".

- DCTL.DCTL.SGOUTNak = 1'b1

2. Wait for the GINTSTS.GOUTNakEff interrupt

3. Disable the required OUT endpoint by programming the following fields.

- DOEPCTLn.EPDisable = 1'b1
- DOEPCTLn.SNAK = 1'b1

4. Wait for the DOEPINTn.EPDisabled interrupt, which indicates that the OUT endpoint is completely disabled. When the EPDisabled interrupt is asserted, the core also clears the following bits.

- DOEPCTLn.EPDisable = 1'b0
- DOEPCTLn.EPEnable = 1'b0

5. The application must clear the Global OUT NAK bit to start receiving data from other non-disabled OUT endpoints.

- DCTL.SGOUTNak = 1'b0

[Generic Non-Isochronous OUT Data Transfers without Thresholding]

This section describes a regular non-isochronous OUT data transfer (control, bulk, or interrupt).

Application Requirements

1. Before setting up an OUT transfer, the application must allocate a buffer in the memory to accommodate all data to be received as part of the OUT transfer, then program that buffer's size and start address (in DMA mode) in the endpoint-specific registers.

2. For OUT transfers, the Transfer Size field in the endpoint's Transfer Size register must be a multiple of the maximum packet size of the endpoint, adjusted to the DWORD boundary.

- transfer size[epnum] = n * (mps[epnum] + 4 – (mps[epnum] mod 4))
- packet count[epnum] = n
- n > 0

3. In DMA mode, the core stores a received data packet in the memory, always starting on a DWORD boundary. If the maximum packet size of the endpoint is not a multiple of 4, the core inserts byte pads at end of a maximum-packet-size packet up to the end of the DWORD.

4. On any OUT endpoint interrupt, the application must read the endpoint's Transfer Size register to calculate the size of the payload in the memory. The received payload size can be less than the programmed transfer size.

- Payload size in memory = application-programmed initial transfer size – core updated final transfer size
- Number of USB packets in which this payload was received = applicationprogrammed initial packet count – core updated final packet count

Internal Data Flow

1. The application must set the Transfer Size and Packet Count fields in the endpointspecific registers, clear the NAK bit, and enable the endpoint to receive the data.

2. Once the NAK bit is cleared, the core starts receiving data and writes it to the receive FIFO, as long as there is space in the receive FIFO. For every data packet received on the USB, the data packet and its status are written to the receive FIFO. Every packet (maximum packet size or short packet) written to the receive FIFO decrements the Packet Count field for that endpoint by 1.

- OUT data packets received with Bad Data CRC are flushed from the receive FIFO automatically.
- After sending an ACK for the packet on the USB, the core discards non-isochronous OUT data packets that the host, which cannot detect the ACK, re-sends. The application does not detect multiple back-to-back data OUT packets on the same endpoint with the same data PID. In this case the packet count is not decremented.
- If there is no space in the receive FIFO, isochronous or non-isochronous data packets are ignored and not written to the receive FIFO. Additionally, non-isochronous OUT tokens receive a NAK handshake reply.
- In all the above three cases, the packet count is not decremented because no data is written to the receive FIFO.

3. When the packet count becomes 0 or when a short packet is received on the endpoint, the NAK bit for that endpoint is set. Once the NAK bit is set, the isochronous or nonisochronous data packets are ignored and not written to the receive FIFO, and nonisochronous OUT tokens receive a NAK handshake reply.

4. After the data is written to the receive FIFO, either the application (in Slave mode) or the core's DMA engine (in External or Internal DMA mode), reads the data from the receive FIFO and writes it to external memory, one packet at a time per endpoint.

5. At the end of every packet write on the AHB to external memory, the transfer size for the endpoint is decremented by the size of the written packet.

6. The OUT Data Transfer Completed pattern for an OUT endpoint is written to the receive FIFO on one of the following conditions.

- The transfer size is 0 and the packet count is 0

- The last OUT data packet written to the receive FIFO is a short packet (0≤packet size< maximum packet size)

7. When either the application or the DMA pops this entry (OUT Data Transfer Completed),

- Transfer Completed interrupt is generated for the endpoint and the endpoint enable is cleared.

Application Programming Sequence

1. Program the DOEPTSIZn register for the transfer size and the corresponding packet count. Additionally, in DMA mode, program the DOEPDMAn register.

2. Program the DOEPCTLn register with the endpoint characteristics, and set the Endpoint Enable and ClearNAK bits.

- DOEPCTLn.EPEna = 1
- DOEPCTLn.CNAK = 1

3. In Slave mode, wait for the GINTSTS.Rx StsQ level interrupt and empty the data packets from the receive FIFO as explained in “Packet Read in Slave Mode”.

- This step can be repeated many times, depending on the transfer size.

4. Asserting the DOEPINTn.XferCompl interrupt marks a successful completion of the nonisochronous OUT data transfer.

5. Read the DOEPTSIZn register to determine the size of the received data payload.

[Generic non-Isochronous OUT Data Transfer with Thresholding]

This section describes a regular non-ISO OUT data transfer (Control/Bulk/Intr) when thresholding is enabled.

Application Requirements

Application requirements is the same as without thresholding.

Internal Data Flow

1. The application should set the transfer size and packet count fields in the endpoint specific registers, clear the NAK bit and enable the endpoint to receive the data.

2. Once the NAK bit is cleared, the core starts receiving the data and writes it into the receive FIFO, as long as there is threshold amount of space in the receive FIFO. For every threshold amount of data received on the USB, the data packet and the threshold status are written into the receive FIFO. At the end of a packet, a last threshold status and also a data update status is written into the receive FIFO.If it was the last packet of the transfer, then a transfer complete status is also written into the receive FIFO.On every packet (mps sized or short packet) written into the receive FIFO, the packet count field for that endpoint is decremented by 1.

- OUT data packets received with Bad Data CRC are flushed out of the receive FIFO automatically.The core also rewind the DMA pointers internally.
- non-ISO OUT data packet re-sent by the host, because the ACK was not seen by the host, will be discarded by the core, after sending an ACK for the packet on the USB. The application will not see multiple back to back data OUT packets on the same endpoint, with the same data PID. In this case the packet count is not decremented.
- If there is no space for at least threshold amount of data in the receive FIFO, the ISO/non-ISO data packets are ignored and not written into the receive FIFO. In addition, the non-ISO OUT tokens are responded with NAK handshake.
- If the core sees an overflow case (no space in the fifo in the middle of a packet reception), then the core stops writing the remaining data into the fifo and sends a NAK handshake on the USB. The core rewinds the fifo pointer to the threshold boundary, so that the portion of the threshold data that is in the fifo is flushed out.The core also rewinds the DMA pointers.The core also sets DOEPINTn.OutPktErr (This interrupt bit is mainly used for debug purpose).

In all the above cases, the packet count is not decremented because no data is written into the receive FIFO.

In High Speed, after the core has received a packet, the core sends a NYET handshake if the core does not find threshold amount of free space available in the FIFO.

3. When the packet count becomes 0 or when a short packet is received on the endpoint, the NAK bit for that endpoint is set. Once the NAK bit is set, the ISO/non-ISO data packets are ignored and not written into the receive FIFO and the non-ISO OUT tokens are responded with a NAK handshake.

4. The DMA engine will transfer data from the receive FIFO to the system memory as soon as it sees one threshold amount of data in the FIFO.

5. At the end of every packet write on the AHB into the external memory, the transfer size for the endpoint is decremented by the size of packet written into the memory.

6. The OUT Data Transfer Complete pattern for an OUT endpoint is written into the receive FIFO, on one of the following conditions.

- The last threshold is written into FIFO and the packet count is decremented to 0
- If it is a short packet and the core sees the end of packet within a threshold.

7. When this entry (OUT Data Transfer Complete) is popped out by the DMA engine, Transfer Complete interrupt for the endpoint is generated and the endpoint enable is cleared.

8. “Rewind OUT Data Transfer” pattern is written into the receive FIFO, on one of the following conditions

- On seeing Overflow condition.
- On seeing a CRC error.

9. When this entry (Rewind OUT Data Transfer) is popped out by the DMA engine, it does the DMA pointer rewind.

Application Programming Sequence

This sequence is the same as in non thresholding case.

[Generic Isochronous OUT Data Transfer without Thresholding]

This section describes a regular isochronous OUT data transfer.

Application Requirements

1. All the application requirements for non-isochronous OUT data transfers also apply to isochronous OUT data transfers

2. For isochronous OUT data transfers, the Transfer Size and Packet Count fields must always be set to the number of maximum-packet-size packets that can be received in a single microframe and no more. Isochronous OUT data transfers cannot span more than 1 microframe.

- $1 \leq$ packet count[epnum] ≤ 3

3. In Slave mode, when isochronous OUT endpoints are supported in the device, the application must read all isochronous OUT data packets from the receive FIFO (data and status) before the end of the periodic frame (GINTSTS.EOPF interrupt). In DMA mode, the application must guarantee enough bandwidth to allow emptying the isochronous OUT data packet from the receive FIFO before the end of each periodic frame.

4. To receive data in the following frame/microframe, an isochronous OUT endpoint must be enabled after the GINTSTS.EOPF and before the GINTSTS.SOF.

Internal Data Flow

1. The internal data flow for isochronous OUT endpoints is the same as that for nonisochronous OUT endpoints, but for a few differences.

2. When an isochronous OUT endpoint is enabled by setting the Endpoint Enable and clearing the NAK bits, the Even/Odd frame/microframe bit must also be set appropriately. The core receives data on a isochronous OUT endpoint in a particular microframe only if the following condition is met.

- DOEPCTLn.Even/Odd microframe = DSTS.SOFFN[0]

3. When either the application or the external/internal DMA completely reads an isochronous OUT data packet (data and status) from the receive FIFO, the core updates the DOEPTSIZn.Received DPID field with the data PID of the last isochronous OUT data packet read from the receive FIFO.

Application Programming Sequence

1. Program the DOEPTSIZn register for the transfer size and the corresponding packet count. When in DMA mode, also program the DOEPDMAn register.

2. Program the DOEPCTLn register with the endpoint characteristics and set the Endpoint Enable, ClearNAK, and Even/Odd frame/microframe bits.

- Endpoint Enable = 1
- CNAK = 1
- Even/Odd frame/microframe = (0: Even/1: Odd)

3. In Slave mode, wait for the GINTSTS.Rx StsQ level interrupt and empty the data packets from the receive FIFO as explained in “Packet Read in Slave Mode”.

- This step can be repeated many times, depending on the transfer size.

4. The assertion of the DOEPINTn.XferCompl interrupt marks the completion of the isochronous OUT data transfer. This interrupt does not necessarily mean that the data in memory is good.

- This interrupt can not always be detected for isochronous OUT transfers. Instead, the application can detect the GINTSTS.incomplete Isochronous OUT data interrupt. See “Incomplete Isochronous OUT Data Transfers”, for more details

5. Read the DOEPTSIZn register to determine the size of the received transfer and to determine the validity of the data received in the microframe. The application must treat the data received in memory as valid only if one of the following conditions is met.

- DOEPTSIZn.RxDPID = D0 and the number of USB packets in which this payload was received = 1
- DOEPTSIZn.RxDPID = D1 and the number of USB packets in which this payload was received = 2
- DOEPTSIZn.RxDPID = D2 and the number of USB packets in which this payload was received = 3
  - The number of USB packets in which this payload was received = App Programmed Initial Packet Count – Core Updated Final Packet Count

The application can discard invalid data packets.

[Generic Isochronous OUT Data Transfer with Thresholding]

This section describes a regular isochronous OUT data transfer.

Application Requirements

There is no change in this section from a non-thresholding mode.

Internal Data Flow

1. The internal data flow for isochronous OUT Endpoints when thresholding is enabled, is the same as that for the non isochronous OUT endpoints when thresholding is enabled, but for a few differences.

2. If MAC sees an overflow condition when writing a packet, it stops writing.The current threshold amount of data that is being written into the receive FIFO is flushed out at the end of packet.This will eventually result in GINTSTS.incomplete ISO OUT Data interrupt. Refer to the section Incomplete isochronous OUT Data Transfers, for more details.

3. If MAC sees a CRC error for the receiving packet, the last threshold being written into the receive FIFO is flushed out. This will eventually result in GINTSTS.incomplete isochronous OUT Data interrupt. Refer to the section Incomplete isochronous OUT Data Transfers, for more details.

4. Assertion of DOEPINTn.XferCompl interrupt marks a completion of the isochronous OUT Data Transfer. This interrupt may not necessarily mean that data in the memory is good data.

5. Read the DOEPTSIZn register, to find out the size of the received transfer and to find out the validity of the data received in the microframe. The application should treat the data received into the memory as valid only if one of the following conditions is met. Invalid data packets may be discarded by the application.

- DOEPTSIZn.RxDPID = D0 and Number of USB Packets in which this payload was received = 1
- DOEPTSIZn.RxDPID = D1 and Number of USB Packets in which this payload was received = 2
- DOEPTSIZn.RxDPID = D2 and Number of USB Packets in which this payload was received = 3

• Number of USB Packets in which this payload was received = App Programmed Initial Packet Count - Core Updated Final Packet Count

[Incomplete Isochronous OUT Data Transfers]

This section describes the application programming sequence when isochronous OUT data packets are dropped inside the core.

Internal Data Flow

1. For isochronous OUT endpoints, the DOEPINTn.XferCompl interrupt possibly is not always asserted. If the core drops isochronous OUT data packets, the application could fail to detect the DOEPINTn.XferCompl interrupt under the following circumstances.

- When the receive FIFO cannot accommodate the complete ISO OUT data packet, the core drops the received ISO OUT data. In thresholding this is same as overflow.
- When the isochronous OUT data packet is received with CRC errors
- When the isochronous OUT token received by the core is corrupted
- When the application is very slow in reading the data from the receive FIFO

2. When the core detects an end of periodic frame before transfer completion to all isochronous OUT endpoints, it asserts the GINTSTS.incomplete Isochronous OUT data interrupt, indicating that a DOEPINTn.XferCompl interrupt is not asserted on at least one of the isochronous OUT endpoints. At this point, the endpoint with the incomplete transfer remains enabled, but no active transfers remains in progress on this endpoint on the USB.

Application Programming Sequence

1. Asserting the GINTSTS.incomplete Isochronous OUT data interrupt indicates that in the current microframe, at least one isochronous OUT endpoint has an incomplete transfer.

- If this occurs because isochronous OUT data is not completely emptied from the endpoint, the application must ensure that the DMA or the application empties all isochronous OUT data (data and status) from the receive FIFO before proceeding.
- When all data is emptied from the receive FIFO, the application can detect the DOEPINTn.XferCompl interrupt. In this case, the application must re-enable the endpoint to receive isochronous OUT data in the next microframe, as described in “Generic Isochronous OUT Data Transfer without Thresholding”.

2. When it receives a GINTSTS.incomplete Isochronous OUT data interrupt, the application must read the control registers of all isochronous OUT endpoints (DOEPCTLn) to determine which endpoints had an incomplete transfer in the current microframe. An endpoint transfer is incomplete if both the following conditions are met.

- DOEPCTLn.Even/Odd microframe bit = DSTS.SOFFN[0]
- DOEPCTLn.Endpoint Enable = 1

3. The previous step must be performed before the GINTSTS.SOF interrupt is detected, to ensure that the current microframe number is not changed.

4. For isochronous OUT endpoints with incomplete transfers, the application must discard the data in the memory and disable the endpoint by setting the DOEPCTLn.Endpoint Disable bit.

5. Wait for the DOEPINTn.Endpoint Disabled interrupt and enable the endpoint to receive new data in the next microframe as explained in “Generic Isochronous OUT Data Transfer without Thresholding”.

- Because the core can take some time to disable the endpoint, the application possibly is not able to receive the data in the next microframe after receiving bad isochronous data.

[Stalling a Non-Isochronous OUT Endpoint]

This section describes how the application can stall a non-isochronous endpoint.

1. Put the core in the Global OUT NAK mode, as described in “Setting the Global OUT NAK”.

2. Disable the required endpoint, as described in “Disabling an OUT Endpoint”.

- When disabling the endpoint, instead of setting the DOEPCTL.SNAK bit, set DOEPCTL.STALL = 1.
  - The Stall bit always takes precedence over the NAK bit.

3. When the application is ready to end the STALL handshake for the endpoint, the DOEPCTLn.STALL bit must be cleared.

4. If the application is setting or clearing a STALL for an endpoint due to a SetFeature.Endpoint Halt or ClearFeature.Endpoint Halt command, the Stall bit must be set or cleared before the application sets up the Status stage transfer on the control endpoint.

[Examples]

This section describes and depicts some fundamental transfer types and scenarios.

Slave Mode Bulk OUT Transaction

Figure below depicts the reception of a single Bulk OUT Data packet from the USB to the AHB and describes the events involved in the process.

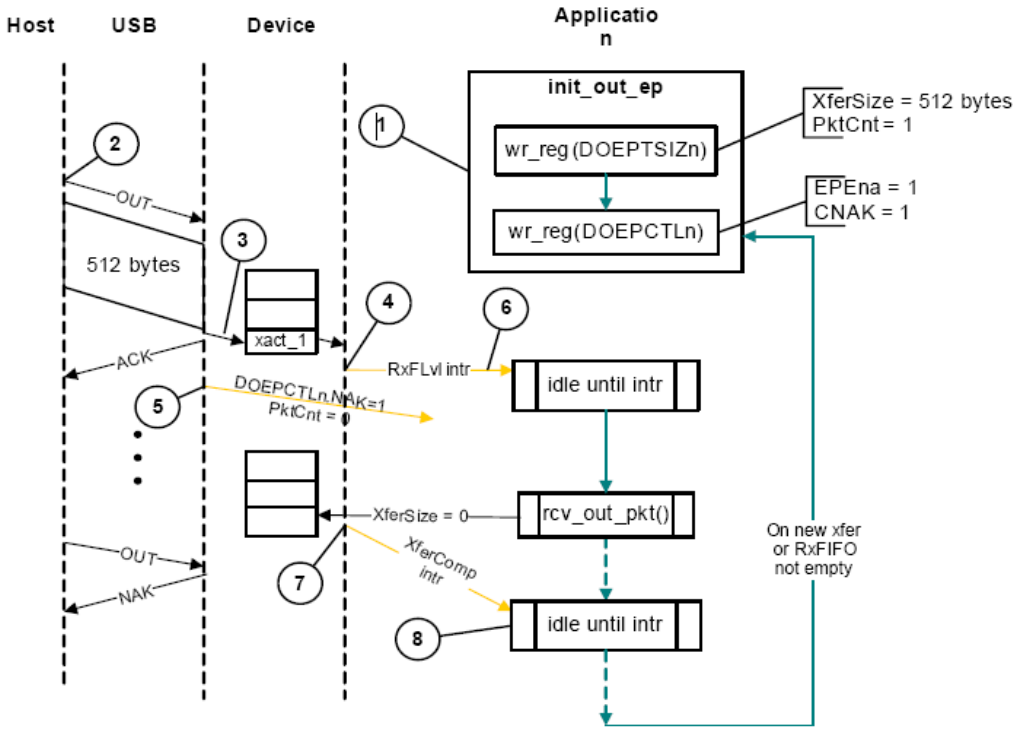


Figure 15.22 Slave Mode Bulk OUT Transaction

1. After a SetConfiguration/SetInterface command, the application initializes all OUT endpoints by setting DOEPCTLn.CNAK = 1 and DOEPCTLn.EPEna = 1, and setting a suitable XferSize and PktCnt in the DOEPTSIZn register.

2. Host attempts to send data (OUT token) to an endpoint.

3. When the core receives the OUT token on the USB, it stores the packet in the RxFIFO because space is available there.

4. After writing the complete packet in the RxFIFO, the core then asserts the GINTSTS.RxFLvl interrupt.

5. On receiving the PktCnt number of USB packets, the core sets the NAK bit for this endpoint internally to prevent it from receiving any more packets.

6. The application processes the interrupt and reads the data from the RxFIFO.

7. When the application has read all the data (equivalent to XferSize), the core generates a DOEPINTn.XferCompl interrupt.

8. The application processes the interrupt and uses the setting of the DOEPINTn.XferCompl interrupt bit to determine that the intended transfer is complete.

15.7.2.2 IN Data Transfers

This section describes IN data transfer details.

[Packet Write in Slave Mode]

This section describes how the application writes data packets to the endpoint FIFO in Slave mode.

1. The application can either choose polling or interrupt mode.

- In polling mode, application monitors the status of the endpoint transmit data FIFO, by reading the DTXFSTSn register, to determine, if there is enough space in the data FIFO.
- In interrupt mode, application waits for the DIEPINTn.TxFEmp interrupt and then reads the DTXFSTSn register, to determine, if there is enough space in the data FIFO.
- To write a single non-zero length data packet, there must be space to write the entire packet is the data FIFO.
- For writing zero length packet, application must not look for FIFO space.

2. Using one of the above mentioned methods, when the application determines that there is enough space to write a transmit packet, the application must first write into the endpoint control register, before writing the data into the data FIFO. The application, typically must do a read modify write on the DIEPCTLn, to avoid modifying the contents of the register, except for setting the Endpoint Enable bit.

The application can write multiple packets for the same endpoint, into the transmit FIFO, if space is available. For periodic IN endpoints, application must write packets only for one microframe. It can write packets for the next periodic transaction, only after getting transfer complete for the previous transaction.

[Setting IN Endpoint NAK]

Internal Data Flow

1. When the application sets the IN NAK for a particular endpoint, the core stops transmitting data on the endpoint, irrespective of data availability in the endpoint's transmit FIFO.

- Non-isochronous IN tokens receive a NAK handshake reply
- Isochronous IN tokens receive a zero-data-length packet reply

2. The core asserts the DIEPINTn.IN NAK Effective interrupt in response to the DIEPCTL.Set NAK bit.

3. Once this interrupt is seen by the application, the application can assume that the endpoint is in IN NAK mode. This interrupt can be cleared by the application by setting the DIEPCTLn. Clear NAK bit.

Application Programming Sequence

1. To stop transmitting any data on a particular IN endpoint, the application must set the IN NAK bit. To set this bit, the

following field must be programmed.

- DIEPCTLn.SetNAK = 1'b1

2. Wait for assertion of the DIEPINTn.NAK Effective interrupt. This interrupt indicates the core has stopped transmitting data on the endpoint.

3. The core can transmit valid IN data on the endpoint after the application has set the NAK bit, but before the assertion of the NAK Effective interrupt.

4. The application can mask this interrupt temporarily by writing to the DIEPMSK.NAK Effective bit.

- DIEPMSK.NAK Effective = 1'b0

5. To exit Endpoint NAK mode, the application must clear the DIEPCTLn.NAK status. This also clears the DIEPINTn.NAK Effective interrupt.

- DIEPCTLn.ClearNAK = 1'b1

6. If the application masked this interrupt earlier, it must be unmasked as follows:

- DIEPMSK.NAK Effective = 1'b1

[IN Endpoint Disable]

Use the following sequence to disable a specific IN endpoint (periodic/non-periodic) that has been previously enabled.

Application Programming Sequence

1. In Slave mode, the application must stop writing data on the AHB, for the IN endpoint to be disabled.

2. The application must set the endpoint in NAK mode. Refer to the section Setting the Endpoint NAK

- DIEPCTLn.SetNAK = 1'b1

3. Wait for DIEPINTn.NAK Effective interrupt.

4. Set the following bits in the DIEPCTLn register for the endpoint that must be disabled.

- DIEPMSK.NAK Effective = 1'b1
- DIEPCTLn.Endpoint Disable = 1
- DIEPCTLn.SetNAK = 1

5. Assertion of DIEPINTn.Endpoint Disabled interrupt indicates that the core has completely disabled the specified endpoint. Along with the assertion of the interrupt, the core also clears the following bits.

- DIEPCTLn.EPEnable = 1'b0
- DIEPCTLn.EPDisable = 1'b0

6. The application must read the DIEPTSIZn register for the periodic IN EP, to calculate how much data on the endpoint was transmitted on the USB.

7. The application must flush the data in the Endpoint transmit FIFO, by setting the following fields in the GRSTCTL register.

- GRSTCTL.TxFIFONum = Endpoint Transmit FIFO Number
- GRSTCTL.TxFFlush = 1

The application must poll the GRSTCTL register, until the TxFFlush bit is cleared by the core, which indicates the end of flush operation. To transmit new data on this endpoint, the application can re-enable the endpoint at a later point.

[Generic Non-periodic IN Data Transfers without Thresholding]

This section describes a regular non periodic IN data transfer when transmit thresholding is not enabled.

Application Requirements

1. Before setting up an IN transfer, the application must ensure that all data to be transmitted as part of the IN transfer is part of a single buffer, and must program the size of that buffer and its start address (in DMA mode) to the endpoint-specific registers.

2. For IN transfers, the Transfer Size field in the Endpoint Transfer Size register denotes a payload that constitutes multiple maximum-packet-size packets and a single short packet. This short packet is transmitted at the end of the transfer.

- To transmit a few maximum-packet-size packets and a short packet at the end of the transfer:
  • Transfer size[epnum] = n * mps[epnum] + sp
  (where n is an integer ≥ 0, and 0 ≤ sp < mps[epnum])
  • If (sp > 0), then packet count[epnum] = n + 1.
  Otherwise, packet count[epnum] = n
- To transmit a single zero-length data packet:
  • Transfer size[epnum] = 0
  • Packet count[epnum] = 1
- To transmit a few maximum-packet-size packets and a zero-length data packet at the end of the transfer, the application must split the transfer in two parts. The first sends maximum-packet-size data packets and the second sends the zero-length data packet alone.
  • First transfer: transfer size[epnum] = n * mps[epnum]; packet count = n;
  • Second transfer: transfer size[epnum] = 0; packet count = 1;

3. In DMA mode, the core fetches an IN data packet from the memory, always starting at a DWORD boundary. If the maximum packet size of the IN endpoint is not a multiple of 4, the application must arrange the data in the memory with pads inserted at the end of a maximum-packet-size packet so that a new packet always starts on a DWORD boundary.

4. Once an endpoint is enabled for data transfers, the core updates the Transfer Size register. At the end of IN transfer or Endpoint Disabled interrupt, the application must read the Transfer Size register to determine how much data posted in the transmit FIFO was already sent on the USB.

- Data fetched into transmit FIFO = Application-programmed initial transfer size – core-updated final transfer size
- Data transmitted on USB = (application-programmed initial packet count – Core updated final packet count) * mps[epnum]
- Data yet to be transmitted on USB = (Application-programmed initial transfer size – data transmitted on USB)

Internal Data Flow

1. The application must set the Transfer Size and Packet Count fields in the endpointspecific registers and enable the endpoint to transmit the data.

2. In Slave mode, the application must also write the required data to the transmit FIFO for the endpoint. In DMA mode, the core fetches the data from memory according to the application setting for the endpoint.

3. Every time a packet is written into the transmit FIFO, either by the core's internal DMA (in DMA mode) or the application (in Slave Mode), the transfer size for that endpoint is decremented by the packet size. The data is fetched from the memory (DMA/
Application), until the transfer size for the endpoint becomes 0. After writing the data into the FIFO, the "number of packets in FIFO" count is incremented (this is a 3-bit count, internally maintained by the core for each IN endpoint transmit FIFO. The maximum number of packets maintained by the core at any time in an IN endpoint FIFO is eight). For zero-length packets, a separate flag is set for each FIFO, without any data in the FIFO.

4. Once the data is written to the transmit FIFO, the core reads it out upon receiving an IN token. For every non-isochronous IN data packet transmitted with an ACK handshake, the packet count for the endpoint is decremented by one, until the packet count is zero. The packet count is not decremented on a TIMEOUT.

5. For zero length packets (indicated by an internal zero length flag), the core sends out a zero-length packet for the IN token and decrements the Packet Count field.

6. If there is no data in the FIFO for a received IN token and the packet count field for that endpoint is zero, the core generates a IN Tkn Rcvd When FIFO Empty Interrupt for the endpoint, provided the endpoint NAK bit is not set. The core responds with a NAK handshake for non-isochronous endpoints on the USB.

7. When the transfer size is 0 and the packet count is 0, the transfer complete interrupt for the endpoint is generated and the endpoint enable is cleared.

Application Programming Sequence

1. Program the DIEPTSIZn register with the transfer size and corresponding packet count. In DMA mode, also program the DIEPDMAn register.

2. Program the DIEPCTLn register with the endpoint characteristics and set the CNAK and Endpoint Enable bits. In DMA mode, ensure that the NextEp field is programmed so that the core fetches the data for IN endpoints in the correct order. See “Non-periodic IN Endpoint Sequencing” for details.

3. When transmitting non-zero length data packet in slave mode, the application must poll the DTXFSTSn register (where n is the FIFO number associated with that endpoint) to determine whether there is enough space in the data FIFO. The application can optionally use DIEPINTn.TxFEmp before writing the data.

[Generic non-periodic IN Data Transfers with Thresholding]

This section describes a regular non periodic IN data transfer when transmit thresholding is enabled.

Application Requirements

Application requirements are the same as those for DMA mode with no thresholding.

Internal Data Flow

1. The application should set the transfer size and packet count fields in the Endpoint pecific registers, and enable the endpoint to transmit the data.

2. The core fetches threshold amount of data for one endpoint before switching to the next in a round robin fashion with equal fairness. The priority is given to periodic endpoints. non-periodic endpoint data is fetched only if data for the periodic endpoints has been fetched or if the currently active token on the USB is a non-periodic one.The core will not switch, and continue to fetch till the complete packet, if it finds that the currently active IN token on the USB is for that particular endpoint and the FIFO does not have the omplete packet.

3. In response to an IN token on the USB, the core starts transmitting data, if the MAC finds t least threshold amount of data in the FIFO for that particular endpoint.

4. With each threshold amount of data written into the FIFO, the transfer size for that endpoint is decremented by the threshold size, except for the last packet. For the last packet, the transfer size is not decremented by the core. After writing the first threshold amount of data into the FIFO, the “number of packets in FIFO” count is incremented. For zero-length packets, a separate flag is set for that endpoint FIFO, without any data in the FIFO. This count is internally maintained by the core and is decremented when a full packet has been read out of the FIFO.

5. If the MAC sees an underrun case, where there is not enough data in the FIFO, the core will corrupt the data (invert the CRC) on the USB. The core will internally flush the FIFO, rewinding the DMA pointers and re fetch the packet(s).Also DIEPINTn.TxfifoUndrn bit will be set as an indication to the application.

6. For every non-ISO data IN packet transmitted, with an ACK handshake, the packet count for the endpoint is decremented by 1, until the packet count becomes zero. The packet count is not decremented on a TIMEOUT or underrun condition.
7. If the zero length flag in the FIFO is set (internally set by the core, when the application enables and endpoint for zero length), then core sends out zero length packet for the IN token and decrements the packet count field.
8. If there is no data (or partial threshold data) in the FIFO for a received IN token, and the packet count field for that endpoint is 0, the core generates a IN Tkn Rcvd When FIFO Empty Interrupt for the endpoint. The core responds with a NAK handshake for the non-ISO endpoints on the USB.

9. If the core does not receive a handshake after sending the packet (TIMEOUT), the core nternally flush the FIFO, rewind the DMA pointers and re-fetch the packet(s).

10. When the packet count is zero, the transfer complete interrupt for the endpoint is generated, and then the endpoint enable and transfer size are both cleared by the core.

Application Programming Sequence

1. Program the DIEPTSIZn register, for the transfer size and the corresponding packet count and the DIEPDMAn register.

2. Program the DIEPCTLn register, with the Endpoint Characteristics and set the CNAK and the Endpoint Enable bit.Also specify the Tx FIFO number in the DIEPCTLn.TXFNum field.

3. Assertion of DIEPINTn.XferCompl interrupt marks the successful completion of the nonperiodic IN transfer. Read to DIEPTSIZn register should indicate, a transfer size = 0 and packet count = 0, indicating all the data is transmitted on the USB.

[Generic Periodic IN Data Transfers without Thresholding]

This section describes a typical Periodic IN data transfer when thresholding is not enabled. Application Requirements

1. Application requirements 1, 2, 3, and 4 of “Generic Non-periodic IN Data Transfers without Thresholding” also apply to periodic IN data transfers, except for a slight modification of Requirement 2.

- The application can only transmit multiples of maximum-packet-size data packets or multiples of maximum-packet-size packets, plus a short packet at the end. To transmit a few maximum-packet-size packets and a short packet at the end of the transfer, the following conditions must be met.
  • transfer size[epnum] = n * mps[epnum] + sp
  (where n is an integer ≥ 0, and 0 ≤ sp < mps[epnum])
  • If (sp > 0), packet count[epnum] = n + 1
  Otherwise, packet count[epnum] = n;
  • mc[epnum] = packet count[epnum]
- The application cannot transmit a zero-length data packet at the end of transfer. It can transmit a single zero-length data packet by it self. To transmit a single zero-length data packet.
  • transfer size[epnum] = 0
  • packet count[epnum] = 1
  • mc[epnum] = packet count[epnum]

2. The application can only schedule data transfers 1 microframe at a time.

- (DIEPTSIZn.MC−1) * DIEPCTLn.MPS ≤ DIEPTSIZn.XferSiz ≤ DIEPTSIZn.MC * DIEPCTLn.MPS
- DIEPTSIZn.PktCnt = DIEPTSIZn.MC
- If DIEPTSIZn.XferSiz < DIEPTSIZn.MC * DIEPCTLn.MPS, the last data packet of the transfer is a short packet.

3. The application can schedule data transfers for multiple microframes, only if multiples of max packet sizes (up to 3 packets), should be transmitted every microframe. This is can be done, only when the core is operating in DMA mode. This is not a recommended mode of operation though.

- ((n*DIEPTSIZn.MC) - 1)*DIEPCTLn.MPS <= DIEPTSIZn.Transfer Size <= n*DIEPTSIZn.MC*DIEPCTLn.MPS
- DIEPTSIZn.Packet Count = n*DIEPTSIZn.MC
- n is the number of microframes for which the data transfers are scheduled

Data Transmitted per microframe in this case would be DIEPTSIZn.MC*DIEPCTLn.MPS, in all the microframes except the last one. In the microframe “n”, the data transmitted would be (DIEPTSIZn.TransferSize - (n-1)*DIEPTSIZn.MC*DIEPCTLn.MPS)

4. For Periodic IN endpoints, the data must always be prefetched 1 microframe ahead for transmission in the next microframe. This can be done, by enabling the Periodic IN endpoint 1 (micro)frame ahead of the (micro)frame in which the data transfer is scheduled.

5. The complete data to be transmitted in the (micro)frame must be written into the transmit FIFO (either by the application or the DMA), before the Periodic IN token is received. Even when 1 DWORD of the data to be transmitted per microframe is missing in the transmit FIFO when the Periodic IN token is received, the core behaves as when the FIFO was empty. When the transmit FIFO is empty,

- A zero data length packet would be transmitted on the USB for ISO IN endpoints
- A NAK handshake would be transmitted on the USB for INTR IN endpoints

6. For a High Bandwidth IN endpoint with three packets in a microframe, the application can program the endpoint FIFO size

to be 2*max_pkt_size and have the third packet load in after the first packet has been transmitted on the USB.

Internal Data Flow

1. The application must set the Transfer Size and Packet Count fields in the endpointspecific registers and enable the endpoint to transmit the data.

2. In Slave mode, the application must also write the required data to the associated transmit FIFO for the endpoint. In DMA mode, the core fetches the data for the endpoint from memory, according to the application setting.

3. Every time either the core's internal DMA (in DMA mode) or the application (in Slave mode) writes a packet to the transmit FIFO, the transfer size for that endpoint is decremented by the packet size. The data is fetched from DMA or application memory until the transfer size for the endpoint becomes 0.

4. When an IN token is received for an periodic endpoint, the core transmits the data in the FIFO, if available. If the complete data payload (complete packet, in dedicated FIFO mode) for the microframe is not present in the FIFO, then the core generates an IN Tkn Rcvd When TxF Empty Interrupt for the endpoint.

- A zero-length data packet is transmitted on the USB for isochronous IN endpoints
- A NAK handshake is transmitted on the USB for interrupt IN endpoints

5. The packet count for the endpoint is decremented by 1 under the following conditions:

- For isochronous endpoints, when a zero- or non-zero-length data packet is transmitted
- For interrupt endpoints, when an ACK handshake is transmitted

6. When the transfer size and packet count are both 0, the Transfer Completed interrupt for the endpoint is generated and the endpoint enable is cleared.

7. At the "Periodic frame Interval" (controlled by DCFG.PerFrint), when the core finds nonempty any of the isochronous IN endpoint FIFOs scheduled for the current (micro)frame non-empty, the core generates a GINTSTS.incompISOIN interrupt.

Application Programming Sequence (Transfer Per Microframe)

1. Program the DIEPTSIZn register. In DMA mode, also program the DIEPDMAn register.

2. Program the DIEPCTLn register with the endpoint characteristics and set the CNAK and Endpoint Enable bits.

3. In Slave mode, write the data to be transmitted in the next microframe to the transmit FIFO as described in "Periodic Packet Write in Slave Mode: Shared FIFO".

4. Asserting the DIEPINTn.In Token Rcvd When TxF Empty interrupt indicates that either the DMA or application has not yet written all data to be transmitted to the transmit FIFO.

- If the interrupt endpoint is already enabled when this interrupt is detected, ignore the interrupt. If it is not enabled, enable the endpoint so that the data can be transmitted on the next IN token attempt.
- If the isochronous endpoint is already enabled when this interrupt is detected, see "Incomplete Isochronous IN Data Transfers" for more details.

5. The core handles timeouts internally on interrupt IN endpoints programmed as periodic endpoints, or when Dedicated FIFO operation is used, without application intervention. The application, thus, never detects a DIEPINTn.TimeOUT interrupt for periodic interrupt IN endpoints.

6. Asserting the DIEPINTn.XferCompl interrupt with no DIEPINTn.In Tkn Rcvd When TxF Empty interrupt indicates the successful completion of an isochronous IN transfer. A read to the DIEPTSIZn register must indicate transfer size = 0 and packet count = 0, indicating all data is transmitted on the USB.

7. Asserting the DIEPINTn.XferCompl interrupt, with or without the DIEPINTn.In Tkn Rcvd When TxF Empty interrupt, indicates the successful completion of an interrupt IN transfer. A read to the DIEPTSIZn register must indicate transfer size = 0 and packet count = 0, indicating all data is transmitted on the USB.

8. Asserting the GINTSTS.incomplete Isochronous IN Transfer interrupt with none of the aforementioned interrupts indicates the core did not receive at least 1 periodic IN token in the current microframe.

- For isochronous IN endpoints, see “Incomplete Isochronous IN Data Transfers” for more details.

[Generic Periodic IN Data Transfers with Thresholding]

This section describes a typical Periodic IN data transfer when thresholding is enabled.

Application Requirements

Application requirements are the same as that for DMA mode with no thresholding.

Internal Data Flow

1. The application should set the transfer size and packet count fields in the Endpoint Specific registers, and enable the endpoint to transmit the data.

2. The core fetches threshold amount of data for one endpoint before switching to the next in a round robin fashion with equal fairness. The priority is given to periodic endpoints. The core will not switch, and continue to fetch till the complete packet, if it finds that the currently active IN token on the USB side is for that particular endpoint and the FIFO does not have the complete packet.

3. In response to an IN token on the USB, the core starts transmitting, if the MAC finds at least threshold amount of data in the FIFO for that particular endpoint.

4. After a full packet has been written into the FIFO, the transfer size for that endpoint is decremented by the packet size (or less). After writing the first threshold amount of data into the FIFO, the “number of packets in FIFO” count is incremented. For zero length packets, a separate flag will be set in the FIFO, without any data in the FIFO. This count is internally maintained b y the core and is decremented when a full packet has been read out of the FIFO.

5. If the MAC sees an underrun case, where there is not enough data in the FIFO, the core will corrupt the data (invert the CRC) on the USB. For isochronous endpoints, Underrun interrupt (DIEPINTn.TxFifoUndrn) is generated by the core for this endpoint. For interrupt endpoints, the core will flush the FIFO, rewind the DMA pointers and re-fetch the data.

6. If the core sees a timeout or an underrun condition for an interrupt endpoint, the core flushes the fifo, rewinds the DMA pointers and re-fetches the packet. No interrupt is generated to the application.

7. When an IN token is received for an periodic endpoint, the core transmits the data in the FIFO, if the core sees at least threshold amount of data. If the threshold amount of data for the packet is not present in the FIFO, the core generates a IN Tkn Rcvd When TxF Empty Interrupt for the endpoint.

- A zero data length packet would be transmitted on the USB for ISO IN endpoints
- A NAK handshake would be transmitted on the USB for INTR IN endpoints

8. The packet count for the endpoint is decremented by 1, on the following conditions

- When a zero or non zero data length for ISO endpoints
- When an ACK handshake is transmitted for INTR endpoints

9. The packet count is not decremented when the core has to corrupt a packet because of underrun condition.
10. When the transfer size is 0 and the packet count is 0, the transfer complete interrupt for the endpoint is generated and the endpoint enable is cleared.

Application Programming Sequence (Transfer Per Microframe)

1. Program the DIEPTSIZn register and in addition, in DMA mode program the DIEPDMAn register.

2. Program the DIEPCTLn register, with the Endpoint Characteristics and set the CNAK bit and the Endpoint Enable bit.Also specify the Tx FIFO number in the DIEPCTLn.TXFNum field.

3. Assertion of DIEPINTn.In Token Rcvd When TxF Empty interrupt indicates that the complete data to be transmitted is not written into the transmit FIFO.

- If the INTR endpoint is already enabled, when this interrupt is seen, ignore the interrupt. If the IN Endpoint in Dedicated Tx FIFO config is not enabled, enable the endpoint, so that the data could be transmitted on the next attempt of the IN token.

- If the ISO endpoint is already enabled, when this interrupt is seen, refer to the section Incomplete ISO IN Data Transfers, for more details.

4. TimeOUT on Interrupt IN endpoints, is handled by the core internally, without any application intervention. The application never sees any interrupt for timeout.

5. Assertion of DIEPINTn.XferCompl interrupt without any DIEPINTn.In Tkn Rcvd when TxF Empty interrupt, marks the successful completion of the ISO IN transfer. Read to DIEPTSIZn register should indicate, a transfer size = 0 and packet count = 0, indicating all the data is transmitted on the USB.

6. Assertion of DIEPINTn.XferCompl interrupt with/without any DIEPINTn.In Tkn Rcvd when TxF Empty interrupt, marks the successful completion of the INTR IN transfer. Read to DIEPTSIZn register should indicate, a transfer size = 0 and packet count = 0, indicating all the data is transmitted on the USB.

7. Assertion of GINTSTS.incomplete ISO IN Transfer interrupt with out any of the previously mentioned interrupts indicates that at least 1 periodic IN token was not received by the core, in the current microframe.

- For ISO IN endpoints, refer to the section Incomplete ISO IN Data Transfers, for more details.

8. Assertion of DIEPINTn.TxFifoUndrn interrupt indicate that there was an underrun condition for the isochronous transaction in the current microframe. The application may choose ignore this interrupt, as this will eventually result in GINTSTS.Incomplete isochronous IN interrupt at the end of periodic frame. The application can also choose to service this interrupt. If they choose to do so, then they can save some time in re-enabling the endpoint for the next microframe. In response to this interrupt,

- Disable the endpoint (Check section Disabling IN Endpoint in Dedicated Tx FIFO config).
- Application reads Endpoint DIEPSIZn register to see how much data is transferred.
- Re-enable the endpoint for the next microframe.

[Incomplete Isochronous IN Data Transfers]

This section describes what the application must do on an incomplete isochronous IN data transfer.

Internal Data Flow

1. An isochronous IN transfer is treated as incomplete in one of the following conditions.

- The core receives a corrupted isochronous IN token on at least one isochronous IN endpoint. In this case, the application detects a GINTSTS.incomplete Isochronous IN Transfer interrupt.
- The application or DMA is slow to write the complete data payload to the transmit FIFO and an IN token is received before the complete data payload is written to the FIFO. In this case, the application detects a DIEPINTn.IN Tkn Rcvd When TxFIFO Empty interrupt. The application can ignore this interrupt, as it eventually results in a GINTSTS.incomplete Isochronous IN Transfer interrupt at the end of periodic frame.
  • The core transmits a zero-length data packet on the USB in response to the received IN token.
- If thresholding is enabled and there was an underrun condition, the core also generates a DINEPINTn.TxfifoUndrn interrupt. The application can ignore this interrupt, which eventually results in a GINTSTS.incomplete ISO IN Transfer interrupt.

2. In either of the aforementioned cases, in Slave mode, the application must stop writing the data payload to the transmit FIFO as soon as possible.

3. The application must set the NAK bit and the disable bit for the endpoint. In DMA mode, the core automatically stops fetching the data payload when the endpoint disable bit is set.

4. The core disables the endpoint, clears the disable bit, and asserts the Endpoint Disable interrupt for the endpoint.

Application Programming Sequence

1. The application can ignore the DIEPINTn.IN Tkn Rcvd When TxFIFO empty interrupt on any isochronous IN endpoint, as it eventually results in a GINTSTS.incomplete Isochronous IN Transfer interrupt. The application can also ignore

DIEPINTn.TxfifoUndrn interrupt when thresholding is enabled.

2. Assertion of the GINTSTS.incomplete Isochronous IN Transfer interrupt indicates an incomplete isochronous IN transfer on at least one of the isochronous IN endpoints.

3. The application must read the Endpoint Control register for all isochronous IN endpoints to detect endpoints with incomplete IN data transfers.

4. In Slave mode, the application must stop writing data to the Periodic Transmit FIFOs associated with these endpoints on the AHB.

5. In both modes of operation, program the following fields in the DIEPCTLn register to disable the endpoint.

- DIEPCTLn.SetNAK = 1
- DIEPCTLn.Endpoint Disable = 1

6. The DIEPINTn.Endpoint Disabled interrupt's assertion indicates that the core has disabled the endpoint.

- At this point, the application must flush the data in the associated transmit FIFO or overwrite the existing data in the FIFO by enabling the endpoint for a new transfer in the next microframe. To flush the data, the application must use the GRSTCTL register.

[Stalling Non-Isochronous IN Endpoints]

This section describes how the application can stall a non-isochronous endpoint.

Application Programming Sequence

1. Disable the IN endpoint to be stalled. Set the Stall bit as well.

- DIEPCTLn.Endpoint Disable = 1, when the endpoint is already enabled
- DIEPCTLn.STALL = 1
- The Stall bit always takes precedence over the NAK bit

2. Assertion of the DIEPINTn.Endpoint Disabled interrupt indicates to the application that the core has disabled the specified endpoint.

3. The application must flush the Non-periodic or Periodic Transmit FIFO, depending on the endpoint type. In case of a non-periodic endpoint, the application must re-enable the other non-periodic endpoints, which do not need to be stalled, to transmit data.

4. Whenever the application is ready to end the STALL handshake for the endpoint, the DIEPCTLn.STALL bit must be cleared.

5. If the application sets or clears a STALL for an endpoint due to a SetFeature.Endpoint Halt command or ClearFeature.Endpoint Halt command, the Stall bit must be set or cleared before the application sets up the Status stage transfer on the control endpoint.

Special Case: Stalling the Control OUT Endpoint

The core must stall IN/OUT tokens if, during the Data stage of a control transfer, the host sends more IN/OUT tokens than are specified in the SETUP packet. In this case, the application must to enable DIEPINTn.INTknTXFEmp and DOEPINTn.OUTTknEPdis interrupts during the Data stage of the control transfer, after the core has transferred the amount of data specified in the SETUP packet. Then, when the application receives this interrupt, it must set the STALL bit in the corresponding endpoint control register, and clear this interrupt.

[Examples]

Slave Mode Bulk IN Transaction

1. The host attempts to read data (IN token) from an endpoint.

2. On receiving the IN token on the USB, the core returns a NAK handshake, because no data is available in the transmit FIFO.

3. To indicate to the application that there was no data to send, the core generates a DIEPINTn.IN Token Rcvd When TxFIFO Empty interrupt.

4. When data is ready, the application sets up the DIEPTSIZn register with the Transfer Size and Packet Count fields.
5. The application writes one maximum packet size or less of data to the Non-periodic TxFIFO.

6. The host reattempts the IN token.

7. Because data is now ready in the FIFO, the core now responds with the data and the host ACKs it.

8. Because the XferSize is now zero, the intended transfer is complete. The device core generates a DIEPINTn.XferCompl interrupt.

9. The application processes the interrupt and uses the setting of the DIEPINTn.XferCompl interrupt bit to determine that the intended transfer is complete.

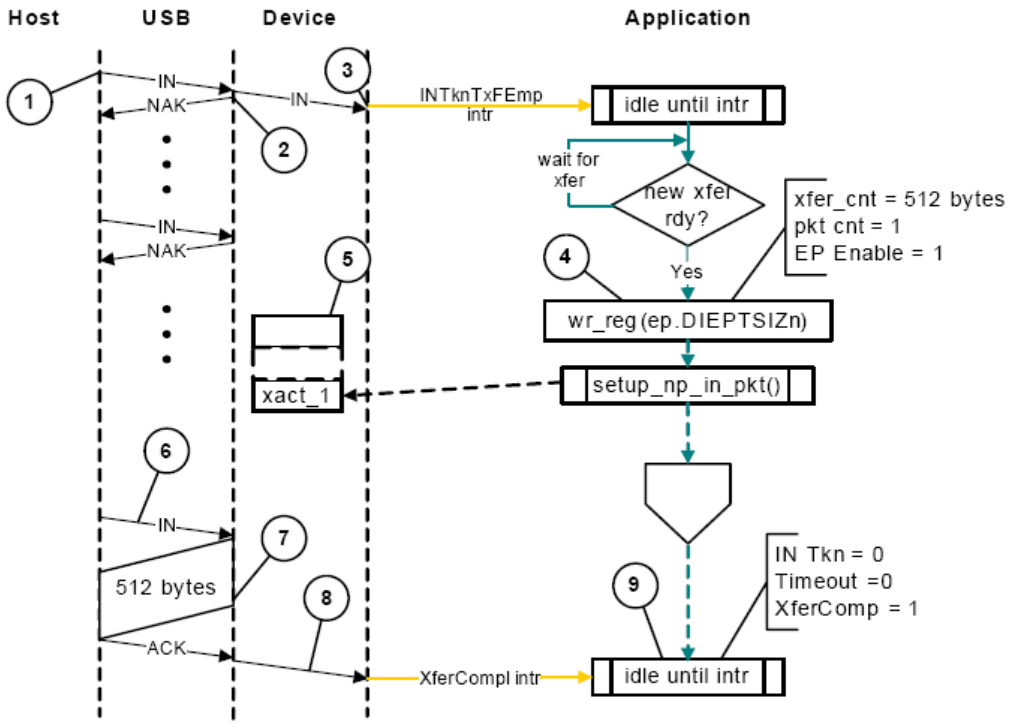


Figure 15.23 Slave Mode Bulk IN Transaction

Slave Mode Bulk IN Transfer (Pipelined Transaction)

1. The host attempts to read data (IN token) from an endpoint.

2. On receiving the IN token on the USB, the core returns a NAK handshake, because no data is available in the transmit FIFO.

3. To indicate that there was no data to send, the core generates an DIEPINTn.InTkn Rcvd When TxFIFO Empty interrupt.

4. When data is ready, the application sets up the DIEPTSIZn register with the transfer size and packet count.

5. The application writes one maximum packet size or less of data to the Non-periodic TxFIFO.

6. The host reattempts the IN token.

7. Because data is now ready in the FIFO, the core responds with the data, and the host ACKs it.

8. When the TxFIFO level falls below the halfway mark, the core generates a GINTSTS.NonPeriodic TxFIFO Empty interrupt. This triggers the application to start writing additional data packets to the FIFO.

9. A data packet for the second transaction is ready in the TxFIFO.

10. A data packet for third transaction is ready in the TxFIFO while the data for the second packet is being sent on the bus.

11. The second data packet is sent to the host.

12. The last short packet is sent to the host.

13. Because the last packet is sent and XferSize is now zero, the intended transfer is complete.

The core generates a DIEPINTn.XferCompl interrupt.

14. The application processes the interrupt and uses the setting of the DIEPINTn.XferCompl interrupt bit to determine that the intended transfer is complete.

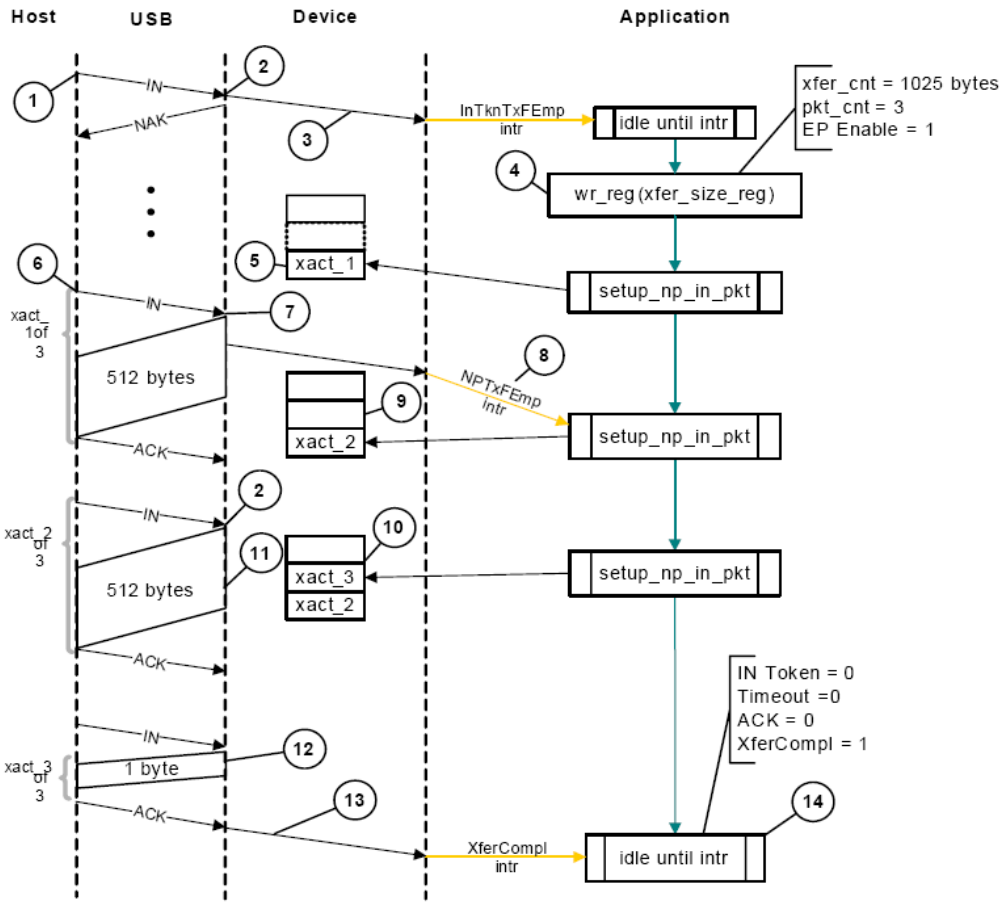


Figure 15.24 Slave Mode Bulk IN Transfer (Pipelined Transaction)

Slave Mode Bulk IN Two-Endpoint Transfer

1. The host attempts to read data (IN token) from endpoint 1.

2. On receiving the IN token on the USB, the core returns a NAK handshake, because no data is available in the transmit FIFO for endpoint 1, and generates a DIE PINT1.InTkn Rcvd When TxFIFO Empty interrupt.

3. The application processes the interrupt and initializes DIEPTSIZ1 register with the Transfer Size and Packet Count fields. The application starts writing the transaction data to the transmit FIFO.

4. The application writes one maximum packet size or less of data for endpoint 1 to the Non-periodic TxFIFO.

5. Meanwhile, the host attempts to read data (IN token) from endpoint 2.

6. On receiving the IN token on the USB, the core returns a NAK handshake, because no data is available in the transmit FIFO for endpoint 2, and the core generates a DIEPINT2.InTkn Rcvd When TxFIFO Empty interrupt.

7. Because the application has completed writing the packet for endpoint 1, it initializes the DIEPTSIZ2 register with the Transfer Size and Packet Count fields. The application starts writing the transaction data into the transmit FIFO for endpoint 2.

8. The host repeats its attempt to read data (IN token) from endpoint 1.

9. Because data is now ready in the TxFIFO, the core returns the data, which the host ACKs.

10. Meanwhile, the application has initialized the data for the next two packets in the TxFIFO (ep2.xact1 and ep1.xact2, in order).

11. The host repeats its attempt to read data (IN token) from endpoint 2.

12. Because endpoint 2's data is ready, the core responds with the data (ep2.xact_1), which the host ACKs.

13. Meanwhile, the application has initialized the data for the next two packets in the TxFIFO (ep2.xact2 and ep1.xact3, in order). The application has finished initializing data for the two endpoints involved in this scenario.

14. The host repeats its attempt to read data (IN token) from endpoint 1.

15. Because data is now ready in the FIFO, the core responds with the data, which the host ACKs.

16. The host repeats its attempt to read data (IN token) from endpoint 2.

17. With data now ready in the FIFO, the core responds with the data, which the host ACKs.

18. With the last packet for endpoint 2 sent and its XferSize now zero, the intended transfer is complete. The core generates a DIEPINT2.XferCompl interrupt for this endpoint.

19. The application processes the interrupt and uses the setting of the DIEPINT2.XferCompl interrupt bit to determine that the intended transfer on endpoint 2 is complete.

20. The host repeats its attempt to read data (IN token) from endpoint 1 (last transaction).

21. With data now ready in the FIFO, the core responds with the data, which the host ACKs.

22. Because the last endpoint one packet has been sent and XferSize is now zero, the intended transfer is complete. The core generates a DIEPINT1.XferCompl interrupt for this endpoint.

23. The application processes the interrupt and uses the setting of the DIEPINT1.XferCompl interrupt bit to determine that the intended transfer on endpoint 1 is complete.

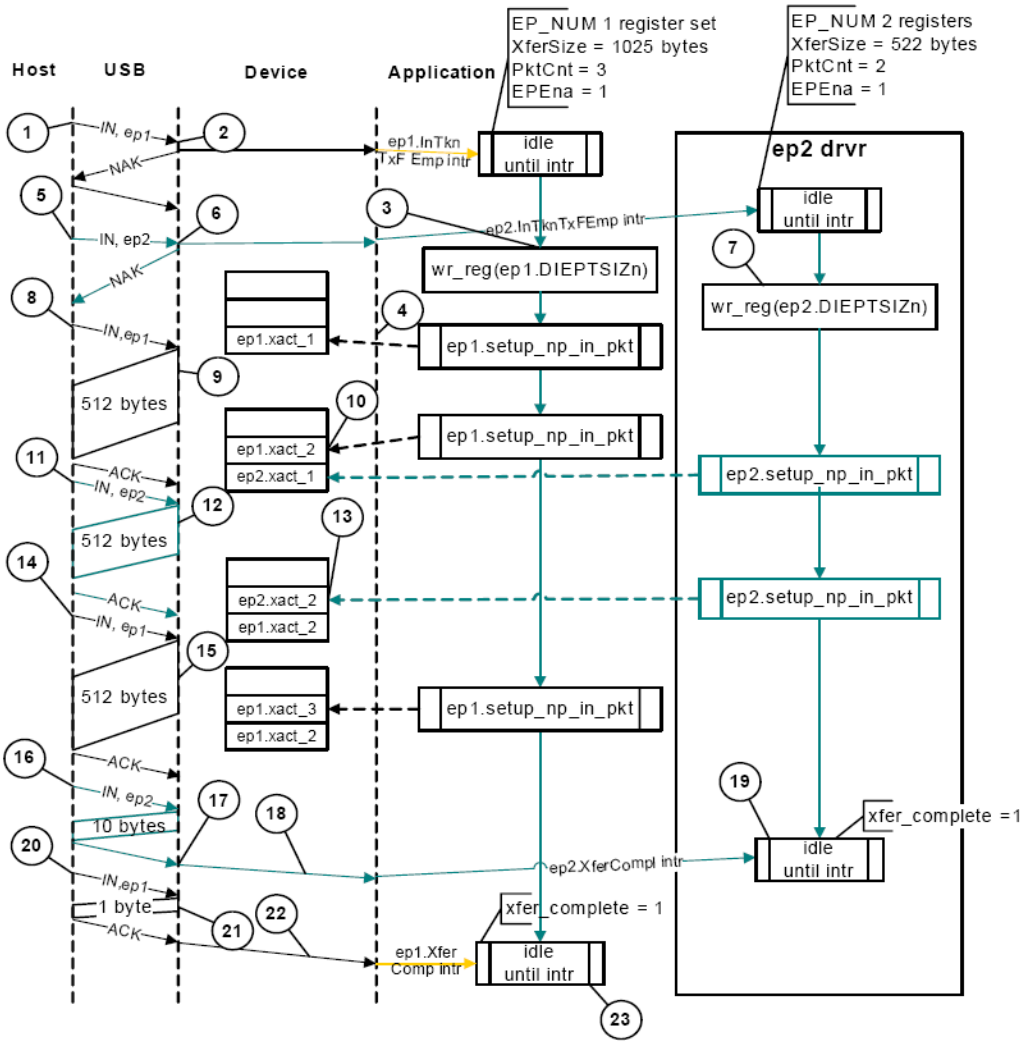


Figure 15.25 Slave Mode Bulk IN Two-Endpoint Transfer

Bulk IN Stall

1. The application has scheduled an IN transfer on receiving the DIEPINTn.InTknRcvd When TxFIFO Empty interrupt.

2. When the transfer is in progress, the application must force a STALL on the endpoint. This could be because the application has received a SetFeature.Endpoint Halt command. The application sets the Stall bit, disables the endpoint and waits for the DIEPINTn.Endpoint Disabled interrupt. This generates STALL handshakes for the endpoint on the USB.
3. On receiving the interrupt, the application flushes the Non-periodic Transmit FIFO and clears the DCTL.GlobalINNPNAK bit.

4. On receiving the ClearFeature.Endpoint Halt command, the application clears the Stall bit.

5. The endpoint behaves normally and the application can re-enable the endpoint for new transfers.

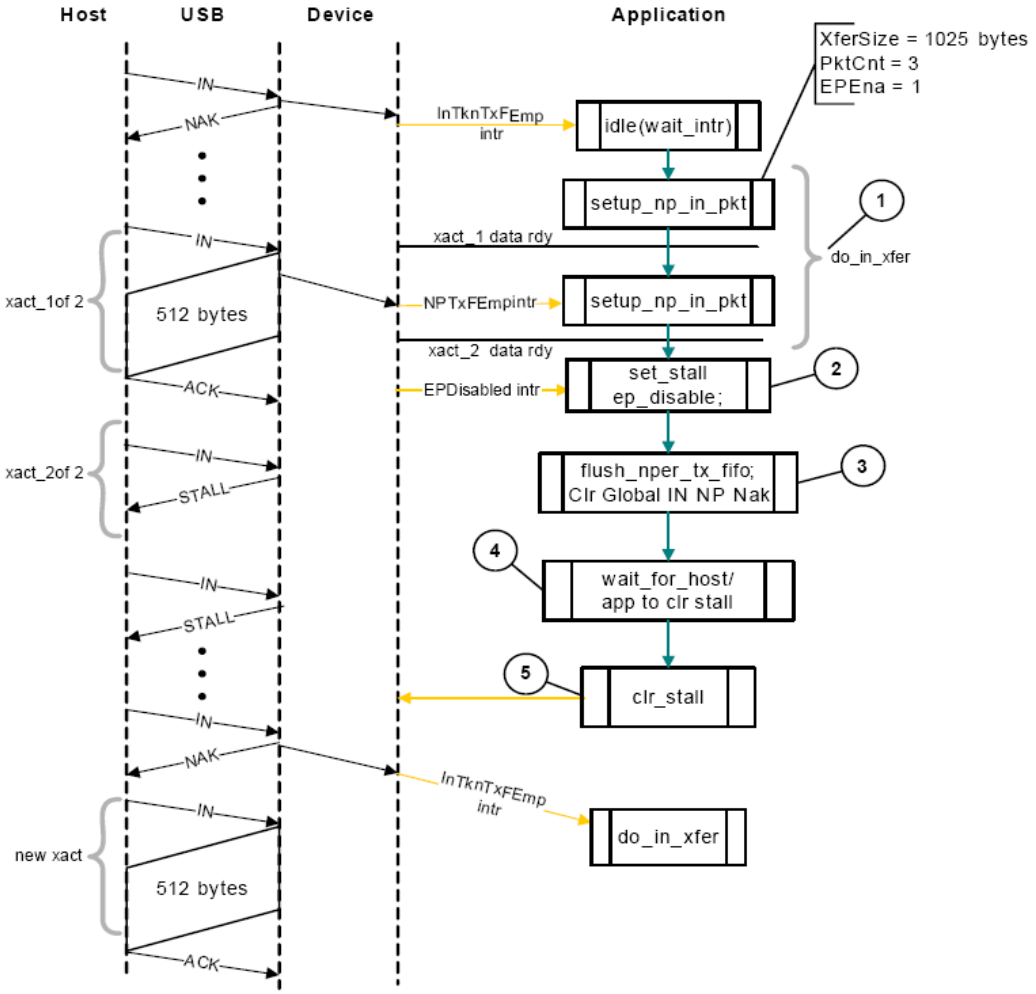


Figure 15.26 Bulk IN Stall

Bulk IN DMA mode with Thresholding

1. The host attempts to read data(IN token) from an endpoint

2. On receiving the IN token on the USB bus, the core sends back a NAK handshake because no data is available in the Transmit FIFO

3. To indicate to the application that there was no data to send, the core generates a DIEPINTn.IN Token Rcvd When TxF Empty interrupt.

4. When data is ready, the application sets up the DIEPTSIZn register with the TransferSize and Packet Count fields and enables the endpoint.

5. On seeing the endpoint enabled, the internal DMA engine start fetching the first threshold amount of data.

6. DMA engine fetching the second threshold.

7. The host re-attempts the IN token, and core start sending the data because it sees at least threshold amount of data in the FIFO.

8. The DMA engine starts fetching the third threshold after it sees threshold amount of space.

9. The MAC sees an underrun condition because of FIFO being empty in the middle of the packet, stops the packet and corrupt the CRC.

10. DMA engine starts to fetch the last threshold for that packet but it is late and will eventually result in underrun.

11. No handshake response from the host, because the packet was corrupted. Core rewinds the pointers and flush the FIFO.

12. The core starts to re-fetch the packet, staring with the first threshold.

13. The host re-attempts the IN token, and the core starts sending the data, since the core detects at least the threshold amount of data in the FIFO.

14. The core fetches the last threshold in time.

15. The core receives the handshake from host and Because the XferSize is now zero, the intended transfer is complete. Device core generates a DIEPINTn.XferCompl interrupt.

16. The application processes the interrupt and uses the setting of DIEPINTn.XferCompl interrupt bit to determine that the intended transfer is complete.

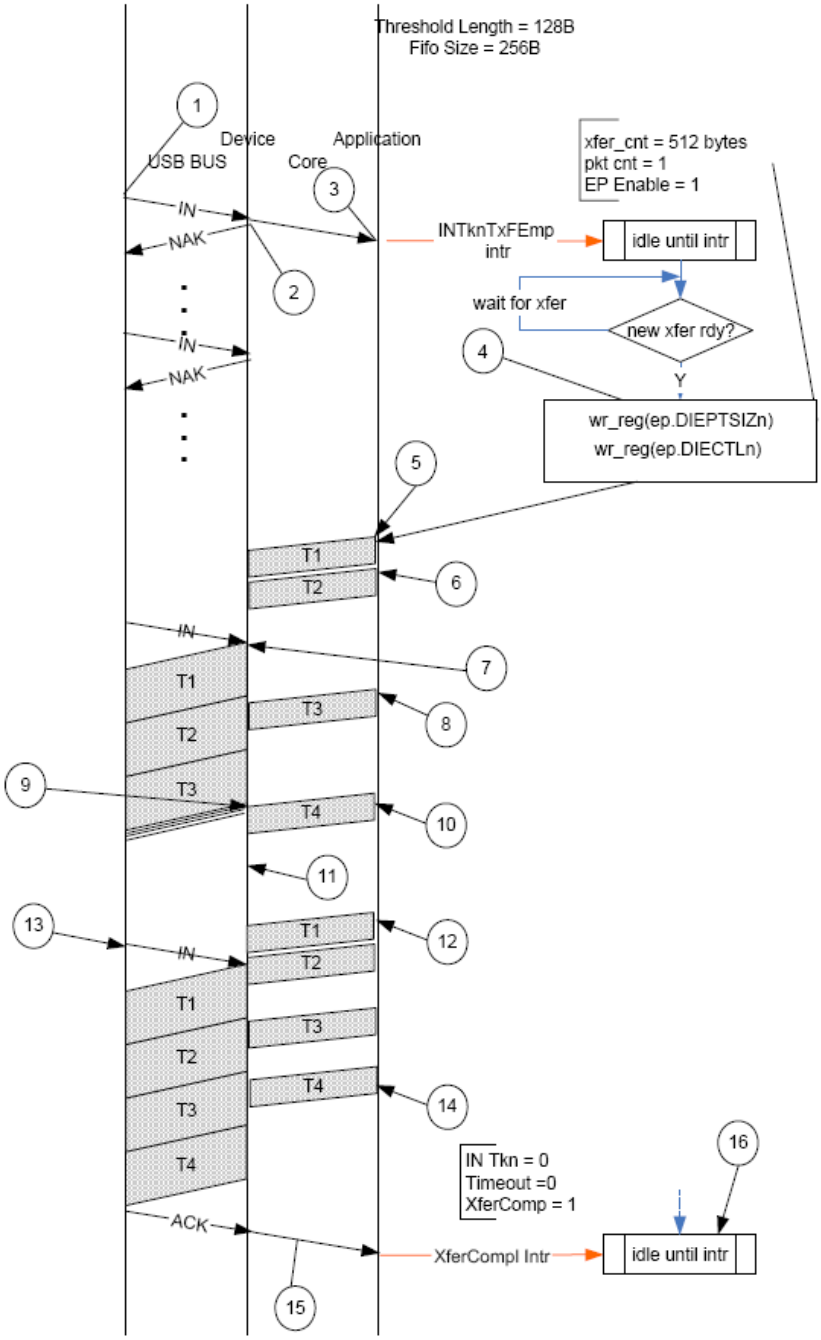


Figure 15.27 Bulk IN DMA mode with Thresholding

Isochronous IN DMA Mode with Thresholding

The FIFO size is assumed to be 512 Bytes and the threshold length is 128 Bytes.

1. Application enabled the isochronous IN endpoint for Odd microframe.

2. Core starts fetching the first 2 thresholds.

3. Core starts sending data out in response to the IN token.

4. Core sees an underrun condition, because the third threshold was not fetched in time.

5. Core generates DIEPINTn.TXFifoUnderrun interrupt (In this case, application ignores this interrupt).

6. At the End of Periodic Frame Interval, core generates GINTSTS.IncomplISO interrupt.

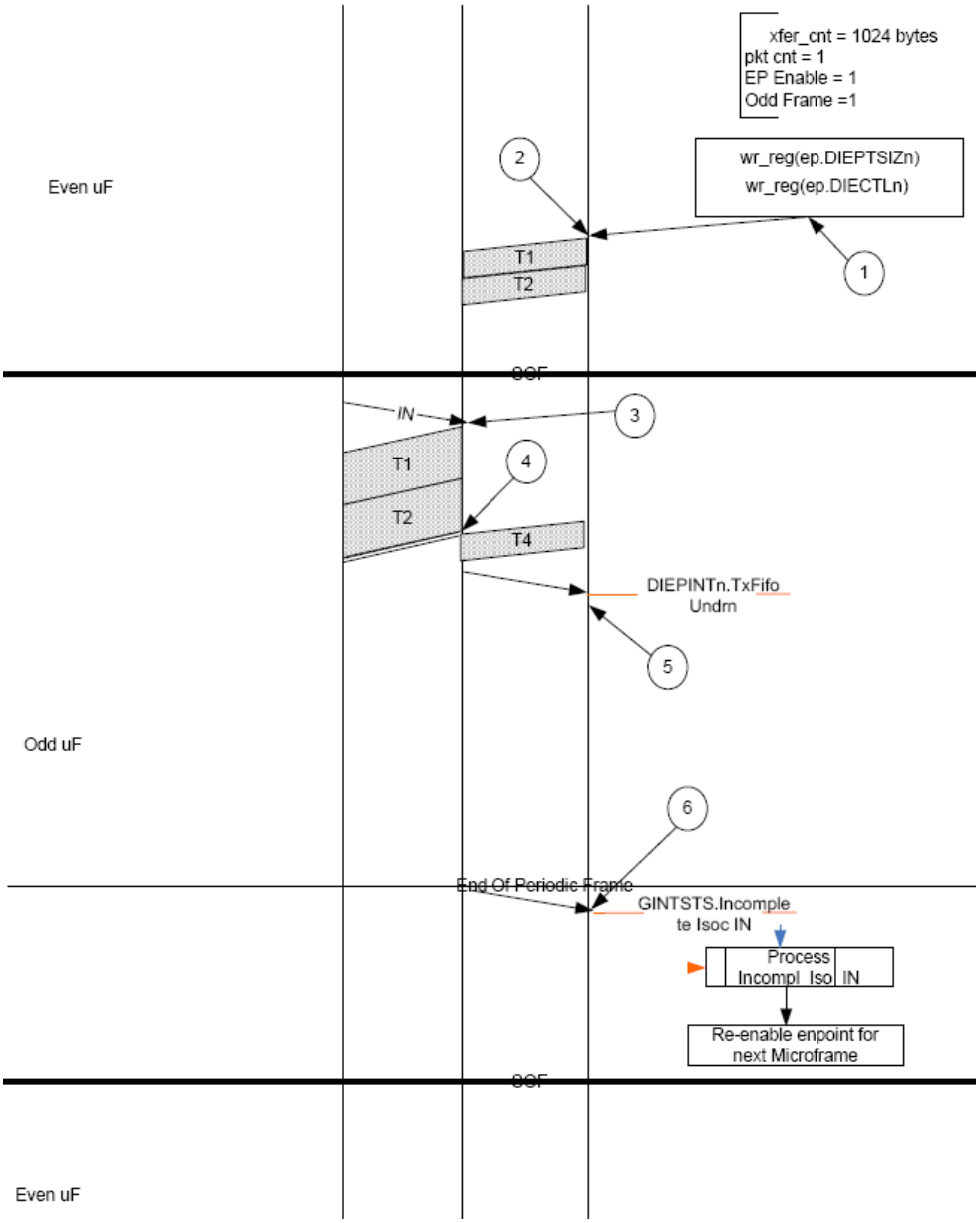


Figure 15.28 Isochronous IN DMA Mode with Thresholding

15.7.2.3 Control Transfers

This section describes the various types of control transfers.

[Control Write Transfers (SETUP, Data OUT, Status IN)]

This section describes control write transfers.

Application Programming Sequence

1. Assertion of the DOEPINTn.SETUP Packet interrupt indicates that a valid SETUP packet has been transferred to the application. See “SETUP Transactions” for more details. At the end of the Setup stage, the application must reprogram the DOEPTSIZn.SUPCnt field to 3 to receive the next SETUP packet.

2. If the last SETUP packet received before the assertion of the SETUP interrupt indicates a data OUT phase, program the core to perform a control OUT transfer as explained in “Generic Non-Isochronous OUT Data Transfers without Thresholding”. In DMA mode, the application must reprogram the DOEPDMAn register to receive a control OUT data packet to a different memory location.

3. In a single OUT data transfer on control endpoint 0, the application can receive up to 64 bytes. If the application is expecting more than 64 bytes in the Data OUT stage, the application must re-enable the endpoint to receive another 64 bytes, and must continue to do so until it has received all the data in the Data stage.

4. Assertion of the DOEPINTn.Transfer Compl interrupt on the last data OUT transfer indicates the completion of the data OUT phase of the control transfer.

5. On completion of the data OUT phase, the application must do the following.

- To transfer a new SETUP packet in DMA mode, the application must re-enable the control OUT endpoint as explained in section in section “SETUP Transactions”.
  • DOEPCTLn.EPEna = 1’b1
- To execute the received Setup command, the application must program the required registers in the core. This step is optional, based on the type of Setup command received.

6. For the status IN phase, the application must program the core as described in “Generic Non-periodic IN Data Transfers without Thresholding” to perform a data IN transfer.

7. Assertion of the DIEPINTn.Transfer Compl interrupt indicates completion of the status IN phase of the control transfer.

8. The previous step must be repeated until the DIEPINTn.Transfer Compl interrupt is detected on the endpoint, marking the completion of the control write transfer.

[Control Read Transfers (SETUP, Data IN, Status OUT)]

This section describes control write transfers.

Application Programming Sequence

1. Assertion of the DOEPINTn.SETUP Packet interrupt indicates that a valid SETUP packet has been transferred to the application. See “SETUP Transactions” for more details. At the end of the Setup stage, the application must reprogram the DOEPTSIZn.SUPCnt field to 3 to receive the next SETUP packet.

2. If the last SETUP packet received before the assertion of the SETUP interrupt indicates a data IN phase, program the core to perform a control IN transfer as explained in “Generic Non-periodic IN Data Transfers without Thresholding”.

3. On a single IN data transfer on control endpoint 0, the application can transmit up to 64 bytes. To transmit more than 64 bytes in the Data IN stage, the application must re-enable the endpoint to transmit another 64 bytes, and must continue to do so, until it has transmitted all the data in the Data stage.

4. The previous step must be repeated until the DIEPINTn.Transfer Compl interrupt is detected for every IN transfer on the endpoint.

5. The DIEPINTn.Transfer Compl interrupt on the last IN data transfer marks the completion of the control transfer’s Data stage.

6. To perform a data OUT transfer in the status OUT phase, the application must program the core as described in “SETUP and OUT Data Transfers”.

- The application must program the DCFG.NZStsOUTHShk handshake field to a proper setting before transmitting an data OUT transfer for the Status stage.
- In DMA mode, the application must reprogram the DOEPDMAn register to receive the control OUT data packet to a different memory location.

7. Assertion of the DOEPINTn.Transfer Compl interrupt indicates completion of the status OUT phase of the control transfer. This marks the successful completion of the control read transfer.

- To transfer a new SETUP packet in DMA mode, the application must re-enable the control OUT endpoint as explained in “SETUP Transactions”.
  • DOEPCTLn.EPEna = 1’b1

[Two-Stage Control Transfers (SETUP/Status IN)]

This section describes two-stage control transfers.

Application Programming Sequence

1. Assertion of the DOEPINTn.SetUp interrupt indicates that a valid SETUP packet has been transferred to the application. See “SETUP Transactions” for more detail. To receive the next SETUP packet, the application must reprogram the DOEPTSIZn.SUPCnt field to 3 at the end of the Setup stage.

2. Decode the last SETUP packet received before the assertion of the SETUP interrupt. If the packet indicates a two-stage control command, the application must do the following.

- To transfer a new SETUP packet in DMA mode, the application must re-enable the control OUT endpoint. See “SETUP Transactions” for details.
  • DOEPCTLn.EPEna = 1’b1
- Depending on the type of Setup command received, the application can be required to program registers in the core to execute the received Setup command.

3. For the status IN phase, the application must program the core described in “Generic Non-periodic IN Data Transfers without Thresholding” to perform a data IN transfer.

4. Assertion of the DIEPINTn.Transfer Compl interrupt indicates the completion of the status IN phase of the control transfer.

5. The previous step must be repeated until the DIEPINTn.Transfer Compl interrupt is detected on the endpoint, marking the completion of the two-stage control transfer.

Example: Two-Stage Control Transfer

1. SETUP packet #1 is received on the USB and is written to the receive FIFO, and the core responds with an ACK handshake. This handshake is lost and the host detects a timeout.

2. The SETUP packet in the receive FIFO results in a GINTSTS.RxFLvl interrupt to the application, causing the application to empty the receive FIFO.

3. SETUP packet #2 on the USB is written to the receive FIFO, and the core responds with an ACK handshake.
4. The SETUP packet in the receive FIFO sends the application the GINTSTS.RxFLvl interrupt and the application empties the receive FIFO.

5. After the second SETUP packet, the host sends a control IN token for the status phase. The core issues a NAK response to this token, and writes a Setup Stage Done entry to the receive FIFO. This entry results in a GINTSTS.RxFLvl interrupt to the application, which empties the receive FIFO. After reading out the Setup Stage Done DWORD, the core asserts the DOEPINTn.SetUp packet interrupt to the application.

6. On this interrupt, the application processes SETUP Packet #2, decodes it to be a two-stage control command, and clears the control IN NAK bit.

- DIEPCTLn.CNAK = 1

7. When the application clears the IN NAK bit, the core interrupts the application with a DIEPINTn.INTknTXFEmp. On this interrupt, the application enables the control IN endpoint with a DIEPTSIZn.XferSize of 0 and a DIEPTSIZn.PktCnt of 1. This results in a zero-length data packet for the status IN token on the USB.

8. At the end of the status IN phase, the core interrupts the application with a DIEPINTn.XferCompl interrupt.

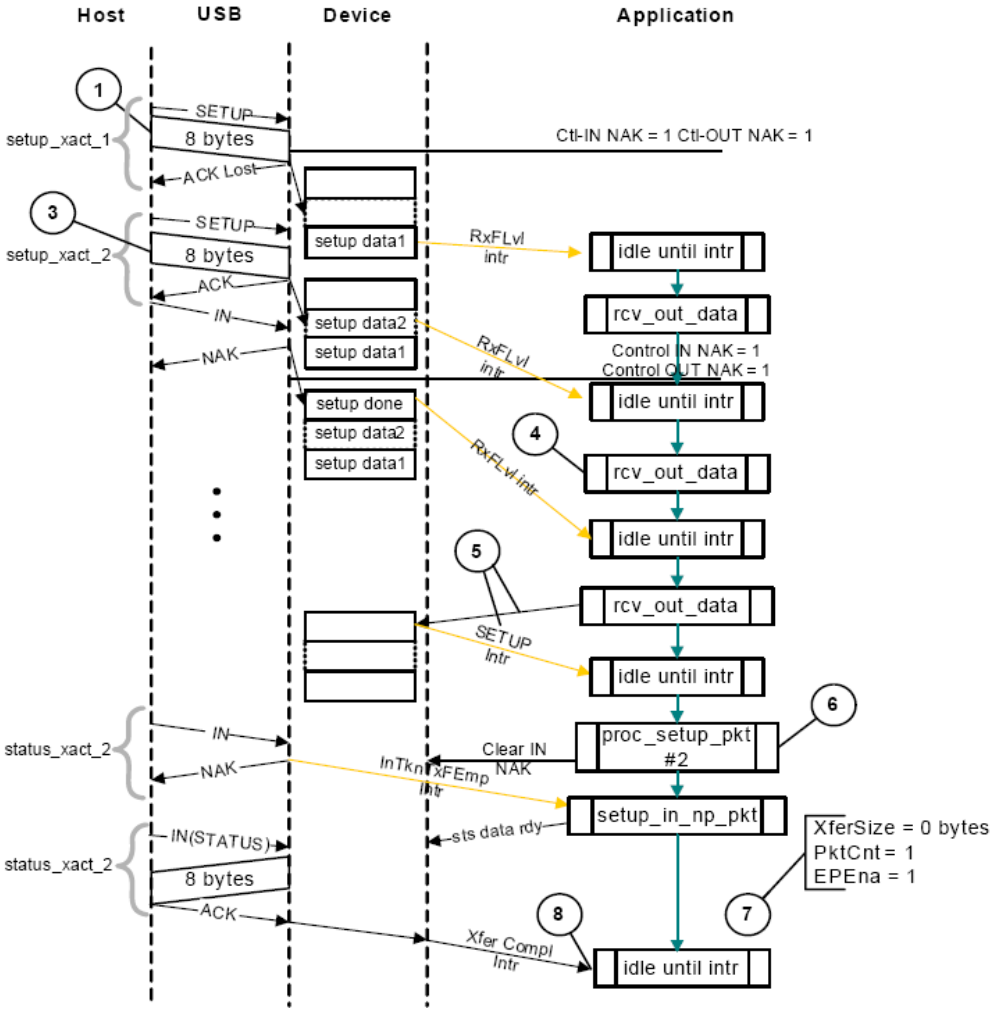


Figure 15.29 Two-Stage Control Transfer

15.7.3 Handling Babble Conditions

If the otg core receives a packet that is larger than the maximum packet size for that endpoint, the core stops writing data to the Rx buffer and waits for the end of packet (EOP). When the core detects the EOP, it flushes the packet in the Rx buffer and does not send any response to the host.

If the core continues to receive data at the EOF2 (the end of frame 2, which is very close to SOF), the core generates an early_suspend interrupt (GINTSTS.ErlySusp). On receiving this interrupt, the application must check the erratic_error status bit (DSTS.ErrticErr). If this bit is set, the application must take it as a long babble and perform a soft reset.

15.7.4 Worst Case Response Time

When the otg core acts as a device, there is a worst case response time for any tokens that follow an isochronous OUT. This worst case response time depends on the AHB clock frequency.

The core registers are in the AHB domain, and the core does not accept another token before updating these register values. The worst case is for any token following an isochronous OUT, because for an isochronous transaction, there is no handshake and the next token could come sooner. This worst case value is 7 PHY clocks when the AHB clock is the same as the PHY clock. When AHB clock is faster, this value is smaller.

If this worst case condition occurs, the core responds to bulk/interrupt tokens with a NAK and drops isochronous and SETUP tokens. The host interprets this as a timeout condition for SETUP and retries the SETUP packet. For isochronous transfers, the incomplISOCIN and incomplISOCOUT interrupts inform the application that isochronous IN/OUT packets were dropped.

15.7.5 Choosing the Value of GUSBCFG.USBTrdTim

The value in GUSBCFG.USBTrdTim is the time it takes for the MAC, in terms of PHY clocks after it has received an IN token, to get the FIFO status, and thus the first data from PFC (Packet FIFO Controller) block. This time involves the synchronization delay between the PHY and AHB clocks. The worst case delay for this is when the AHB clock is the same as the PHY clock. In this case, the delay is 5 clocks. If the PHY clock is running at 60 MHz and the AHB is running at 30 MHz, this value is 9 clocks.

Once the MAC receives an IN token, this information (token received) is synchronized to the AHB clock by the PFC (the PFC runs on the AHB clock). The PFC then reads the data from the SPRAM and writes it into the dual clock source buffer. The MAC then reads the data out of the source buffer (4 deep).

If the AHB is running at a higher frequency than the PHY, the application can use a smaller value for GUSBCFG.USBTrdTim. Next figure explains the 5-clock delay. This diagram has the following signals:

- tkn_rcvd: Token received information from MAC to PFC
- dynced_tkn_rcvd: Doubled sync tkn_rcvd, from pclk to hclk domain
- spr_read: Read to SPRAM
- spr_addr: Address to SPRAM
- spr_rdata: Read data from SPRAM
- srcbuf_push: Push to the source buffer
- srcbuf_rdata: Read data from the source buffer. Data seen by MAC

The application can use the following formula to calculate the value of GUSBCFG.USBTrdTim:

4 * AHB Clock + 1 PHY Clock = (2 clock sync + 1 clock memory address + 1 clock memory data from sync RAM) + (1 PHY Clock (next PHY clock MAC can sample the 2-clock FIFO output)

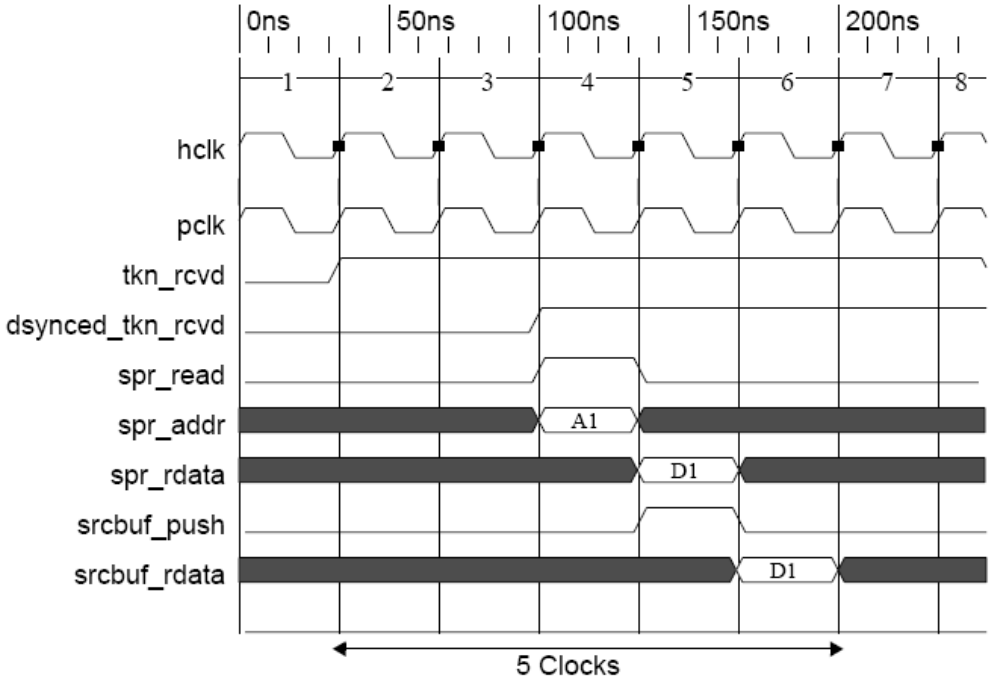


Figure 15.30 USBTrdTim Max Timing Case

15.8 OTG Programming Model

The otg core is an OTG device supporting HNP and SRP. When the core is connected to an “A” plug, it is referred to as an A-device. When the core is connected to a “B” plug it is referred to as a B-device. In Host mode, the otg core turns off VBUS to conserve power. SRP is a method by which the B-device signals the A-device to turn on VBUS power. A device must perform both data-line pulsing and VBUS pulsing, but a host can detect either data-line pulsing or VBUS pulsing for SRP. HNP is a method by which the B-device negotiates and switches to host role. In Negotiated mode after HNP, the B-device suspends the bus and reverts to the device role.

15.8.1 A-Device Session Request Protocol

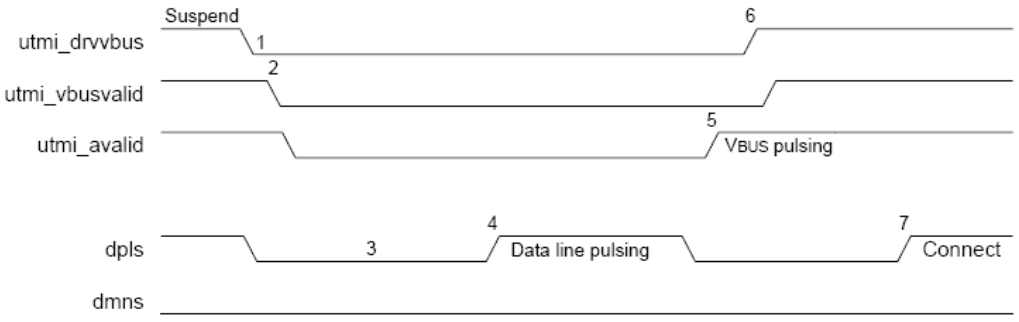


Figure 15.31 A-Device SRP

The application must set the SRP-Capable bit in the Core USB Configuration register. This enables the otg core to detect SRP as an A-device.

1. To save power, the application suspends and turns off port power when the bus is idle by writing the Port Suspend and Port Power bits in the Host Port Control and Status register.

2. PHY indicates port power off by deasserting the utmi_vbusvalid signal.

3. The device must detect SE0 for at least 2 ms to start SRP when VBUS power is off.

4. To initiate SRP, the device turns on its data line pull-up resistor for 5 to 10 ms. The otg core detects data-line pulsing.

5. The device drives VBUS above the A-device session valid (2.0 V minimum) for VBUS pulsing. The otg core interrupts the application on detecting SRP. The Session Request Detected bit is set in Global Interrupt Status register (GINTSTS.SessReqInt).

6. The application must service the Session Request Detected interrupt and turn on the Port Power bit by writing the Port Power bit in the Host Port Control and Status register. The PHY indicates port power-on by asserting utmi_vbusvalid signal.
7. When the USB is powered, the device connects, completing the SRP process.

15.8.2 B-Device Session Request Protocol

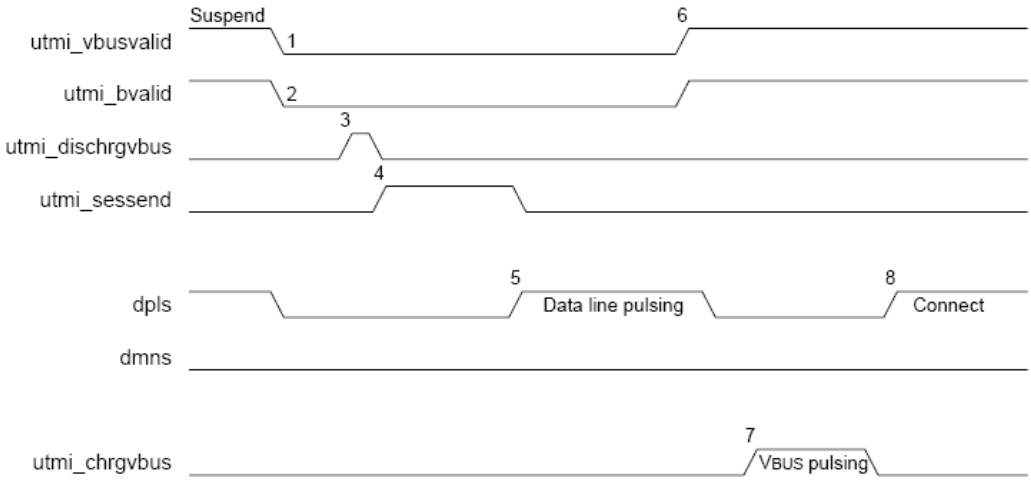


Figure 15.32 B-Device SRP

The application must set the SRP-Capable bit in the Core USB Configuration register. This enables the otg core to initiate SRP as a B-device. SRP is a means by which the otg core can request a new session from the host.

1. To save power, the host suspends and turns off port power when the bus is idle. PHY indicates port power off by deasserting the utmi_vbusvalid signal. The otg core sets the Early Suspend bit in the Core Interrupt register after 3 ms of bus idleness. Following this, the otg core sets the USB Suspend bit in the Core Interrupt register.

2. The PHY indicates the end of the B-device session by deasserting the utmi_bvalid signal.

3. The otg core asserts the utmi_dischrgvbus signal to indicate to the PHY to speed up VBUS discharge.

4. The PHY indicates the session's end by asserting the utmi_sessend signal. This is the initial condition for SRP. The otg core requires 2 ms of SE0 before initiating SRP. For a USB 1.1 full-speed serial transceiver, the application must wait until VBUS discharges to 0.2 V after GOTGCTL.BSesVld is deasserted. This discharge time can be obtained from the transceiver vendor and varies between different transceivers.

5. The application initiates SRP by writing the Session Request bit in the OTG Control and Status register. The otg core perform data-line pulsing followed by VBUS pulsing.

6. The host detects SRP from either the data-line or VBUS pulsing, and turns on VBUS. The PHY indicates VBUS power-on by asserting utmi_vbusvalid.

7. The otg core performs VBUS pulsing by asserting utmi_chrgvbus. The host starts a new session by turning on VBUS, indicating SRP success. The otg core interrupts the application by setting the Session Request Success Status Change bit in the OTG Interrupt Status register. The application reads the Session Request Success bit in the OTG Control and Status register.

8. When the USB is powered, the otg core connects, completing the SRP process.

15.8.3 A-Device Host Negotiation Protocol

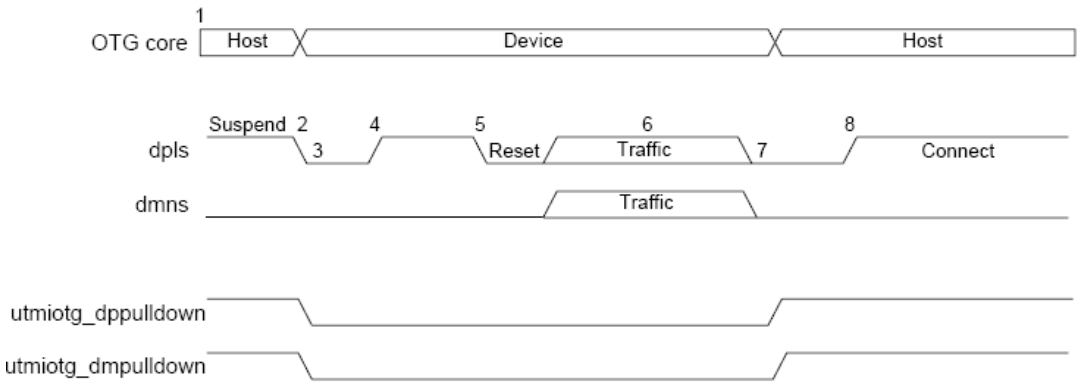


Figure 15.33 A-Device HNP

HNP switches the USB host role from the A-device to the B-device. The application must set the HNP-Capable bit in the Core USB Configuration register to enable the otg core to perform HNP as an A-device.

1. The otg core sends the B-device a SetFeature b_hnp_enable descriptor to enable HNP support. The B-device's ACK response indicates that the B-device supports HNP. The application must set Host Set HNP Enable bit in the OTG Control and Status register to indicate to the otg core that the B-device supports HNP.

2. When it has finished using the bus, the application suspends by writing the Port Suspend bit in the Host Port Control and Status register.

3. When the B-device observes a USB suspend, it disconnects, indicating the initial condition for HNP. The B-device initiates HNP only when it must switch to the host role; otherwise, the bus continues to be suspended. The otg core sets the Host Negotiation Detected interrupt in the OTG Interrupt Status register, indicating the start of HNP.

4. The otg core deasserts the utmiotg_dppulldown and utmiotg_dmpulldown signals to indicate a device role. The PHY enable the D + pull-up resistor indicates a connect for B-device. The application must read the Current Mode bit in the OTG Control and Status register to determine Device mode operation.

5. The B-device detects the connection, issues a USB reset, and enumerates the otg core for data traffic.

6. The B-device continues the host role, initiating traffic, and suspends the bus when done. The otg core sets the Early Suspend bit in the Core Interrupt register after 3 ms of bus idleness. Following this, the otg core sets the USB Suspend bit in the Core Interrupt register.

7. In Negotiated mode, the otg core detects the suspend, disconnects, and switches back to the host role. The otg core asserts the utmiotg_dppulldown and utmiotg_dmpulldown signals to indicate its assumption of the host role. The otg core sets the Connector ID Status Change interrupt in the OTG Interrupt Status register. The application must read the connector ID status in the OTG Control and Status register to determine the otg core's operation as an A-device. This indicates the completion of HNP to the application. The application must read the Current Mode bit in the OTG Control and Status register to determine Host mode operation.

8. The B-device connects, completing the HNP process.

15.8.4 B-Device Host Negotiation Protocol

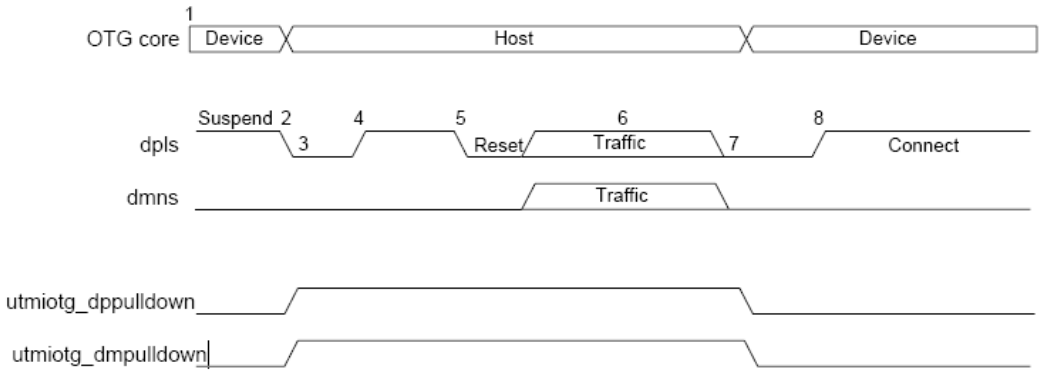


Figure 15.34 B-Device HNP

HNP switches the USB host role from B-device to A-device. The application must set the HNP-Capable bit in the Core USB Configuration register to enable the otg core to perform HNP as a B-device.

1. The A-device sends the SetFeature b_hnp_enable descriptor to enable HNP support. The otg core's ACK response indicates that it supports HNP. The application must set the Device HNP Enable bit in the OTG Control and Status register to indicate HNP support. The application sets the HNP Request bit in the OTG Control and Status register to indicate to the otg core to initiate HNP.

2. When it has finished using the bus, the A-device suspends by writing the Port Suspend bit in the Host Port Control and Status register. The otg core sets the Early Suspend bit in the Core Interrupt register after 3 ms of bus idleness. Following this, the otg core sets the USB Suspend bit in the Core Interrupt register.

3. The otg core disconnects and the A-device detects SE0 on the bus, indicating HNP. The otg core asserts the utmiotg_dppulldown and utmiotg_dmpulldown signals to indicate its assumption of the host role.

4. The A-device responds by activating its D + pull-up resistor within 3 ms of detecting SE0. The otg core detects this as a connect. The otg core sets the Host Negotiation Success Status Change interrupt in the OTG Interrupt Status register, indicating the HNP status. The application must read the Host Negotiation Success bit in the OTG Control and Status register to determine host negotiation success. The application must read the Current Mode bit in the Core Interrupt register (GINTSTS) to determine Host mode operation.

5. The otg core issues a USB reset and enumerates the A-device for data traffic.

6. The otg core continues the host role of initiating traffic, and when done, suspends the bus by writing the Port Suspend bit in the Host Port Control and Status register.

7. In Negotiated mode, when the A-device detects a suspend, it disconnects and switches back to the host role. The otg core deasserts the utmiotg_dppulldown and utmiotg_dmpulldown signals to indicate the assumption of the device role. The application must read the Current Mode bit in the Core Interrupt (GINTSTS) register to determine the Host mode operation.

8. The otg core connects, completing the HNP process.

15.8.5 Clock Gating

You can use clock gating to reduce power consumption when the USB is suspended or the session is not valid. The PHY turns off the PHY clock for as long as the core asserts the suspend_n signal to the PHY. The AHB clock to the HSOTG internal modules can also be gated by writing to the Gate hclk bit in the Power and Clock Gating Control register. The following sections show the procedures you must follow to use the clock gating feature.

15.8.5.1 Host Mode Suspend and Resume With Clock Gating

1. The application sets the Port Suspend bit in the Host Port CSR, and the core drives a USB suspend.

2. The application sets the Stop PHY Clock bit in the Power and Clock Gating Control register, the core asserts the suspend_n signal to the PHY, and the PHY clock stops. The application sets the Gate hclk bit in the Power and Clock Gating Control register, the core gates the hclk (hclk_gated) to AHB-domain modules other than the BIU.

3. The core remains in Suspend mode.

4. The application clears the Gate hclk and Stop PHY Clock bits, and the PHY clock is generated.

5. The application sets the Port Resume bit, and the core starts driving Resume signaling.

6. The application clears the Port Resume bit after at least 20 ms.

7. The core is in normal operating mode.

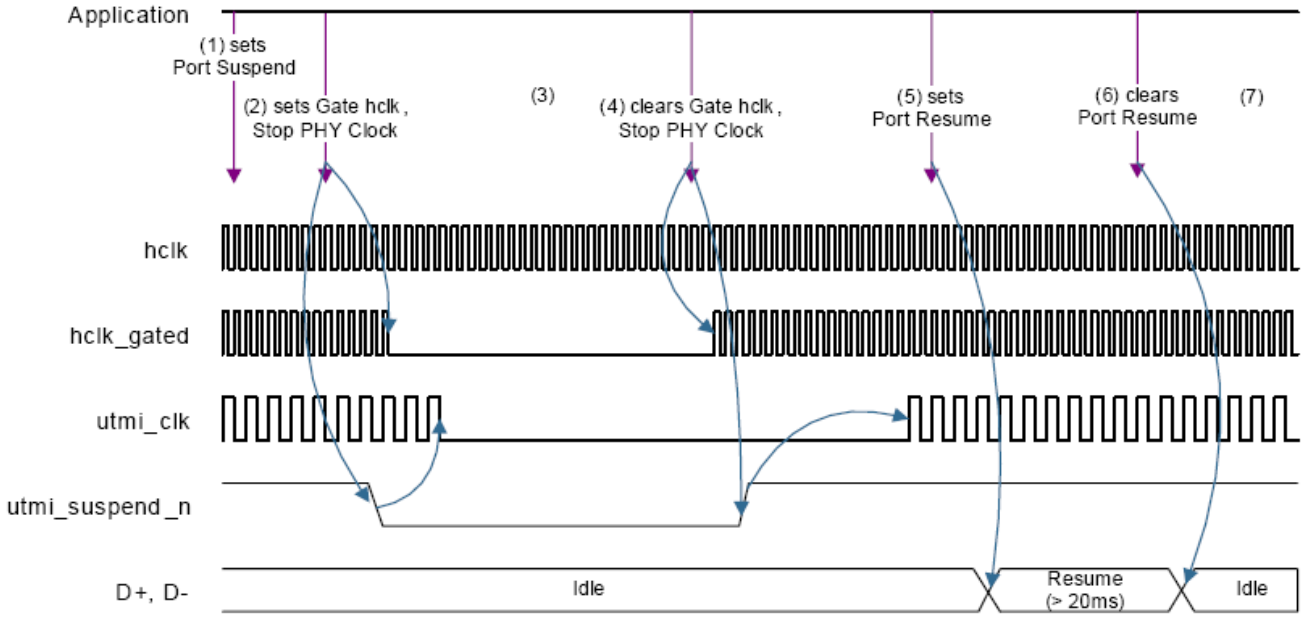


Figure 15.35Host Mode Suspend and Resume With Clock Gating

15.8.5.2 Host Mode Suspend and Remote Wakeup With Clock Gating

Sequence of operations

1. The application sets the Port Suspend bit in the Host Port CSR, and the core drives a USB suspend.

2. The application sets the Stop PHY Clock bit in the Power and Clock Gating Control register, the core asserts the suspend_n signal to the PHY, and the PHY clock stops. The application sets the Gate hclk bit in the Power and Clock Gating Control register, and the core gates the hclk (hclk_ctl) to AHB-domain modules other than the BIU.

3. The core remains in Suspend mode.

4. The Remote Wakeup signaling from the device is detected. The core deasserts the suspend_n signal to the PHY to generate the PHY clock. The core generates a Remote Wakeup Detected interrupt.

5. The application clears the Gate hclk and Stop PHY Clock bits. The core sets the Port Resume bit.

6. The application clears the Port Resume bit after at least 20 ms.

7. The core is in normal operating mode.

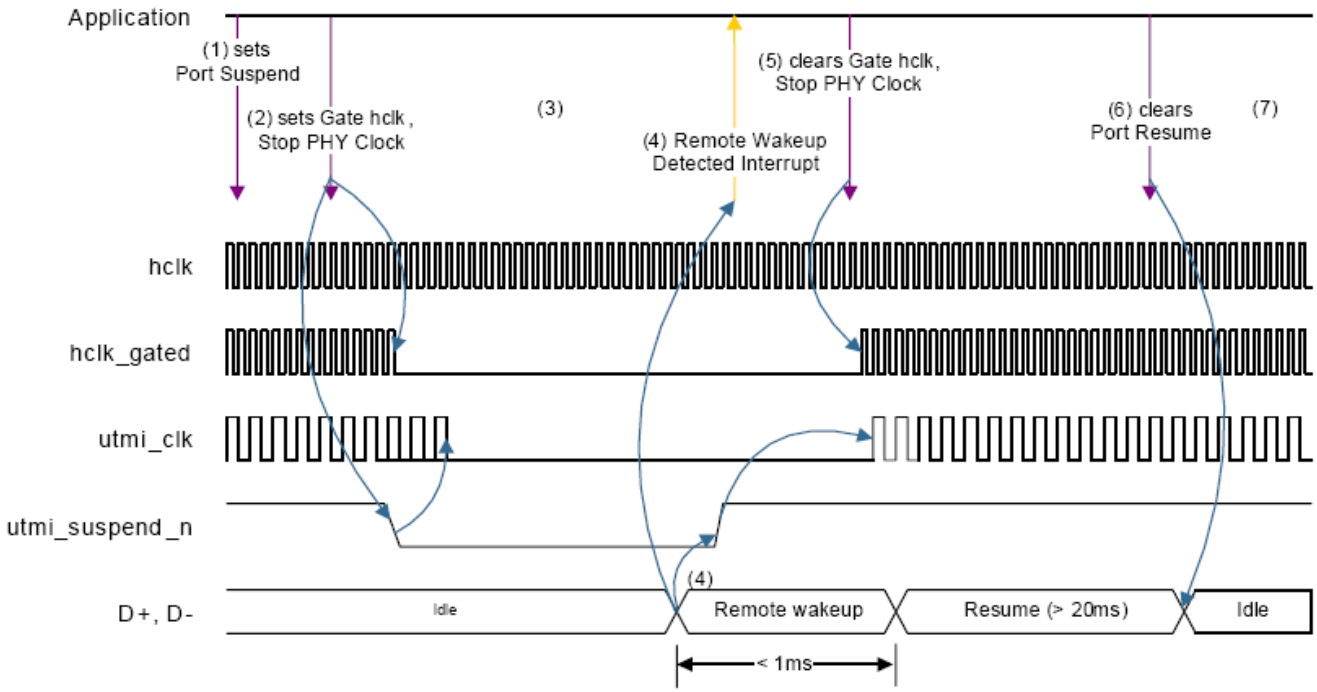


Figure 15.36Host Mode Suspend and Remote Wakeup With Clock Gating

15.8.5.3 Host Mode Session End and Start With Clock Gating

Sequence of operations

1. The application sets the Port Suspend bit in the Host Port CSR, and the core drives a USB suspend.

2. The application clears the Port Power bit. The core turns off VBUS.

3. The application sets the Stop PHY Clock bit in the Power and Clock Gating Control register, the core asserts the suspend_n signal to the PHY, and the PHY clock stops. The application sets the Gate hclk bit in the Power and Clock Gating Control register, and the core gates the hclk (hclk_ctl) to AHB-domain modules other than the BIU.

4. The core remains in Low-Power mode.

5. The application clears the Gate hclk bit and the application clears the Stop PHY Clock bit to start the PHY clock.

6. The application sets the Port Power bit to turn on VBUS.

7. The core detects device connection and drives a USB reset.

8. The core is in normal operating mode.

15.8.5.4 Host Mode Session End and SRP With Clock Gating

Sequence of operations

1. The application sets the Port Suspend bit in the Host Port CSR, and the core drives a USB suspend.

2. The application clears the Port Power bit. The core turns off VBUS.

3. The application sets the Stop PHY Clock bit in the Power and Clock Gating Control register, the core asserts the suspend_n signal to the PHY, and the PHY clock stops. The application sets the Gate hclk bit in the Power and Clock Gating Control register, and the core gates the hclk (hclk_ctl) to AHB-domain modules other than the BIU.

4. The core remains in Low-Power mode.

5. SRP (data line pulsing) from the device is detected. The core deasserts the suspend_n signal to the PHY to generate the PHY clock. An SRP Request Detected interrupt is generated.

6. The application clears the Gate hclk bit and the Stop PHY Clock bit.

7. The core sets the Port Power bit to turn on VBUS.

8. The core detects device connection and drives a USB reset.

9. The core is in normal operating mode.

15.8.5.5 Device Mode Suspend and Resume With Clock Gating

Sequence of operations

1. The core detects a USB suspend and generates a Suspend Detected interrupt.

2. The application sets the Stop PHY Clock bit in the Power and Clock Gating Control register, the core asserts the suspend_n signal to the PHY, and the PHY clock stops. The application sets the Gate hclk bit in the Power and Clock Gating Control register, and the core gates the hclk (hclk_ctl) to AHB-domain modules other than the BIU.

3. The core remains in Suspend mode.

4. The Resume signaling from the host is detected. The core deasserts the suspend_n signal to the PHY to generate the PHY clock. A Resume Detected interrupt is generated.

5. The application clears the Gate hclk bit and the Stop PHY Clock bit.

6. The host finishes Resume signaling.

7. The core is in normal operating mode.

15.8.5.6 Device Mode Suspend and Remote Wakeup With Clock Gating

Sequence of operations

1. The core detects a USB suspend and generates a Suspend Detected interrupt.

2. The application sets the Stop PHY Clock bit in the Power and Clock Gating Control register, the core asserts the suspend_n signal to the PHY, and the PHY clock stops. The application sets the Gate hclk bit in the Power and Clock Gating Control register, the core gates the hclk (hclk_ctl) to AHB-domain modules other than the BIU.

3. The core remains in Suspend mode.

4. The application clears the Gate hclk bit and the Stop PHY Clock bit.

5. The application sets the Remote Wakeup bit in the Device Control register, the core starts driving Remote Wakeup signaling.

6. The host drives Resume signaling.

7. The core is in normal operating mode.

15.8.5.7 Device Mode Session End and Start With Clock Gating

Sequence of operations

1. The core detects a USB suspend, and generates a Suspend Detected interrupt. The host turns off VBUS.

2. The application sets the Stop PHY Clock bit in the Power and Clock Gating Control register, the core asserts the suspend_n signal to the PHY, and the PHY clock stops. The application sets the Gate hclk bit in the Power and Clock Gating Control register, and the core gates the hclk (hclk_ctl) to AHB-domain modules other than the BIU.

3. The core remains in Low-Power mode.

4. The new session is detected (bsessvld is high). The core deasserts the suspend_n signal to the PHY to generate the PHY clock. A New Session Detected interrupt is generated.

5. The application clears the Gate hclk and Stop PHY Clock bits.

6. The core detects USB reset.

7. The core is in normal operating mode

15.8.5.8 Device Mode Session End and SRP With Clock Gating

Sequence of operations

1. The core detects a USB suspend, and generates a Suspend Detected interrupt. The host turns off VBUS.

2. The application sets the Stop PHY Clock bit in the Power and Clock Gating Control register, the core asserts the suspend_n signal to the PHY, and the PHY clock stops. The application sets the Gate hclk bit in the Power and Clock Gating Control register, and the core gates the hclk (hclk_ctl) to AHB-domain modules other than the BIU.

3. The core remains in Low-Power mode.

4. The application clears the Gate hclk and Stop PHY Clock bits.

5. The application sets the SRP Request bit, and the core drives data line and VBUS pulsing.

6. The host turns on VBUS, detects device connection, and drives a USB reset.

7. The core is in normal operating mode.

15.9 Miscellaneous Topics

15.9.1 Data FIFO RAM Allocation

If Dynamic FIFO Sizing is enabled in the core, the External RAM must be allocated among different FIFOs in the core before any transactions can start. The application must follow this procedure every time it changes core FIFO RAM allocation.

The application must allocate data RAM per FIFO based on the AHB's operating frequency, the PHY Clock frequency, the available AHB bandwidth, and the performance required on the USB. Based on the above mentioned criteria, the application must provide a table as described below with RAM sizes for each FIFO in each mode.

15.9.1.1 Device Mode

[Dedicated Tx FIFO Operation]

When allocating data RAM for FIFOs in Device mode when dedicated TX FIFO is used, keep in mind these factors:

1. Receive FIFO RAM Allocation:

- RAM for SETUP Packets: 4 * n + 6 locations must be Reserved in the receive FIFO to receive up to n SETUP packets on control endpoints, where n is the number of control endpoints the device core supports. The core does not use these locations, which are Reserved for SETUP packets, to write any other data.
- 1 location for Global OUT NAK
- Status information is written to the FIFO along with each received packet. Therefore, a minimum space of (Largest Packet Size / 4) + 1 must be allotted to receive packets. If a high-bandwidth endpoint is enabled, or multiple isochronous endpoints are enabled, then at least two (Largest Packet Size / 4) + 1 spaces must be allotted to receive back-to-back packets. Typically, two (Largest Packet Size / 4) + 1 spaces are recommended so that when the previous packet is being transferred to AHB, the USB can receive the subsequent packet. If AHB latency is high, you must allocate enough space to receive multiple packets. This is critical to prevent dropping any isochronous packets.
- Along with each endpoint's last packet, transfer complete status information is also pushed to the FIFO. Typically, one location for each OUT endpoint is recommended.

2. Transmit FIFO RAM Allocation:

- The minimum RAM space required for each IN Endpoint Transmit FIFO is the maximum

packet size for that particular IN endpoint.

- The more space allocated in the transmit IN Endpoint FIFO results in a better performance on the USB and can hide the latencies on the AHB.

FIFO Name	Data RAM Size
Receive dta FIFO	rx_fifo_size. This must include RAM for setup packets, OUT endpoint control information and data OUT packets, as mentioned earlier.
Transmit FIFO 0	Tx_fifo_size[0]
Transmit FIFO 1	Tx_fifo_size[1]
….	…
Transmit FIFO i	Tx_fifo_size[i]

With this information at hand, the following registers must be programmed as follows:

1. Receive FIFO Size Register (GRXFSIZ)

- GRXFSIZ.Receive FIFO Depth = rx_fifo_size;

2. Device IN Endpoint Transmit FIFO0 Size Register (GNPTXFSIZ)

- GNPTXFSIZ.non-periodic Transmit FIFO Depth = tx_fifo_size[0];
- GNPTXFSIZ.non-periodic Transmit RAM Start Address = rx_fifo_size;

3. Device IN Endpoint Transmit FIFO#1 Size Register (DIEPTXF1)

- DIEPTXF1. Transmit RAM Start Address = GNPTXFSIZ.FIFO0 Transmit RAM Start Address + tx_fifo_size[0];

4. Device IN Endpoint Transmit FIFO#2 Size Register (DIEPTXF2)

- DIEPTXF2.Transmit RAM Start Address = DIEPTXF1.Transmit RAM Start Address + tx_fifo_size[1];

5. Device IN Endpoint Transmit FIFO#i Size Register (DIEPTXFi)

- DIEPTXFm.Transmit RAM Start Address = DIEPTXFi-1.Transmit RAM Start Address + tx_fifo_size[i-1];

6. The transmit FIFOs and receive FIFO must be flushed after the RAM allocation is done, for the proper functioning of the FIFOs.

- GRSTCTL.TxFNum = 5'h10
- GRSTCTL.TxFFlush = 1'b1
- GRSTCTL.RxFFlush = 1'b1
- The application has to wait until the TxFFlush bit and the RxFFlush bits are cleared, before performing any operation on the core.

[Dedicated Tx FIFO operation with Thresholding]

1. Receive FIFO RAM allocation

- RAM for Setup Packets: 7*n + 6 locations have to be Reserved in the receive FIFO, to receive up to "n" setup packets on control endpoints, where "n" is the number of control endpoints supported by the device core. These locations Reserved for Setup Packets are not used by the core, to write any other data.
- 1 location for Global OUT NAK
- It is recommended to have an Rx FIFO space for two thresholds. Along with each received threshold, a status information is also written in to the FIFO. With the last threshold of a packet, two status DWORDs are written into the FIFO. With the last packet of a transfer, transfer complete status information is written into the FIFO. So worst case ,

a minimum of 2*(Rx_threshold length/4 + 4) space is needed to be allocated to receive a packet.

2. Transmit FIFO RAM Allocation:

- The minimum RAM space required for each IN Endpoint Transmit FIFO is min(2*Tx_threshold_length, endpoint_max_pkt_size).
- The more space allocated in the transmit IN Endpoint FIFO results in a better performance on the USB and can hide the latencies on the AHB.

If high AHB latencies results in underrun error very often, then there is a possibility that the Host might disable this endpoint because of Errors in the packet (typically when there is error in 3 consecutive packets). So thresholding must be enabled only when the AHB latency is not very high.

15.9.1.2 Host Mode

With this information at hand, the following registers must be programmed as follows:

1. Receive FIFO Size Register (GRXFSIZ)

- GRXFSIZ.RxFDep = rx_fifo_size;

2. Non-periodic Transmit FIFO Size Register (GNPTXFSIZ)

- GNPTXFSIZ.NPTxFDe = tx_fifo_size[0];
- GNPTXFSIZ.NPTxFStAddr = rx_fifo_size;

3. Host Periodic Transmit FIFO Size Register (HPTXFSIZ)

- HPTXFSIZ.PTxFSize = tx_fifo_size[1];
- HPTXFSIZ.PTxFStAddr = GNPTXFSIZ.NPTxFStAddr + tx_fifo_size[0];

4. The transmit FIFOs and receive FIFO must be flushed after RAM allocation for proper FIFO function.

- GRSTCTL.TxFNum = 5'h10
- GRSTCTL.TxFFlush = 1'b1
- GRSTCTL.RxFFlush = 1'b1
- The application must wait until the TxFFlush bit and the RxFFlush bits are cleared before performing any operation on the core.

15.9.1.3 Calculating the Total FIFO Size for OTG

[Dedicated FIFO Mode with No Thresholding]

The otg RxFIFO is shared between the host and device. The Host TxFIFOs are also shared with Device IN endpoint TxFIFOs 0 through n.

There are three ways to calculate the total FIFO size. The total FIFO size will depend on whether you support high-bandwidth transfers with high AHB latency.

Method 1

Use this method if you are using the following conditions:

- Minimum FIFO depth allocation
- No support for high-bandwidth endpoints
- The FIFO must equal at least one MaxPacketSize (MPS).

Device RxFIFO = (4 * number of control endpoints + 6) + ((largest USB packet used / 4) + 1 for status information) + (2 * number of OUT endpoints) + 1 for Global NAK

Note : Include the control OUT endpoint in the "number of OUT endpoints."

Host RxFIFO = (largest USB packet used / 4) + 1 for status information + 1 transfer complete

Host Non-Periodic TxFIFO = largest non-periodic USB packet used / 4

Host Periodic TxFIFO = largest periodic USB packet used / 4

Device IN Endpoint TxFIFOs (a separate FIFO is allocated to each IN endpoint) = IN Endpoints Max packet Size / 4

OTG Total RAM = (Device RxFIFO or Host RxFIFO; choose the largest one) + {((Host Non-Periodic TxFIFO + Host peiodic TxFIFO) OR Device IN Endpoint TxFIFO #0 + #1 + #2 + #n); choose the largest one}

Example

The maximum packet size (MPS) is 1,024 bytes for a periodic USB packet and 512 bytes for a non-periodic USB packet. There are three OUT endpoints, three IN endpoints, and one control endpoints. One of the IN endpoints is isochronous.

Device RxFIFO = (4 * 1 + 6) + ((1,024 / 4) +1) + (2 * 4) + 1 = 276

Host RxFIFO = ((1,024 / 4) + 1) + 1 = 258

Host Non-Periodic TxFIFO = (512 / 4) = 128

Host Periodic TxFIFO = (1,024 / 4) = 256

Device IN Endpoint TxFIFO:

FIFO #0= (64/ 4) = 16 (Assuming this is used for EP0)
FIFO #1= (512 / 4) = 128
FIFO #2 = (512/4) = 128
FIFO #3= (1,024 / 4) = 256 (Assuming this is used for Isochronous).

OTG Total RAM = max(276,258) + max(384,528) = 804.

Method 2

Use this method if you are using the recommended minimum FIFO depth allocation with support for high-bandwidth endpoints. This FIFO allocation enables the core to transfer a packet on the USB while the previous (next) packet is simultaneously transferred to the AHB.

This FIFO allocation improves the core's performance .

Device RxFIFO = (4 * number of control endpoints + 6) + 2 * ((largest USB packet used / 4) + 1) + (2 * number of OUT endpoints) + 1

Note: Include the control OUT endpoint in the "number of OUT endpoints."

Host RxFIFO = 2 * ((largest USB packet used / 4) + 1 + 1)

Host Non-Periodic TxFIFO = 2 * (largest non-periodic USB packet used / 4)

Host Periodic TxFIFO = 2 * (largest periodic USB packet used / 4)

Device IN Endpoint-Specific TxFIFOs (a separate FIFO is allocated to each endpoint) = 2*(max_pkt_size for the endpoint)/4.

OTG Total RAM = (Device RxFIFO or Host RxFIFO; choose the largest one) + {((Host Non-Periodic TxFIFO + Host peiodic TxFIFO) OR Device IN Endpoint TxFIFO #0 + #1 + #2 + #n); choose the largest one }

Example

The MPS is 1,024 bytes for a periodic USB packet and is 512 bytes for a non-periodic USB packet. There are three OUT endpoints, three IN endpoints, and one control endpoint. .One of the IN endpoint is Isochronous and the maximum number of periodic data packets per transfer for Endpoint 3 is 3.

Device RxFIFO = (4 * 1 + 6) + 2 * ((1,024 / 4) + 1) + (2 * 4) + 1 = 533

Host RxFIFO = 2 * ((1,024 / 4) + 1 + 1) = 516

Host Non-Periodic TxFIFO = 2 * (512 / 4) = 256

Host Periodic TxFIFO = 2 * (1,024 / 4) = 512

Device IN Endpoint TxFIFO:

FIFO #0 = 2*(64 / 4) = 32(Assuming used for Control Endpoint)
FIFO #1 = 2*(512 / 4) = 256
FIFO #2 = 2*(512 / 4) = 256
FIFO #3 = 2*(1024 / 4) = 256 * 2= 512 (Assuming used for High Bandwidth Endpoint)

OTG total RAM = max(533,516) + max(768,1056)= 1589

Method 3

Use this method if you are using the recommended FIFO depth allocation that supports high-bandwidth endpoints and high AHB latency.

Notes: x = (AHB latency + time to transfer largest packet on AHB) / time to transfer largest packet on USB. The value of x is an integer. Any fractional value is rounded to the nearest integer. For example: x = 20 µs / 17,039 µs = 1.17 µs = 2 µs.

Device RxFIFO = (4 * number of control endpoints + 6) + (x + 1) * ((largest USB packet used / 4) + 1) + (2 * number of OUT endpoints) + 1

Note : Include the control OUT endpoint in the “number of OUT endpoints.”

Host RxFIFO = (x + 1) * ((largest USB packet used / 4) + 1 + 1)

Host Non-Periodic TxFIFO = (x + 1) * (largest non-periodic USB packet used / 4)

Host Periodic TxFIFO = (x + 1) * (largest periodic USB packet used / 4)

Device IN Endpoint-Specific TxFIFOs (a separate FIFO is allocated to each endpoint) = (x+1)*(max_pkt_size for the endpoint)/4e

OTG Total RAM = (Device RxFIFO or Host RxFIFO; choose the largest one) + {((Host Non-Periodic TxFIFO + Host peiodic TxFIFO) OR Device IN Endpoint TxFIFO #0 + #1 + #2 + #n); choose the largest one}

For example, the otg core is operating as a device and receives three isochronous backto-back packets (high bandwidth). The RxFIFO is configured for only 2 MPS (2 kB). On the USB, it takes 2.08 ns (high speed) to transfer 1 bit of data. To transfer 1,024 bytes (8,192 bits), it takes approximately 17,039 µs (2.08 ns * 8,192) to complete the packet transfer from the USB to RxFIFO. After the full packet is in the RxFIFO (assume you are using DMA mode), the DMA begins to transfer the data packet from the RxFIFO to system memory. The AHB latency plus the time to transfer 1,024 bytes on the AHB is 20 µs.

A break-down of the sequence is as follows:

1. First isochronous packet from USB to RxFIFO = 17,039 µs.

2. DMA begins to fetch data from RxFIFO to memory; time = 0.

3. Second isochronous packet from USB to RxFIFO = 17,039 µs.

4. Because there is an extra 1 MPS RxFIFO, the core can receive the next packet regardless of whether the DMA has completed the transfer on the AHB.

5. Third isochronous packet from USB to RxFIFO = 17,039 µs.

If the DMA transfer on the AHB for the first packet is still not complete after 17,039 µs, no FIFO space is available for the third transaction. In this case, the AHB latency is 20 µs; therefore, the transfer of the first packet still has not completed. The OTG core does not issue an interrupt for the transfer completed for the third packet and the transfer times out on the USB side. To compensate for the high AHB latency, you can configure the RxFIFO for 3 more MPS.

Example

The MPS for a periodic USB packet is 1,024 bytes and 512 bytes for a non-periodic USB packet. There are three OUT

endpoints, three IN endpoints, and one control endpoint. One of the IN Endpoint is Isochronous and the maximum number of periodic data packets per transfer for Endpoint 3 is 3. The AHB latency plus the time to transfer 1,024 bytes on the AHB is 20 µs.

Device RxFIFO = (4 * 1 Host Periodic TxFIFO = (2 + 1) * (1,024 / 4) = 768

Host RxFIFO = (2 + 1) * ((1,024 / 4) + 1 + 1) = 774

Host Non-Periodic TxFIFO = (2 + 1) * (512 / 4) = 384

Device IN ENdpoint TxFIFO:

FIFO #0 = (2+1) * (64/4) = 48 (Assuming to be used for Control Endpoint)
FIFO #1 = (2+1) * (512 / 4) = 384
FIFO #2 = (2+1) * (512 / 4) = 384
FIFO #3 = 3* (1,024 / 4) = 256 * 3= 768 (Assuming to be used for Isochronous Endpoint)

OTG total RAM = max(790,774) + max(1152, 1584) = 2374+ 6) + (2 + 1) * ((1,024 / 4) +1) + (2 * 4) + 1 =790

[Dedicated FIFO Mode with thresholding]

The main intention of threshold support are the following

- To have a smaller FIFO size
- To have faster DMA response.

Thresholding is supported only in device mode. So if your device does not have many IN endpoints (not more than 2) OR if your calculated total device FIFO depth without thresholding is not more than the required host FIFO depth, then you are not going to save FIFO space much by enabling thresholding.

It is strongly recommended that you enable dynamic FIFO sizing when thresholding is enabled.

Device RxFIFO = (7 * number of control endpoints + 6) + 2 * ((min(largest USB packet used, Rx_threshold_length) / 4) + 4) + 1 for Global NAK.

Host RxFIFO = (largest USB packet used / 4) + 1 for status information + 1 transfer complete

Host Non-Periodic TxFIFO = largest non-periodic USB packet used / 4

Host Periodic TxFIFO = largest periodic USB packet used / 4

Device IN Endpoint TxFIFOs (a separate FIFO is allocated to each IN endpoint) = min(2 x Transmit Threshold size, IN Endpoints Max packet Size) / 4

OTG Total RAM = (Device RxFIFO or Host RxFIFO; choose the largest one) + {((Host Non-Periodic TxFIFO + Host peiodic TxFIFO) OR Device IN Endpoint TxFIFO #0 + #1 + #2 + #n); choose the largest one}

Example

The maximum packet size (MPS) is 1,024 bytes for a periodic USB packet and 512 bytes for a non-periodic USB packet. There are three OUT endpoints, three IN endpoints, and one control endpoints. One of the IN endpoints is isochronous. The Rx threshold size is 128 bytes and the Tx threshold size is also 128 bytes.

Device RxFIFO = (7 * 1 + 6) + 2* ((128 / 4) +4) + 1 = 86

Host RxFIFO = ((1,024 / 4) + 1) + 1 = 258

Host Non-Periodic TxFIFO = (512 / 4) = 128

Host Periodic TxFIFO = (1,024 / 4) = 256

Device IN Endpoint TxFIFO:

FIFO #0= (64/ 4) = 16 (Assuming this is used for EP0)
FIFO #1= (128/ 4) = 32
FIFO #2 = (128/4) = 32
FIFO #3= (128 / 4) = 32(Asuming this is used for Isochronous).

OTG Total RAM = (Device RxFIFO or Host RxFIFO; choose the largest one) + ((Host Non-Periodic TxFIFO + Host periodic TxFIFO) OR Device IN Endpoint TxFIFO #0 + #1 + #2 + #n; choose the largest one)

OTG_TOTAL_RAM = max(258,86) + max((128+256),(16+32+32+32)) = 258 + 384 = 642.

15.9.2 Dynamic FIFO Allocation

The application can change the RAM allocation for each FIFO during the operation of the core.

Host Mode

In Host mode, before changing FIFO data RAM allocation, the application must determine the following.

- All channels are disabled
- All FIFOs are empty

Once these conditions are met, the application can reallocate FIFO data RAM as explained in "Data FIFO RAM Allocation".

After reallocating the FIFO data RAM, the application must flush all FIFOs in the core using the GRSTCTL.TxFIFO Flush and GRSTCTL.RxFIFO Flush fields. Flushing is required to reset the pointers in the FIFOs for proper FIFO operation after reallocation.

Device Mode

In Device mode, before changing FIFO data RAM allocation, the application must determine the following.

- All IN and OUT endpoints are disabled
- NAK mode is enabled in the core on all IN endpoints
- Global OUT NAK mode is enabled in the core
- All FIFOs are empty

Once these conditions are met, the application can reallocate FIFO data RAM as explained in "Data FIFO RAM Allocation". When NAK mode is enabled in the core, the core responds with a NAK handshake on all tokens received on the USB, except for SETUP packets.

After the reallocating the FIFO data RAM, the application must flush all FIFOs in the core using the GRSTCTL.TxFIFO Flush and GRSTCTL.RxFIFO Flush fields. Flushing is required to reset the pointers in the FIFOs for proper FIFO operation after reallocation.

15.9.3 Core Interrupt Handler

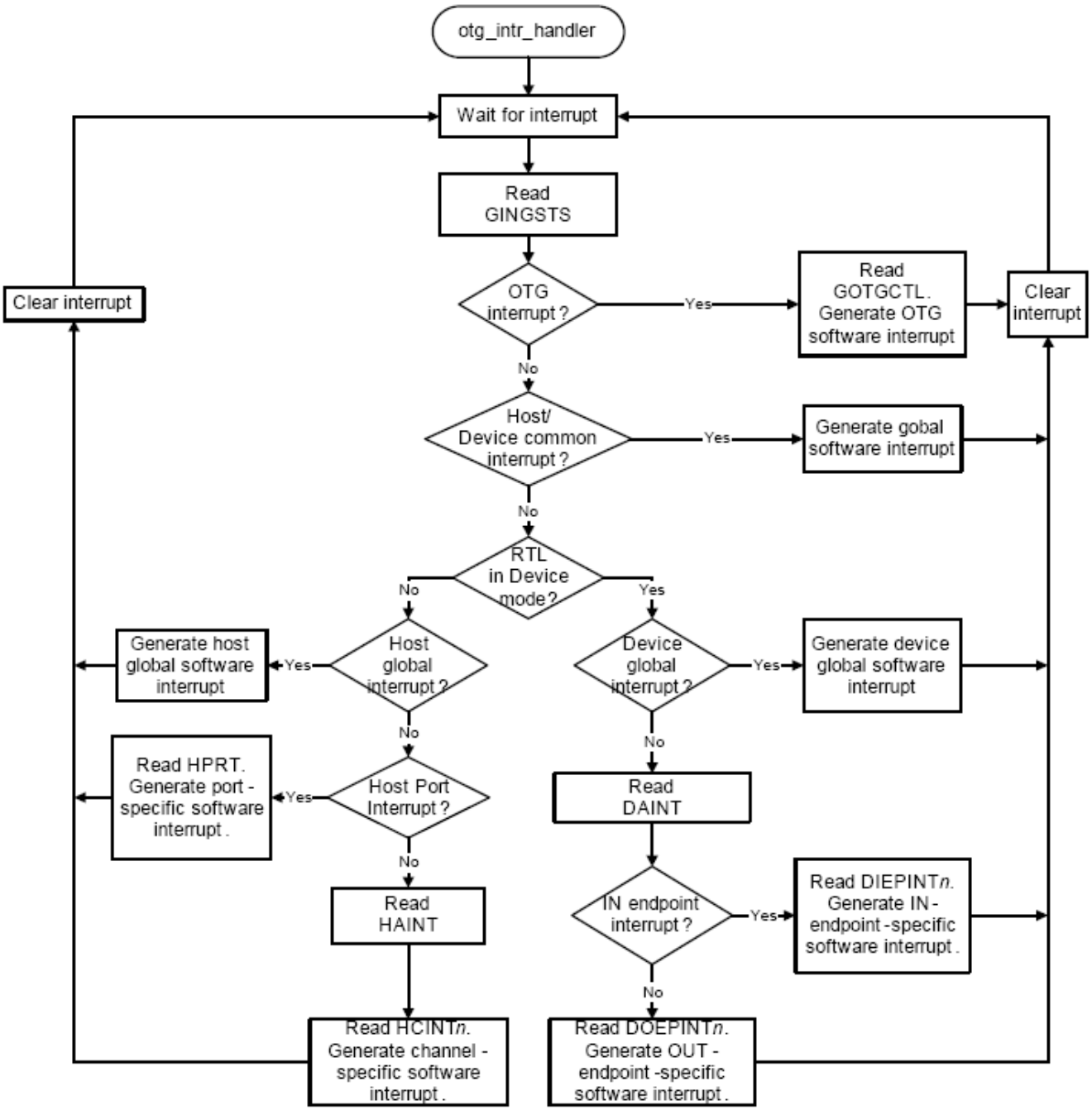


Figure 15.37 Core Interrupt Handler

16 EHI

16.1 Overview

This LSI has the external host interface (EHI) that allows the external host device to be connected to the on-chip system bus. The external host device can be directly connected to 68/80-series interfaces and access the memory area of this LSI. For software based data transfer, EHI can generate the internal interrupt of this LSI, and this LSI can also send interrupt request to the external host controller.

The features of EHI are as follows.

- 68/80 series interface with 8/16bits data can be supported.
- Burst transfer is supported and address can be incremented automatically.
- External host device can generate an internal interrupt of this LSI.
- Interrupt request can be sent to the external host by programming specific bits in EHI control register.
- Semaphore is supported for improving data transfer efficiency.

The interface signals are shown in Table 16.1. HPCSN_L and HPCSN should not be asserted at the same time.

Table 16.1 EHI External Interface Pin

PIN Name	I/O	Function
HPCSN_L	I	Chip select signal 1
HPCSN	I	Chip select signal 0
HPXA[0]	I	Address signal. It is used for switching between normal access and EHIND/EHST register access. Normal access (HPXA [0]= 0): EHI register indicated by EHIND register is accessed. EHIND/EHST access (HPXA[0] = 1) : Writing to EHIND register and reading from EHST register.
HPWRN	I	68-interface: Enable signal 80-interface: Write strobe signal
HPRDN	I	68-interface: Data reading or writing select signal 80-interface: Read strobe signal
HPXD[17:0]	B	Data bus
HPINTO	O	External host interrupt request signal / Ready signal for HPCSN
HPINTO1	O	External host interrupt request signal / Ready signal for HPCSN_L

16.2 Register Description

The EHI registers are shown in Table 16.2. Chip Select 0 (HPCSN) base address is 0xF0570000 and Chip Select 1 (HPCSN_L) base address is 0xF0580000.

Table 16.2 EHI register map(EHI_Base = 0xF0570000, 0xF0580000)

Name	Offset	Int.	Ext.	Initial	Description
EHST	0x00	R/W	R/W	0x00000080	Status register
EHIINT *	0x04	R/W	R/W	0x00000000	Internal interrupt control register
EHEINT*	0x08	R/W	R/W	0x00000000	External interrupt control register
EHA	0x0C	R	R/W	0x00000000	Address register
EHAM	0x10	R/W	R	0x00000000	Address masking register
EHD	0x14	R/W	R/W	0x00000000	Data register
EHSEM*	0x18	R/W	R/W	0x00000000	Semaphore register
EHCFG*	0x1C	R/W	R/W	0x00000000	Configuration registers
EHIND	0x20	R	W	0x00000000	Index register
EHRWCS*	0x24	R	R/W	0x00000000	Read/Write Control/Status register

Note : When the external host device writes to these registers, EHIND [1:0] bits are ignored.

EHI has two clock inputs; one is HCLK, which is for the on-chip system bus interface and the other is ECLK (EHI CLK) which is for interface with the external host.

Before the on-chip CPU accesses these registers, HCLK and ECLK should be enabled.

ECLK should be enabled for the external host to access these registers except for EHST register. Refer to 16.3.1 Access to EHI registers for more information.

EHST (Status Register) **EHI_BASE+0x00**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
			RFNE	WFNE	WFF	FUR	FOR	RDY	ST[6:0]						

Bit	Name	Initial	Int.	Ext.	Description
31-13	-	0	R	R	These bits are read as 0.
12	RFNE	0	R	R	It represents that Read-FIFO is not empty. When it is equal to 0, Read-FIFO is empty.
11	WFNE	0	R	R	It represents that the Write-FIFO is not empty. When it is equal to 0, Write-FIFO is empty.
10	WFF	0	R	R	It represents that the Write-FIFO is full.
9	FUR	0	R/C	R	It represents that Read-FIFO underrun is occurred. It is cleared when the on-chip CPU writes 1 to this bit.
8	FOR	0	R/C	R	It represents that Write-FIFO overrun is occurred. It is cleared when the on-chip CPU writes 1 to this bit.
7	RDY	1	R	R	It represents that an external host device can access the on-chip system bus.
6-0	ST	0	R/W	R/W	It is read and written by an external host connected to EHI and by the on-chip CPU.

The external host device has two methods for reading this register. One is that the external host device issues the read operation while HPXA is high. The other is that the external host sets EHIND to 0x00 or 0x01 (EHST offset) and issues the read operation while HPXA is low.

When using the 8-bit interface, EHST[7:0] can be read by the external host device while HPXA is high regardless of ECLK. EHST[15:8] is read by the external host device while HPXA is low and EHIND is set to 0x01. Therefore, when ECLK is disabled, the external host device can not read EHST[15:8].

When using the 16-bit interface, EHST[15:0] can be read by the external host device while HPXA is high. Therefore the external host device can read EHST[15:0] regardless of ECLK.

EHIINT (Internal Interrupt Control Register) EHI_BASE+0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								IIRQ_ST							IIRQ

Bit	Name	Initial	Int.	Ext.	Description
31-8	-	0	R	R	These bits are read as 0.
7-1	IIRQ_ST	0	R/W	R/W	It is only updated by software. It can specify the interrupt request number.
0	IIRQ	0	R/W	R/W	When it is equal to 1, EHI generates the interrupt request and it is sent to the on-chip interrupt controller. Note that it should be cleared manually. The external host device may need to check that the previous issued interrupt is processed by the on-chip CPU before issuing the new interrupt request. It is only updated by software.

EHIINT register is used to generate interrupts from the external host device. When the interrupt is generated, the on-chip CPU processes the interrupt service routine and should be clear EHIINT.IIRQ bit. Therefore, the external host can detect whether the requested interrupt is processed as it reads EHIINT register.

EHEINT (External Interrupt Control Register) EHI_BASE+0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								EIRQ_ST							EIRQ

BIT	Name	Initial	Int.	Ext.	Descriptions
31-8	-	0	R	R	These bits are read as 0
7-1	EIRQ_ST	0	R/W	R/W	This specifies EIRQ interrupt source. The external host device can detect interrupt source via EIRQ_ST.
0	EIRQ	0	R/W	R/W	If HPINT is connected to the external interrupt input of the external host, this LSI can send interrupt request to the external host. When EHCFG.RDYE is equal to 0, EHEINT.EIRQ value is output through HPINT pin. Otherwise, this bit is not applicable.

EHEINT is used to issue interrupts to the external host via HPINT pin by the on-chip CPU. HPINT pin is used for interrupt request pin when EHCFG.RDYE is set to 0.

EHA (Address Register) EHI_BASE+0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EHA[15:2]														0	

Bit	Name	Initial	Int.	Ext.	Description
31-2	-	0	R	R/W	The address for the on-chip system bus access of this LSI.
1-0	-	0	R	R	These bits are always 0.

EHA maps to the memory space of this LSI when the external host accesses the system bus of this LSI.

When the external host issues the incremental writing operation (EHRWCS.AI = 1 and EHRWCS.RW = 1), EHA increments automatically during data transfer and when it is completed (EHRWCS.RW = 0), EHA has the incremented address as the number of written data.

But, when the incremental reading operation is completed, EHA do not has the incremented address as the number of read data, but the prefetched address. Therefore, when the new incremental reading operation is issued, EHA should be updated.

EHA can be masked by EHAM register. Refer to the next page, EHAM register description, for more information.

EHAM (Address Masking Register) EHI_BASE+0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHAM[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EHAM[15:2]														0	

Bit	Name	Initial	Int.	Ext.	Description
31-2	-	0	R/W	R	It masks EHA register. Therefore, the effective address on the on-chip system bus is EHA[31:2] & ~EHAM[31:2].
1-0	-	0	R	R	These bits are read as 0.

EHA can be masked by EHAM register and then the effective address is EHA[31:2] & ~EHAM[31:2]. When EHRWCS.AI is set to 1, the generated address of every transfers is (EHA[31:2] + 1) & ~EHAM[31:2].

For example, when EHA = 0x1000000, EHRWCS.AI=1, and EHAM = 0x0F00, EHA has the address between 0x10000000 and 0x100000FC.

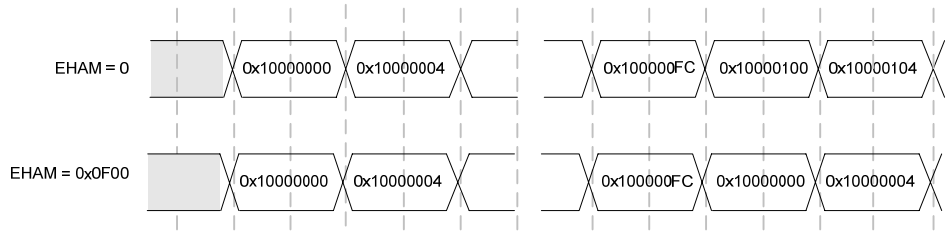


Figure 16.1 Example of EHAM usage

EHD (Data Register) EHI_BASE+0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHD[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EHD[15:0]															

Bit	Name	Initial	Int.	Ext.	Description
31-0	-	0	R/W	R/W	While EHRWCS.RW = 1 or 2, the external host device accesses the memory area of this LSI using EHA register. In this case, it is used for writing or reading data.

EHD is used to transfer data for system bus access mode.

EHSEM (Semaphore Register) EHI_BASE+0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								ST						FLG	

Bit	Name	Initial	Int.	Ext.	Description
7-2	ST	0	-	-	FLG = 00b Reading only: On-chip CPU, External host device FLG = 01b Reading/Writing: External host device Reading only: On-chip CPU FLG = 10b Reading/Writing: On-chip CPU Reading only: External host device
1-0	FLG	0	-	-	If EHSEM is read and then FLG is 00b, this means EHSEM is not occupied by any master. If the external host device reads EHSEM which is not occupied by this LSI, then EHSEM.FLG becomes 01b. If the on-chip CPU reads EHSEM which is not occupied by the external host device, then EHSEM.FLG becomes 10b. If the on-chip CPU and the external host read EHSEM simultaneously, return value for external host is 00b and it for the on-chip CPU is 01b. 00b (NOT OCCUPIED) Reading only: On-chip CPU, External host device 01b (EXTERNAL HOST) Reading/Writing: External host device Reading only: On-chip CPU 10b (ON-CHIP CPU) Reading/Writing: On-chip CPU Reading only: External host device 11b N/A

EHCFG (Configuration Register) **EHI_BASE+0x1C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								WIRQ	CSIRQ		RDYP	RDYE	BW		MD

Bit	Name	Initial	Int.	Ext.	Description
31-8	-	0	R	R	These bits are read as 0.
7	WIRQ	0	R/W	R/W	Writing operation becomes the internal interrupt source. When HPCSN and HPWRN are asserted in 80-mode or HPCSN, HPWRN(E), and HPRDN(R/W) in 68-mode, the internal interrupt request is generated. For preventing additional interrupts by write operation, this bit should be cleared manually.
6	CSIRQ	0	R/W	R/W	HPCSN becomes the internal interrupt source. When HPCSN is asserted regardless of HPWRN and HPRDN, the internal interrupt request is generated. For preventing additional interrupts by HPCSN, this bit should be cleared manually.
5	-	0	R	R	This bit reads as 0.
4	RDYP	0	R/W	R/W	When RDYE is equal to 1, EHST.RDY ⊕ EHCFG.RDYP is outputted through HPINT. Therefore, it is only valid when RDYE is equal to 1. Refer to RDYE description.
3	RDYE	0	R/W	R/W	0: HPINT is used for the external interrupt. EHEINT.EIRQ is outputted through HPINT. 1: HPINT is used for READY signal. In this case, RDYP is valid.
2	BW	0	R/W	R/W	0 : 8-bit interface mode 1 : 16-bit interface mode
2-1	BW	0	R/W	R/W	00b: 8-bit interface mode 01b: Reserved 10b: 16-bit interface mode 11b: 18-bit interface mode 18-bit data is written to and is read from lower 18bits of 32-bit word memory. Therefore, upper 14bits in 32-bit word memory cannot be used.
0	MD	0	R/W	R/W	0 : 80 interface mode 1 : 68 interface mode

EHCFG configures the overall EHI operation. The on-chip CPU should select the interface mode, 80-interface mode or 68-interface mode.

When the external device can not write to the EHI registers, because of clock configuration or interface timing problem, it can issues interrupts for the on-chip CPU using EHCFG.WIRQ or EHCFG.CSIRQ (Figure 16.2).

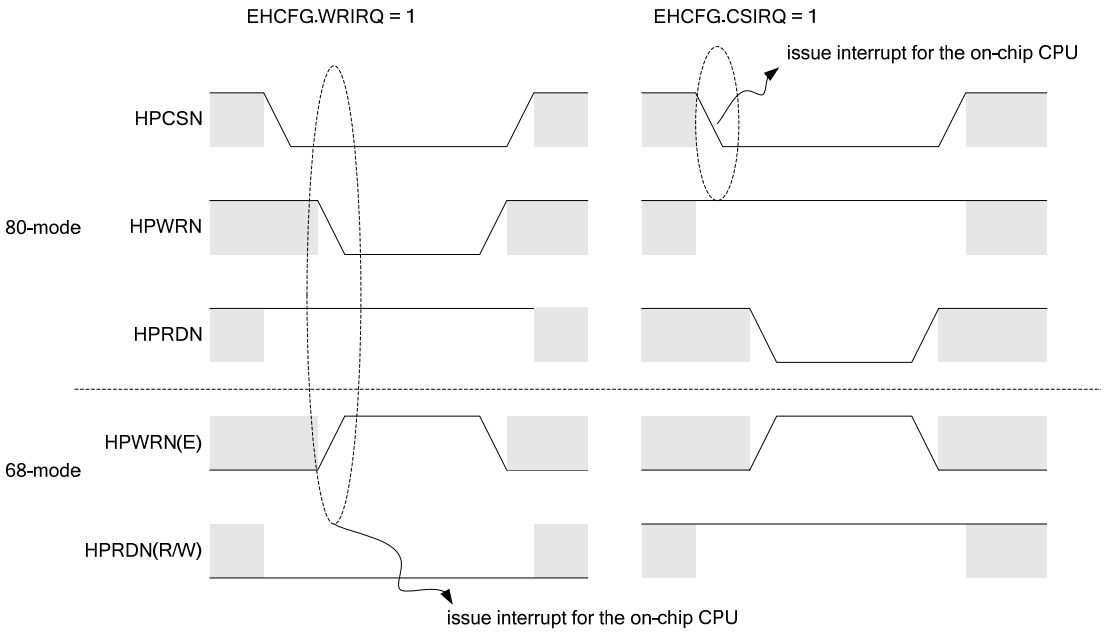


Figure 16.2 Interrupt request using EHCFG.WRIRQ and CSIRQ

EHIND (Index Register) **EHI_BASE+0x20**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								EHIND[7:0]							

Bit	Name	Initial	Int.	Ext.	Description
31-8	-	0	R	R	These bits read as 0.
7-0	EHIND	0	R	W	It selects the internal register for the external host. It can be only written by the external host device while HPXA is high. Refer to Table 16.2 for value corresponding to each register. 0x00: EHST 0x04: EHIINT 0x08: EHEINT 0x0C: EHA 0x10: EHAM 0x14: EHD 0x18: EHSEM 0x1C: EHCFG 0x20: EHIND 0x24: EHRWCS When writing to EHIINT , EHEINT, EHSEM, EHCFG, and EHRWCS, EHIND[1:0] is ignored. For example, when writing to EHIINT[7:0] register, EHIND can be 4, 5, 6, or 7

EHIND is used to index other EHI registers and only written by the external host while HPXA is high.

EHRWCS (Read/Write Control/Status Register) **EHI_BASE+0x24**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								AI	LK	RW		BSIZE			

Bit	Name	Initial	Int.	Ext.	Description
31-8	-	0	R	R	These bits are read as 0.
7	AI	0	R	R/W	EHA is auto-incremented while EHRWCS.RW is 1 or 2.
6	LK	0	R	R/W	Once the external host device is a master of the system bus of this LSI, bus-handover cannot be occurred until burst transfer as EHRWCS.BSIZE is completed.
5-4	RW	0	R	R/W	0 : Access to EHI registers 1 : Writing to the system bus of this LSI. When BSIZE = 0(single transfer), it is cleared automatically. 2 : Reading from the system bus of this LSI. When BSIZE = 0, it is cleared automatically. 3 : Undefined
3-0	BSIZE	0	R	R/W	It specifies how many words(1word = 32bits) will be transferred and EHRWCS.BSIZE + 1 words will be transferred. It should be programmed before EHRWCS.RW is set to 2 (Reading operation) or 1(Writing operation). For continuous burst transfer, EHST.RDY may need to be checked every burst size transfer.

EHRWCS register is used to control the on-chip system bus access.

Before EHRWCS.RW is issued for reading or writing operation, other fields of this register and EHA should be configured. Refer to 16.3.2 Access to the On-chip System Bus about reading and writing operation for the on-chip system bus.

16.3 Operation

16.3.1 Access to EHI registers

Table 16.3 Access to the EHI

HCLK	ECLK	Reading From EHI register		Writing to EHI register		access to the on-chip system bus
		on-chip	Ext.Host	on-chip	Ext.Host	Ext.Host
X	X	X	△	X	X	X
X	O	X	O	X	O	X
O	X	X	△	X	X	X
O	O	O	O	O	O	O

△ = When 8bits interface is used, EHST[7:0] can be only read. When 16bits interface is used, EHST[15:0] can be only read.

ECLK and HCLK should be enabled for the on-chip CPU to access the EHI registers. And the external host can only access the EHI registers when ECLK is enabled except for EHST register. When ECLK is disabled, the external host can only read EHST[15:0] register in the 16-bit interface mode and EHST[7:0] register in the 8-bit interface mode. The relationship between clocks and accessibility is shown in Table 16.3.

Most of EHI registers are accessed by indirect addressing using EHIND register. To write to or read from an EHI register except for EHIND and EHST registers, EHIND register should be set to its offset value in advance and then HPXA should be low during reading or writing operation. If the external host intends to write data to EHIND , it should drive HPXA pin to high. When the external host device issues the reading operation while HPXA is high, it reads EHST register. Note that EHIND register is write-only. When the external host device intents to write to EHST register, it should also use EHIND register. The following shows how to write data (=0x12345678) to EHA register.

(68) Write EHA[15:0] offset value(=0x0C) while HPXA = 1. EHIND register indicates EHA[15:0] register.
(69) Write 0x5678 for EHA[15:0] while HPXA=0.
(70) Write EHA[31:16] offset value(=0x0E) while HPXA = 1. EHIND register indicates EHA[31:16] register.
(71) Write 0x1234 for EHA[31:16] while HPXA = 0.

Figure 16.3 and Figure 16.4 show that the external host device writes to and reads from EHA register.

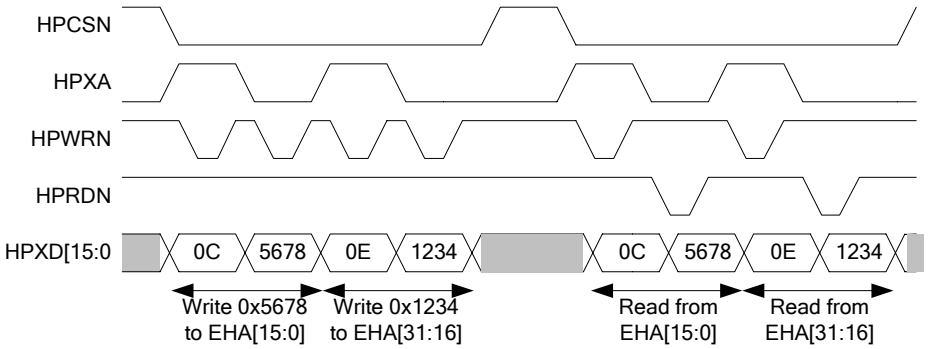


Figure 16.3 Example of writing / reading operation (80 interface, 16bits)

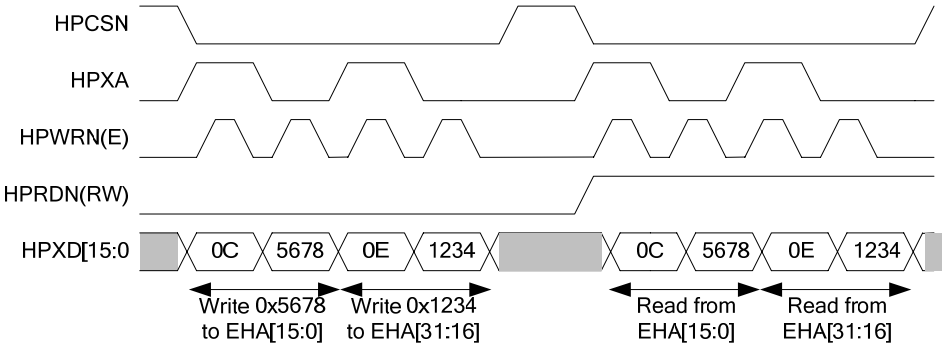


Figure 16.4 Example of writing / reading operation (68 interface, 16bits)

16.3.2 Access to the On-chip System Bus

The external host device that is connected to this LSI via EHI can access the on-chip system bus. This connection supports word-aligned (32-bits) burst transfer. When the external host device access the on-chip system bus, HCLK and ECLK should be enabled in advance.

The length of burst transfer is determined by EHRWCS.BSIZE. Once transfer is occurred, the length of burst cannot be modified until the requested transfer is completed. Therefore, EHRWCS.BSIZE, EHRWCS.AI, and EHRWCS.LK can be only modified while EHRWCS.RW = 0. If EHST.RDY is checked every burst transfer, the FIFO underrun or overrun cannot be occurred. But, if the burst transfers are intended to be issued without EHST.RDY check to improve data transfer efficiency, HCLK of this LSI should be fast enough not to underrun and overrun. FIFO underrun and overrun can be detected via EHST.FUR and EHST.FOR respectively.
Figure 16.5 shows how to program the external host device for access to the on-chip system bus.

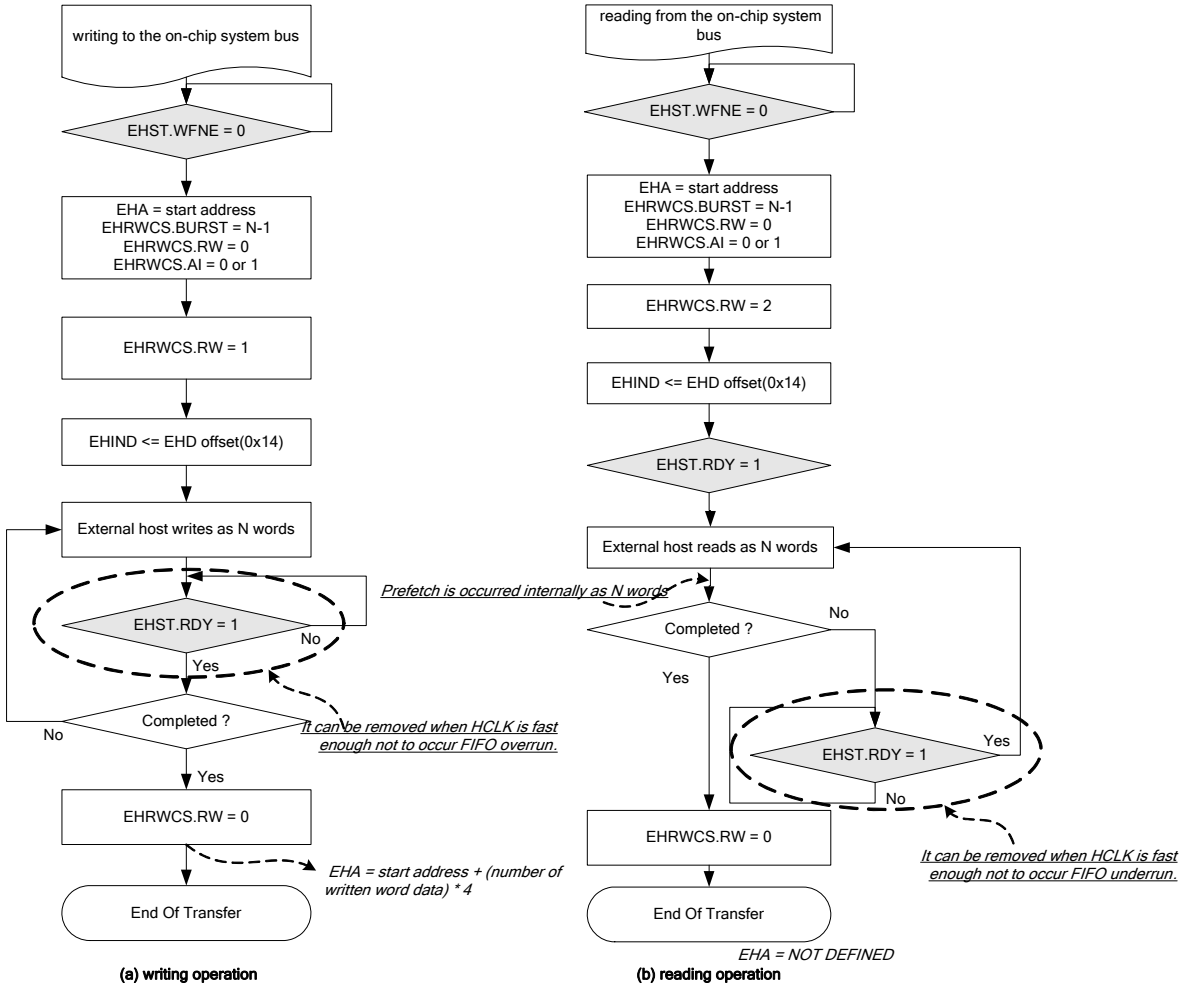


Figure 16.5 Access to the On-chip System Bus

Write-operation

The external host should configure writing operation before writing data; start address, address auto increment, lock or normal, and burst length.

The start address should be written to EHA register. When address auto increment is enable, address of the on-chip system bus is increment automatically every word transfer. This generated address is masked by EHAM. Therefore, address on the on-chip system bus is next EHA = (EHA[31:2] + 1) & ~EHAM[31:2]. When the writing operation is completed (EHRWCS.RW = 0), EHA has start address + number of written data.

EHRWCS.BSIZE specifies the maximum of burst length on the on-chip system. When the lock transfer mode is enabled, the bus handover is not occurred during burst transfer. But, when lock transfer mode is disabled and a higher priority master requests bus access, the bus handover is occurred and EHI burst transfer is terminated. And when the higher priority master looses access to bus, EHI burst transfer is continued. But, when the lock transfer mode is enabled, the overall system performance of the on-chip can be degraded.

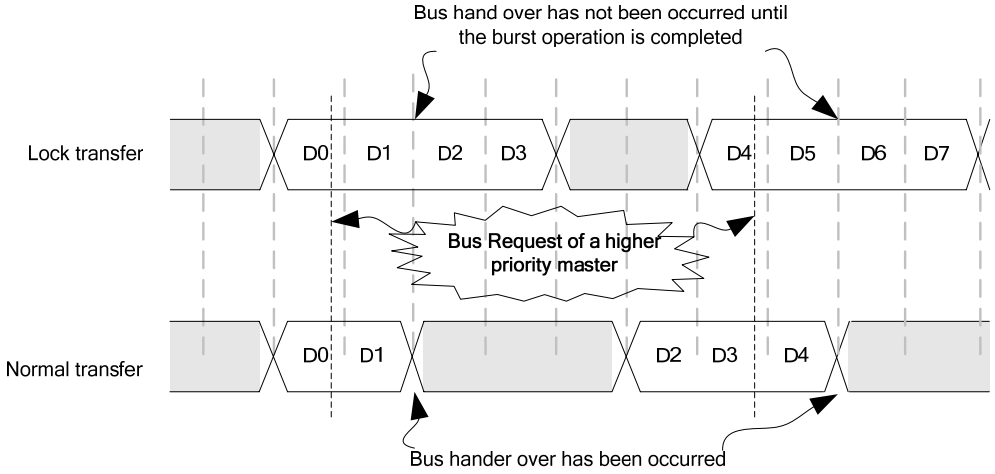


Figure 16.6 Lock tranfser vs Normal transfer (EHRWCS.BSIZE=3)

After configuring writing operation, the external host enables writing operation as EHRWCS.RW is set to 1 and EHIND is set to EHD offset (0x14) sequentially. From now on, the external host can write data to the on-chip system bus while HPXA is low. For finishing the writing operation, the external host should set EHRWCS.RW to 0. After that, to confirm that all of written data arrived at destination area, the external host can check whether EHST.RDY is equal to 1.

The auto-increment writing operation which has that start address is 0x12345678, and the burst length is 8 is shown in Figure 16.7.

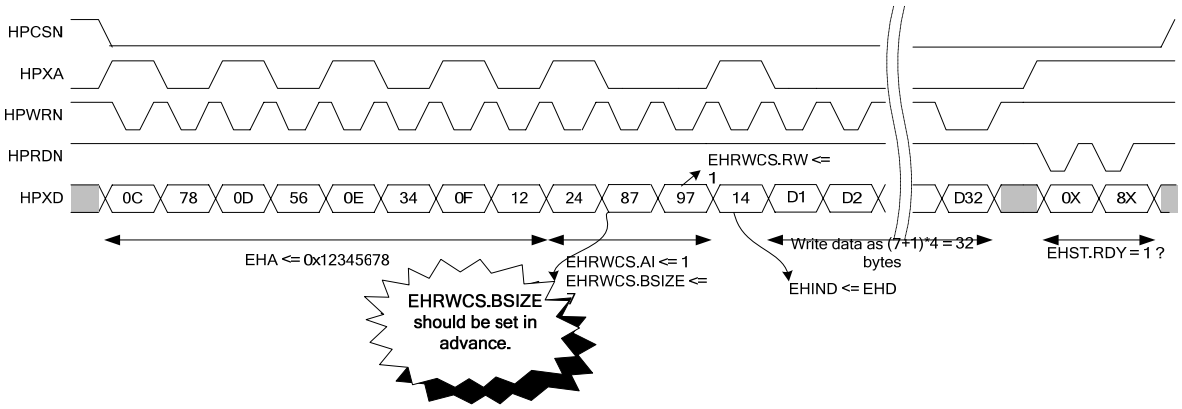


Figure 16.7 Example of Writing to the On-chip System Bus (80 interface, 8bits)

Read-operation

The configuration of reading operation is similar to that of the writing operation; start address, address auto increment, lock or normal, and burst length.

EHST.RDY may be checked every EHRWCS.BSIZE+1 words. If HCLK is fast enough not to underrun, EHST.RDY do not need to be check.

The auto-increment reading operation which has that start address is 0x12345678, and burst length is 8 is shown in Figure 16.8.

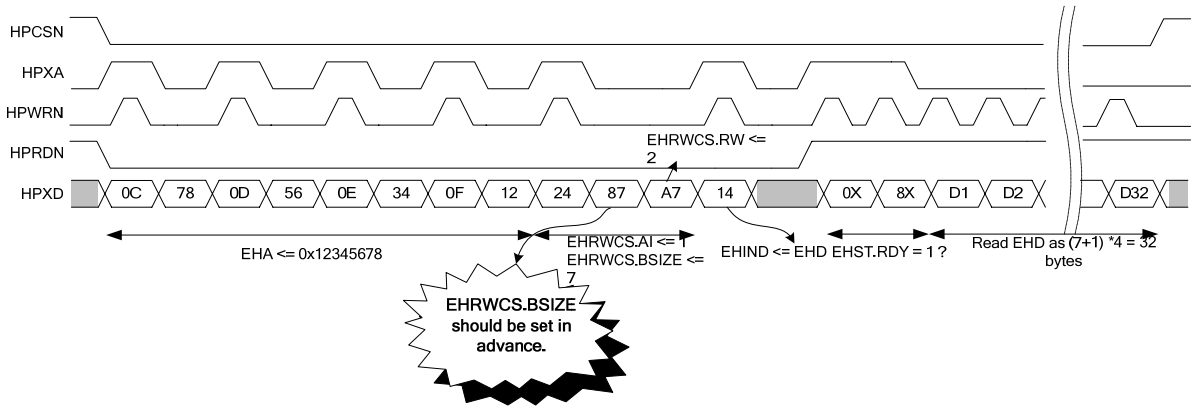


Figure 16.8 Example of Reading from the on-chip System Bus (68 interface, 8bits)

Notice that EHRWCS.RW is cleared to 0 automatically after the single reading and writing operation.

16.3.3 Entrance to the Critical Section Using the Semaphore Register

EHI supports the semaphore for improving data transfer efficiency and EHSEM register is used.

```
int get_sem (void)
{
        if (EHSEM.FLG & 1)
        {
                return FALSE;
        }
        else
        {
                return TRUE;
        }
}

void release_sem (void)
{
        EHSEM.FLG = 0;
}
```

On-Chip CPU

```
int get_sem (void)
{
        sem = read EHSEM reg.;
        if (sem & 2)
        {
                return FALSE;
        }
        else
        {
                return TRUE;
        }
}

void release_sem (void)
{
        write 0 to EHSEM.FLG;
}
```

External Host

Figure 16.9 Pseudo code for getting and releasing a semaphore

16.3.4 Interrupt Request

The EHI has three kinds of interrupt requests; external software interrupt request, internal software interrupt request, and internal CS/WR interrupt request

The first, the external software interrupt request is generated via HPINT pin. Therefore, the HPINT pin should be configured as interrupt request pin. It is default value. After this configuration, the HPINT pin is controlled by EHEINT.IRQ bit directly.
The second, the external host can request an interrupt to the on-chip CPU. This internal software interrupt request is issued when EHIINT .IIRQ is set to 1.

Finally, internal CS/WR interrupt request uses HPCSN, HPWRN, HPRDN signals as interrupt source. For using HPCSN signal as the interrupt source, EHCFG.CSIRQ bit should be set to 1. When EHCFG.CSIRQ is equal to 1 and HPCSN is

asserted, the interrupt request is issued. The interrupt service routine of the on-chip CPU should clear EHCFG.CSIRQ to 0 not to generate the additional interrupt request by HPCSN signal. In the case of using writing operation as interrupt source, EHCFG.WRIRQ bit should be set to 1. After that, when a writing operation is issued (80-mode: HPCSN=0 HPWRN=0, 68-mode: HPCSN=0, HPWRN=1, HPRDN=0), the interrupt is generated. The external host can issue this interrupt request regardless of ECLK.

17 GPSB (General Purpose Serial Bus)

17.1 Functional Description

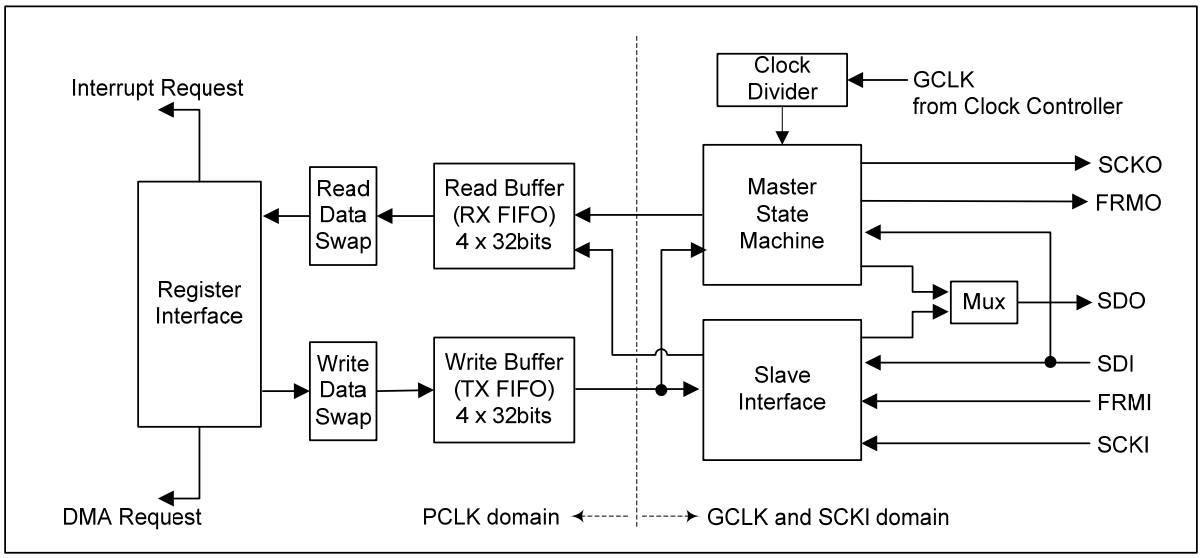


Figure 17.1 GPSB Interface Block Diagram

The Figure 17.1 shows the block diagram of GPSB interface circuit. The following figure shows the overall hardware block diagram for GPSB.

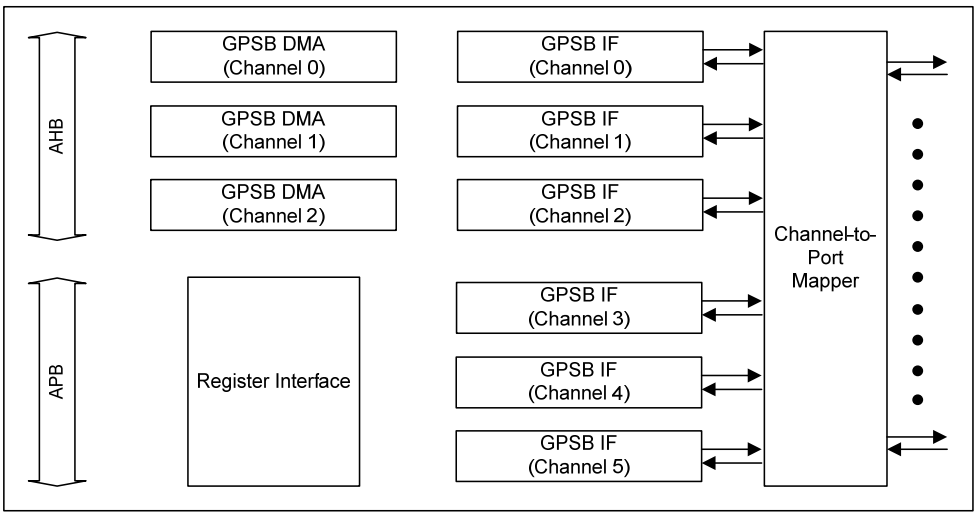


Figure 17.2 Overall GPSB Block Diagram

The TCC8900 provides 6 channel General Purpose Serial Bus Controller, which can be configured as a SPI compatible master/slave and MPEG-2 TS interface. The channel 0,1 and 2 have the DMA function which can transfer the bulk data. The channel 3,4 and 5 can interface the general purpose DMA for fast and bulk data transfer. The GPSB DMA can be used for MPEG2-TS serial interface with PID matching function.

The “Channel-to-Port Mapper” is for mapping the hardware channel and external ports. Each channel can be mapped to one of the ports.

17.2 Feature

The feature of GPSB in TCC8900 supports Motorola SPI, TI SSP compatible timing and changes the programmable master / slave configuration, timing parameters, data bit width, shift direction, polarity, data swap and so on.

The frequency of the SCKO pin is determined by the Clock Divider block which uses GCLK as clock source. The GCLK is generated by the Clock Controller Module.

17.3 Register Description

Table 17.1 GPSB Register Map (Base Address = 0xF0536000)

Ch	Name	Addr. Offset	Type	Reset	Description
Channel 0	PORT	0x000	R/W	0x0000	Data port
	STAT	0x004	R/W	0x0000	Status register
	INTEN	0x008	R/W	0x0000	Interrupt enable
	MODE	0x00C	R/W	0x0004	Mode register
	CTRL	0x010	R/W	0x0000	Control register
	EVTCTRL	0x014	R/W	0x0000	Counter & Ext. Event Control
	CCV	0x018	R	0x0000	Counter Current Value
	TXBASE	0x020	R/W	0x0000	TX base address register
	RXBASE	0x024	R/W	0x0000	RX base address register
	PACKET	0x028	R/W	0x0000	Packet register
	DMACTR	0x02C	R/W	0x0000	DMA control register
	DMASTR	0x030	R/W	0x0000	DMA status register
	DMAICR	0x034	R/W	0x0000	DMA interrupt control register
Channel 1	PORT	0x100	R/W	0x0000	Data port
	STAT	0x104	R/W	0x0000	Status register
	INTEN	0x108	R/W	0x0000	Interrupt enable
	MODE	0x10C	R/W	0x0004	Mode register
	CTRL	0x110	R/W	0x0000	Control register
	EVTCTRL	0x114	R/W	0x0000	Counter & Ext. Event Control
	CCV	0x118	R	0x0000	Counter Current Value
	TXBASE	0x120	R/W	0x0000	TX base address register
	RXBASE	0x124	R/W	0x0000	RX base address register
	PACKET	0x128	R/W	0x0000	Packet register
	DMACTR	0x12C	R/W	0x0000	DMA control register
	DMASTR	0x130	R/W	0x0000	DMA status register
	DMAICR	0x134	R/W	0x0000	DMA interrupt control register
Channel 2	PORT	0x200	R/W	0x0000	Data port
	STAT	0x204	R/W	0x0000	Status register
	INTEN	0x208	R/W	0x0000	Interrupt enable
	MODE	0x20C	R/W	0x0004	Mode register
	CTRL	0x210	R/W	0x0000	Control register
	EVTCTRL	0x214	R/W	0x0000	Counter & Ext. Event Control
	CCV	0x218	R	0x0000	Counter Current Value
	TXBASE	0x220	R/W	0x0000	TX base address register
	RXBASE	0x224	R/W	0x0000	RX base address register
	PACKET	0x228	R/W	0x0000	Packet register
	DMACTR	0x22C	R/W	0x0000	DMA control register
	DMASTR	0x230	R/W	0x0000	DMA status register
	DMAICR	0x234	R/W	0x0000	DMA interrupt control register
Channel 3	PORT	0x300	R/W	0x0000	Data port
	STAT	0x304	R/W	0x0000	Status register
	INTEN	0x308	R/W	0x0000	Interrupt enable
	MODE	0x30C	R/W	0x0004	Mode register
	CTRL	0x310	R/W	0x0000	Control register
	EVTCTRL	0x314	R/W	0x0000	Counter & Ext. Event Control
	CCV	0x318	R	0x0000	Counter Current Value
Channel 4	PORT	0x400	R/W	0x0000	Data port
	STAT	0x404	R/W	0x0000	Status register
	INTEN	0x408	R/W	0x0000	Interrupt enable
	MODE	0x40C	R/W	0x0004	Mode register
	CTRL	0x410	R/W	0x0000	Control register
	EVTCTRL	0x414	R/W	0x0000	Counter & Ext. Event Control
	CCV	0x418	R	0x0000	Counter Current Value
Channel 5	PORT	0x500	R/W	0x0000	Data port
	STAT	0x504	R/W	0x0000	Status register
	INTEN	0x508	R/W	0x0000	Interrupt enable
	MODE	0x50C	R/W	0x0004	Mode register
	CTRL	0x510	R/W	0x0000	Control register
	EVTCTRL	0x514	R/W	0x0000	Counter & Ext. Event Control
	CCV	0x518	R	0x0000	Counter Current Value
Port Config	PCFG0	0x800	R/W	0x03020100	Port Configuration Register 0
	PCFG1	0x804	R/W	0x00000504	Port Configuration Register 1
	CIRQST	0x808	R	0x0000	Channel IRQ Status Register

PID Table	PIDT	0xF00	R/W	-	

PORT Register (PORT) **0xF0536*n*[20]00**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA[15:0]															

Data Read / Write Port. Any data written to this register is sent to the write(TX) FIFO. Any data read from this register is from the read(RX) FIFO. Read data is valid only if RBVCNT is not zero. Although this port can be written even when the core disabled, no serial bus cycle is generated until the core is enabled with valid parameters.

[20] n = Channel Number 0 ~ 5

Status Register (STATUS) 0xF0536*n*[21]04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-				WBVCNT				-				RBVCNT			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-							WOR	RUR	WUR	ROR	RF	WE	RNE	WTH	RTH

Field	Name	Reset	RW	Description
31 ~ 28	-	-	-	Undefined
27 ~ 24	WBVCNT	-	RW	Write(Transmit) FIFO valid entry count Maximum value dependent on the data bit width
23 ~ 20	-	-	-	Undefined
19 ~ 16	RBVCNT	-	RW	Read(Receive) FIFO valid entry count Maximum value dependent on the data bit width
15 ~ 9	-	-	-	Undefined
8	WOR	-	R/C	Write FIFO over-run error flag When writing 1, it is cleared. If INTEN.RC = 1, it is also cleared when reading it.
7	RUR	-	R/C	Read FIFO under-run error flag When writing 1, it is cleared. If INTEN.RC = 1, it is also cleared when reading it.
6	WUR	-	R/C	Write FIFO under-run error flag When writing 1, it is cleared. If INTEN.RC = 1, it is also cleared when reading it.
5	ROR	-	R/C	Read FIFO over-run error flag When writing 1, it is cleared. If INTEN.RC = 1, it is also cleared when reading it.
4	RF	-	R	Read FIFO full flag
3	WE	-	R	Write FIFO empty flag
2	RNE	-	R	Read FIFO not empty flag
1	WTH	-	R	Write FIFO valid entry count is under threshold.
0	RTH	-	R	Read FIFO valid entry increased over threshold.

[21] n = Channel Number 0 ~ 5

Interrupt Enable Register (INTEN) 0xF0536*n*[22]08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DW	DR	--		SHT	SBT	SHR	SBR	-	CFGWTH			-	CFGRTH		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RC	--						IRQEN								

Field	Name	RW	Reset	Description
31	DW	RW	0	DMA request enable for TX FIFO '1' for enable, '0' for disable
30	DR	RW	0	DMA request enable for RX FIFO '1' for enable, '0' for disable
27	SHT	RW	0	TX half-word swap in word
26	SBT	RW	0	TX byte swap in half-word
25	SHR	RW	0	RX half-word swap in word
24	SBR	RW	0	RX byte swap in half-word
22 ~ 20	CFGWTH	RW	0	Transmit FIFO threshold for interrupt/DMA request
18 ~ 16	CFGRTH	RW	0	Receive FIFO threshold for interrupt/DMA request
15	RC	RW	0	Clear status[8:0] at the end of read cycle. When this bit is set as "1", status[8:0] is all cleared whenever GPSBSTAT register is read.
14 ~ 9	-	-	-	Undefined
8 ~ 0	IRQEN	RW	0	Interrupt enable signals [8] : TX FIFO over-run error [7] : RX FIFO under-run error [6] : TX FIFO under-run error [5] : RX FIFO over-run error [4] : RX FIFO full [3] : TX FIFO empty [2] : RX FIFO not empty [1] : Valid entry count of TX FIFO is under threshold [0] : Valid entry count of RX FIFO is more than threshold

[22] n = Channel Number 0 ~ 5

Mode Register (MODE) **0xF0536*n*[23]0C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIVLDV								TRE	THL	TSU	PCS	PCD	PWD	PRD	PCK
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRF	CWF	-	BWS					SD	LB	SDO	CTF	EN	SLV	MD	

Field	Name	RW	Reset	Description
31 ~ 24	DIVLDV	RW	0	Clock divider load value FSCKO = FGCLK / ((DIVLDV+1)*2)
23	TRE	RW	0	Master recovery time tRECV = (TRE + 1) * (SCKO period)
22	THL	RW	0	Master hold time tHOLD = (THL + 1) * (SCKO period)
21	TSU	RW	0	Master setup time tSETUP = (TSU + 1) * (SCKO period)
20	PCS	RW	0	Polarity control for CS(FRM) – Master Only ‘0’ for active low (default) / must be ‘0’ for SSP ‘1’ for active high
19	PCD	RW	0	Polarity control for CMD(FRM) – Master only ‘0’ for active low / must be ‘1’ for SSP ‘1’ for active high
18	PWD	RW	0	Polarity control for transmitting data – Master Only ‘0’ for falling edge of SCK ‘1’ for rising edge of SCK / must be ‘1’ for SSP
17	PRD	RW	0	Polarity control for receiving data – Master only ‘0’ for rising edge of SCK ‘1’ for falling edge of SCK / must be ‘1’ for SSP
16	PCK	RW	0	Polarity control for serial clock ‘0’ for master SCKO start from “low” for SPI timing 0, and “high” for SSP timing ‘0’ for slave SCKI not inverted For SPI timing 0 and 3 ‘1’ for master SCKO starts from “1” For SPI timing 2 and 3. ‘1’ for slave SCKI inverted. For SPI timing 1, 2 and SSP timing
15	CRF	RW	0	Clear receive FIFO counter
14	CWF	RW	0	Clear transmit FIFO counter
12 ~ 8	BWS	RW	0	Bit width selection Data bit width == BWS + 1 Valid range for BWS is 7 ~ 31 The FIFOs are configured according to BWS[4] as follows, “BWS[4]==1”, 4x32 bit. “BWS[4]==0”, 8x16bits.
7	SD	RW	0	Data shift direction control ‘0’ for shift left (MSB first)

[23] n = Channel Number 0 ~ 5

6	LB	RW	0	Data loop-back enable * SDO is feedback to SDI internally and the incoming data from external I/O is ignored. SDO output is not affected by this bit.
5	SDO	R/W	0	SDO output disable (slave mode only) '0' for enable '1' for disable
4	CTF	RW	0	Continuous transfer mode enable '0' for single mode, '1' for continuous mode *If set to continuous mode, the CS signal keeps the active state until that 'CTF' is cleared and transmit FIFO is empty.*
3	EN	RW	0	Operation enable bit '0' for disable '1' for enable
2	SLV	RW	0	Slave mode configuration '0' for master mode '1' for slave mode
1 ~ 0	MD	RW	0	Operation mode "00" for SPI compatible "01" for SSP compatible "1x" reserved for future use

Control Register (CTRL) 0xF0536*n*[24]10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCW	LCR	-	CMDEND					-			CMDSTART				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-			RDSTART					PLW	-		PSW				

Field	Name	RW	Reset	Description
31	LCW	RW	0	Last clock disable for write cycle '1' for enable
30	LCR	RW	0	Last clock disable for read cycle '1' for enable
29	-	-	-	Undefined
28 ~ 24	CMDEND	RW	0	Command end position
20 ~ 16	CMDSTART	RW	0	Command start position
12 ~ 8	RDSTART	RW	0	Read data start position
7	PLW	RW	0	Polarity control for write command
6 ~ 5	-	-	-	Undefined
4 ~ 0	PSW	RW	0	Write command position

Counter & Ext. Event Control Register (EVTCTRL) 0xF0536*n*14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXCRX	TXCREP	EXTEN	EXTDCHK	EXTDPOL											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXCV															

Field	Name	RW	Reset	Description
31	TXCRX	RW	0	Tx Counter Rx Mode
30	TXCREP	RW	0	Tx Counter Repeate Enable
29	EXTEN	RW	0	External Event Enable
28	EXTDCHK	RW	0	Data Check Enable
27	EXTDPOL	RW	0	GSDI Polarity
15 ~ 0	TXCV	RW	0	Tx Counter Load Value

Counter Current Value Register (CCV) 0xF0536*n*18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FSDI															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXC															

Field	Name	RW	Reset	Description
31	FSDI	R	0	Current Serial data Input
15 ~ 0	TXC	R	0	Current Tx Counter Value

[24] n = Channel Number 0 ~ 5

TX Base Register (TX base) **0xF0536*m*[25]20**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TX_BASE [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_BASE [15:0]															

RX Base Register (RX base) **0xF0536*m* 24**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RX_BASE [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_BASE [15:0]															

Packet Register (PACKET) **0xF0536*m* 28**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-			COUNT												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-			SIZE												

Field	Name	RW	Reset	Description
28 ~ 16	COUNT	RW	0	Packet number information (COUNT + 1)
12 ~ 0	SIZE	RW	0	Packet size information

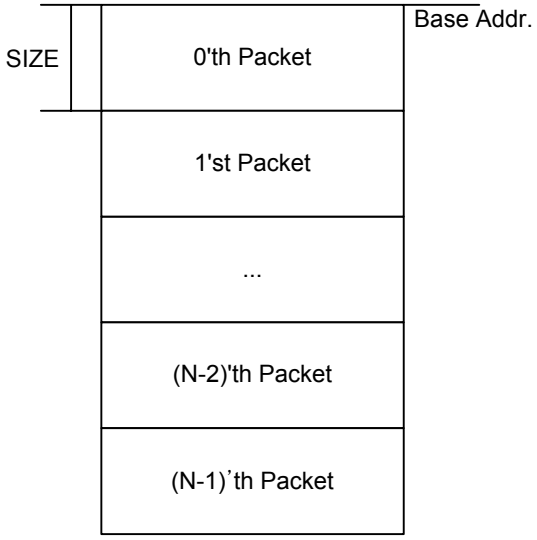


Figure 17.3 Packet Structure

[25] m = Channel Number 0 ~ 2

DMA Control Register (DMA CTRL) 0xF0536*m*[26]2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTE	DRE	CT	END	-								MP	MS	TXAM	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXAM		-								MD		-	PCLR	-	EN

Field	Name	RW	Reset	Description
31	DTE	RW	0	Transmit DMA request enable '1' for enable
30	DRE	RW	0	Receive DMA request enable '1' for enable
29	CT	RW	0	Continuous mode enable (Master) '1' for enable
28	END	RW	0	Byte endian mode register '0' for little-endian '1' for big-endian *If the byte or half-word transfer mode of GPSB mode were used, this field should be '1'.* *The value of '0' is allowed for word transfer only.*
19	MP	RW	0	PID match mode register '1' for enable (MPEG2-TS mode)
18	MS	RW	0	Sync byte match control register '1' for enable (MPEG2-TS mode)
17 ~ 16	TXAM	RW	0	TX addressing mode '0' : Multiple Packet '1' : Fixed address (base) '2', '3' : Single Packet
15 ~ 14	RXAM	RW	0	RX addressing mode '0' : Multiple Packet '1' : Fixed address (base) '2', '3' : Single Packet
5 ~ 4	MD	RW	0	DMA mode register "00" : normal mode "01" : MPEG2-TS mode "1x" : Reserved
2	PCLR	W	0	Clear TX/RX packet counter
0	EN	RW	0	DMA enable register '1' for enable

[26] m = Channel Number 0 ~ 2

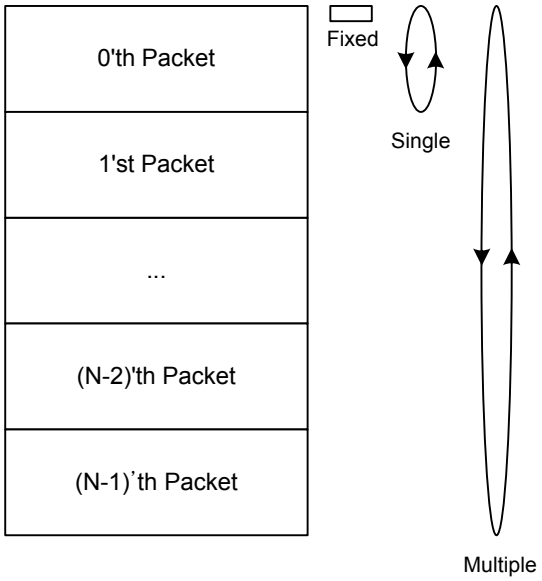


Figure 17.4 TX/RX Addressing Modes

DMA STATUS Register (DMA_STAT) 0xF0536*m*[27]30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		RXPCNT													0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0		TXPCNT													

Field	Name	RW	Reset	Description
29 ~ 17	RXPCNT	R	0	Receive packet count register
12 ~ 0	TXPCNT	R	0	Transmit packet count register

DMA IRQ Register (DMAICR) 0xF0536*m*34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-		ISD	ISP	-							IRQS	-		IED	IEP
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-		IRQPCNT													

Field	Name	RW	Reset	Description
29	ISD	R/C	0	IRQ status for “Done Interrupt” When the number of packets which DMA transfers/receives is equal to the number of packets specified in PACKET register, it is issued. Therefore, it is not valid in continuous mode. When writing 1, it is cleared.
28	ISP	R/C	0	IRQ status for “Packet Interrupt”. It is issued every IRQPCNT packets which DMA transfers/receives. When writing 1, it is cleared.
20	IRQS	R/W	0	IRQ select register ‘0’ for receiving ‘1’ for transmitting
17	IED	R/W	0	IRQ enable for “Done Interrupt”
16	IEP	R/W	0	IRQ enable for “Packet Interrupt”
12 ~ 0	IRQPCNT	R/W	0	IRQ packet count register

[27] m = Channel Number 0 ~ 2

PID Table (PIDT) **0xF0536F00 ~ 0xF0536F80**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH2	CH1	CH0													
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-			PID												

Field	Name	RW	Reset	Description
31	CH2	R/W	X	Channel 2 enable
30	CH1	R/W	X	Channel 1 enable
29	CH0	R/W	X	Channel 0 enable
12 ~ 0	PID	R/W	X	PID value

- Before starting the PID matching, the CH2, CH1 and CH0 fields of all the table entries should be initialized.

Port Configuration Register 0(PCFG0) **0xF0536800**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3								CH2							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CH1								CH0							

Field	Name	RW	Reset	Description
31 ~ 24	CH3	R/W	3	Channel 3 port mapping register
23 ~ 16	CH2	R/W	2	Channel 2 port mapping register
15 ~ 8	CH1	R/W	1	Channel 1 port mapping register
7 ~ 0	CH0	R/W	0	Channel 0 port mapping register

- The port map value for each channel should be different.

Port Configuration Register 1(PCFG1) **0xF0536804**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CH5								CH4							

Field	Name	RW	Reset	Description
15 ~ 8	CH5	R/W	5	Channel 5 port mapping register
7 ~ 0	CH4	R/W	4	Channel 4 port mapping register

- The port map value for each channel should be different.

Channel IRQ Status Register (CIRQST) **0xF0536808**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				0	ISTC5	0	ISTC4	0	ISTC3	ISTD2	ISTC2	ISTD1	ISTC1	ISTD0	ISTC0

Field	Name	RW	Reset	Description
10	ISTC5	R	0	GPSB core IRQ status register for channel 5
8	ISTC4	R	0	GPSB core IRQ status register for channel 4
6	ISTC3	R	0	GPSB core IRQ status register for channel 3
5	ISTD2	R	0	GPSB DMA IRQ status register for channel 2
4	ISTC2	R	0	GPSB core IRQ status register for channel 2
3	ISTD1	R	0	GPSB DMA IRQ status register for channel 1
2	ISTC1	R	0	GPSB core IRQ status register for channel 1
1	ISTD0	R	0	GPSB DMA IRQ status register for channel 0

0	ISTC0	R	0	GPSB core IRQ status register for channel 0

** If IRQ status of each channel were cleared, the corresponding field would be read '0'.*
** The IRQ mode of GPSB channel is preferred to "level-trigger" mode*.

17.4 GPSB Timing Diagram

The table and figures on the following page are the examples of GPSB MODE values to be programmed for SPI and SSP interface.

Timing	CPOL/CPHA	Master	Slave
		PWD,PRD,PCK	PCK
SPI timing 0	00	000	0
SPI timing 1	01	110	1
SPI timing 2	10	001	1
SPI timing 3	11	111	0
SSP timing	NA	110	1

The figures below show the timing of each case listed in the table DIVLDV = 0 (GCLK / 2), Single Transfer Mode assumed

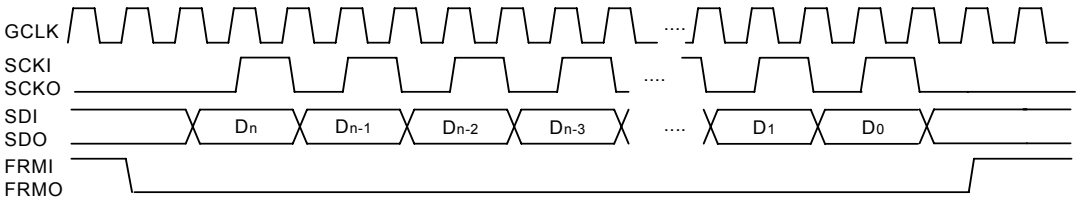


Figure 17.5 SPI Timing 0

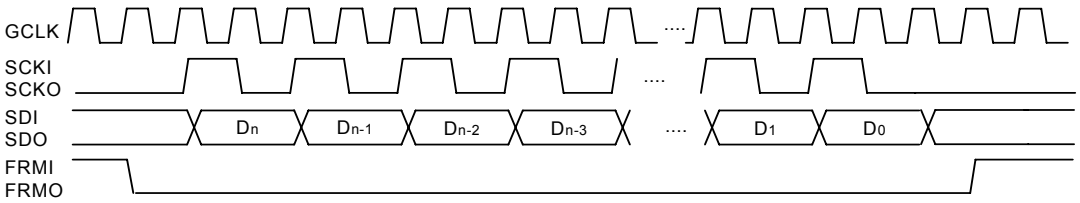


Figure 17.6 SPI Timing 1

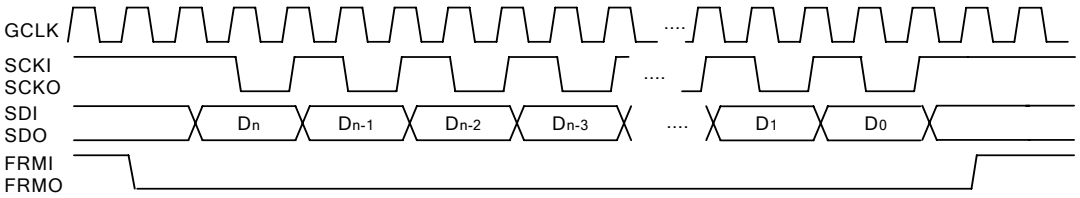


Figure 17.7 SPI Timing 2

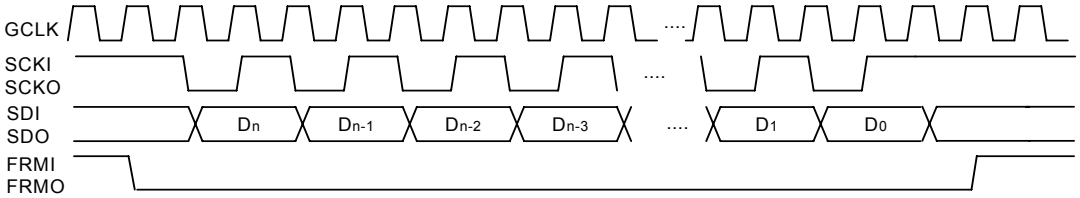


Figure 17.8 SPI Timing 3

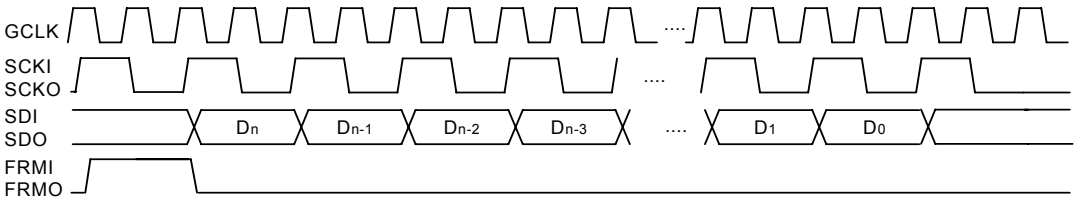


Figure 17.9 SSP Timing

17.5 MPEG2-TS Interface

The figures on the following page are the examples of MPEG2-TS bit stream. The packet of MPEG2-TS (Transport Stream) is composed of two groups that are the 188-bytes Payload data and 16bit Parity data. The Payload data including TS header within Sync byte is shown below Figure 17.10.

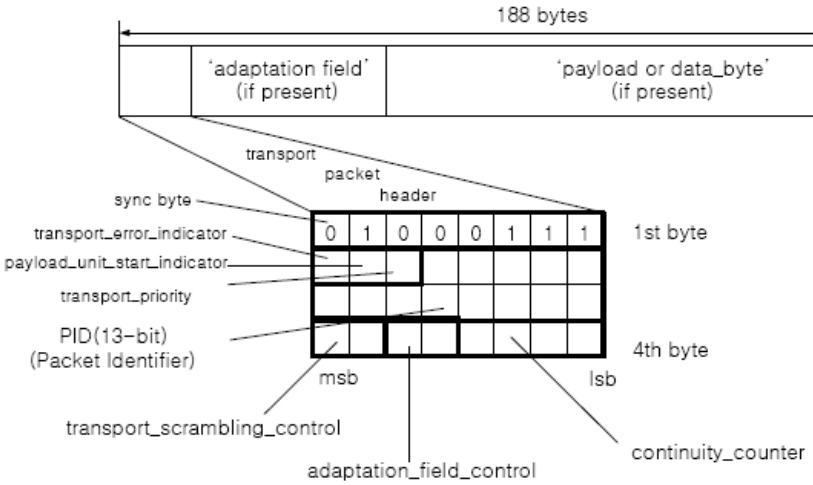


Figure 17.10 Bitstream of MPEG2-TS

In the Figure 17.10, the first byte of TS packet header is '0x47'. This sync byte arranges the serial stream data that is processed, stored, and read.

The input of MPEG2-TS interface is TS_CLK and TS packet data. TS packet data is composed in 204-byte unit and is separated by SYNC signal. Figure 17.11 shows the TS timing information.

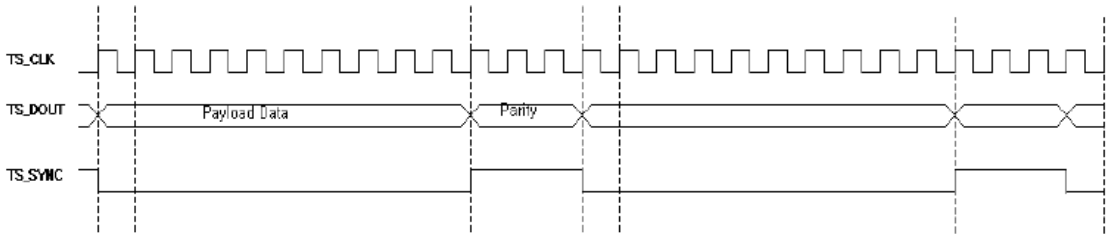


Figure 17.11 MPEG2-TS timing information

18 TSIF(The transport Stream Interface)

18.1 Overview

The TCC8900 includes two independent transport stream interface(TSIF) modules. The TSIF is able to receive 2 sets of parallel/serial input stream at once, supplied from overall 4 set of input ports.

The block diagram of TSIF is shown in Figure 18.1.

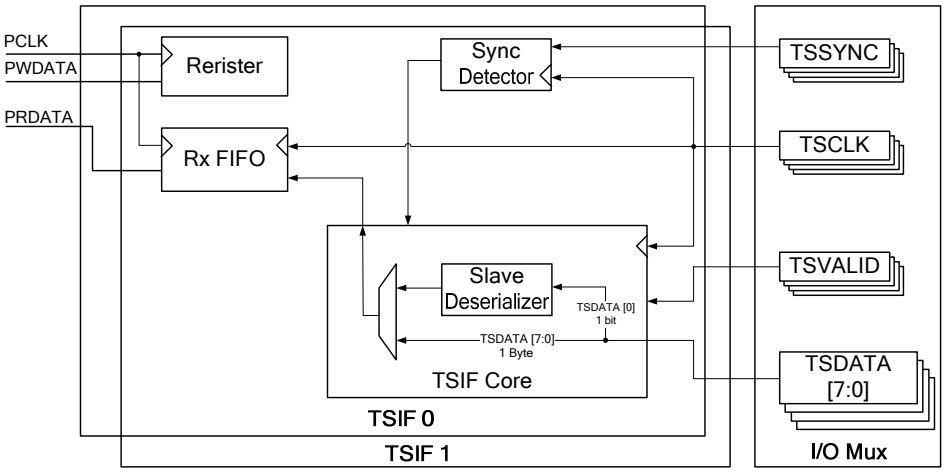


Figure 18.1 TSIF Block Diagram

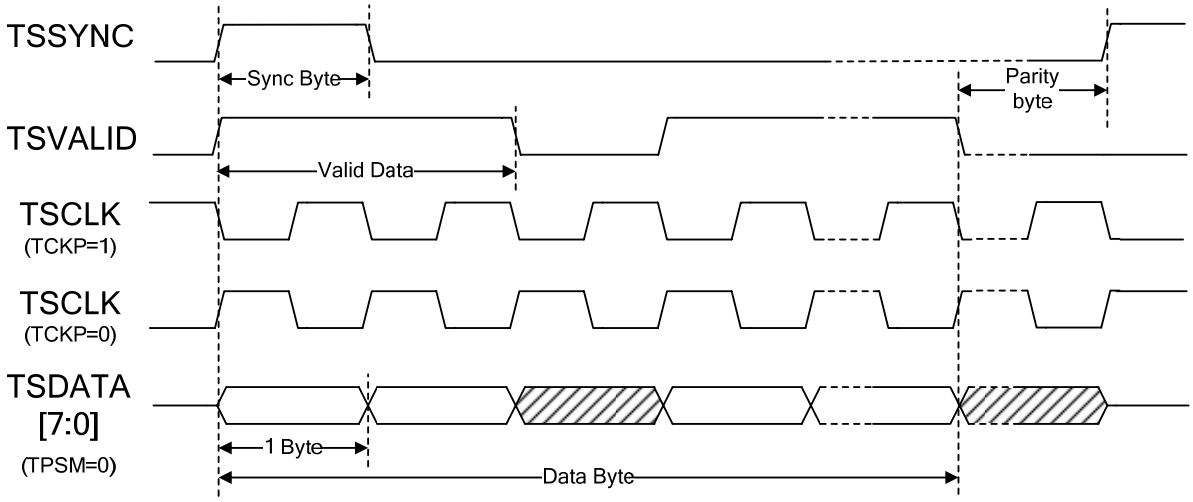


Figure 18.2 Timing of TS Data Input (Parallel Mode)

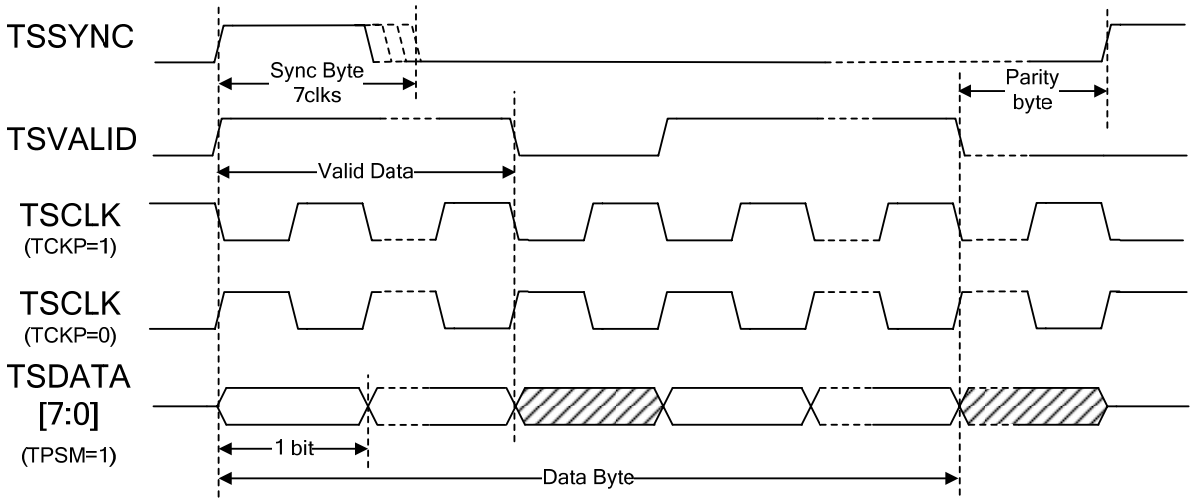


Figure 18.3 Timing of TS Data Input (Serial Mode)

Table 18.1 GPIO Port Map

GPIO_C	GPIO_D	GPIO_E	GPIO_F	I/O	TSIF
GPIO_C[0]	GPIO_D[9]	GPIO_E[19]	GPIO_F[7]	I	TSDATA [7]
GPIO_C[1]	GPIO_D[10]	GPIO_E[18]	GPIO_F[6]	I	TSDATA [6]
GPIO_C[2]	GPIO_D[13]	GPIO_E[17]	GPIO_F[5]	I	TSDATA [5]
GPIO_C[24]	GPIO_D[14]	GPIO_E[16]	GPIO_F[4]	I	TSDATA [4]
GPIO_C[25]	GPIO_D[17]	GPIO_E[15]	GPIO_F[3]	I	TSDATA [3]
GPIO_C[26]	GPIO_D[18]	GPIO_E[14]	GPIO_F[2]	I	TSDATA [2]
GPIO_C[27]	GPIO_D[19]	GPIO_E[13]	GPIO_F[1]	I	TSDATA [1]
GPIO_C[30]	GPIO_D[20]	GPIO_E[12]	GPIO_F[0]	I	TSDATA [0]
GPIO_C[28]	GPIO_D[15]	GPIO_E[22]	GPIO_F[8]	I	TSVALID
GPIO_C[29]	GPIO_D[16]	GPIO_E[20]	GPIO_F[9]	I	TSCLK
GPIO_C[31]	GPIO_D[12]	GPIO_E[21]	GPIO_F[10]	I	TSSYNC

18.2 Register Description

Table 18.2 TSIF Register Map (Base Address = 0xF053B000)

Name	Address	Type	Reset	Description
TSDI	0x00	R	0x0000	TSIF Input Data Register
TSCR	0x04	R/W	0x0000	TSIF Control Register
TSPID	0x08	R/W	0x0000	TSIF PID Register
TSCTRL	0x10	R/W	0x0000	TSIF Interrupt Control Register
TSSTS	0x10(14)	R	0x0000	TSIF Interrupt Status Register(Test)
TSCHS	0x800	R/W	0x0000	TSIF Channel(Port) Select Register

* TSIF Ch_1 Address = TSIF Ch_0 + 0x100

TSIF Input Data Register (TSDI) **0xF053B000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSDI[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TSDI[15:0]															

TSDI[31:0]	TSDI Input Data
N	This is a Rx FIFO port. It has a 32bit 8depth FIFO.

TSIF Control Register (TSCR) **0xF053B004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN	Reserved													SBE	M/L
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
END	TPSM	TCKP	TVEP	TSP	TrgPos	TBS		CRF	TVSC	TSDE	TSD				

EN[31]	TSIF Enable
0	Disable TSIF
1	Enable TSIF. EN bit has to be set after other bits are already set. If you want to change control bits, you have to set these after EN is cleared.

SBE[17]	Sync Byte Enable
0	Disable
1	Enable

When the SBE is set to high, sync byte input value is compared with 0x47 and if it differs, the input data is not saved until next sync byte occurs.

M/L[16]	MSB/LSB
0	Send LSB firstly
1	Send MSB firstly

END[15]	Select Endian
0	Little Eian transfer mode
1	Big Endian transfer mode

TPSM[14]	TSIF Data Mode
0	TSIF Parallel Mode (TSDATA [7:0])
1	TSIF Serial Model (TSDATA [0])

TCKP[13]	TSCLK Polarity
0	Starts transfer at the rising edge of TSCLK after TSSYNC is low
1	Starts transfer at the falling edge of TSCLK after TSSYNC is low

TVEP[12]	TSIF Valid Data Enable Polarity
0	TSVALID has low active pulse
1	TSVALID has high active pulse

TSP[11]	TSIF Sync Pulse Polarity
0	TSSYNC has low active pulse

1	TSSYNC has high active pulse

TrgPos[10]	Trigger Position
0	Regard the head of frame signal as a start point
1	Regard the tail of frame signal as a start point

TBS[9:8]	TSIF Byte Size
00	1 Byte FIFO Save (Only Serial)
01	2 Byte FIFO Save
11	4 Byte FIFO Save

CRF[7]	Clear Rx FIFO
0	Don't empty Rx FIFO
1	Empties Rx FIFO

TVSC[6]	TSIF Valid Port Change
0	Disable
1	TSIF Valid port → TSIF Sync port(msm chip)

If TSSYNC is not supplied, it can be replaced by TSVALID. In this case, only 3 signals(TSCLK, TSVALID, TSDATA) are used.

TSDE[5]	TSIF Sync Delay Enable
0	Disable
1	Enable (msm chip)

TSD[4:0]	TSIF Sync Delay
N	Sync for msm chip (1~32 Delay)

TSIF PID Register (TSPID) **0xF053B008**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		PIDE	PIDV												

PIDE[13]	Packet ID Enable
0	Disable
1	Enable

PIDV[12:0]	Packet ID Value
N	Value (0~8191)

The PID of new input data can be compared with PIDV. If it matches, it can set the pending bit of interrupt control register.

TSIF Interrupt Control Register (TSCTRL) **0xF053B00C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RxCnt			RCFul	REmp	RFul	ROR	RUR

These enable bits are used to monitor through Interrupt Status Register or make interrupt trigger or DMA request. It is recommended for you to set only what you want.

PID[9]	Packet ID Matching
1/0	Packet ID matching interrupt

RxCnt[7:5]	Rx FIFO Threshold Level
0~7	Sets threshold level of Rx FIFO to notify how many data it is filled with. It designates N depth data is available at Rx FIFO. Exceptionally, '0' stands for full state.

RCFul[4]	Rx FIFO Count Full Interrupt Enable
1/0	Enables / Disables Rx FIFO count full interrupt

REmp[3]	Rx FIFO Empty Interrupt Enable
1/0	Enables / Disables Rx FIFO empty interrupt

RFul[2]	Rx FIFO Full Interrupt Enable
1/0	Enables / Disables Rx FIFO full interrupt

ROR[1]	Packet ID Enable Interrupt Enable
1/0	Enables / Disables Rx FIFO overrun error interrupt

RUR[1]	Packet ID Enable Interrupt Enable
1/0	Enables / Disables Rx FIFO underrun error interrupt

TSIF Interrupt Status Register (TSSTS) **0xF053B010(14)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						PID	R*	Reserved			RCFul	REmp	RFul	ROR	RUR

* After you read this status register, it will be cleared

PID[9]	Packet ID Matching
1/0	Packet ID matching status flag (read only)

RCFul[4]	Rx FIFO Count Full
1/0	Rx FIFO counter full status flag (read only)

REmp[3]	Rx FIFO Empty
1/0	Rx FIFO empty status flag (read only)

RFul[2]	Rx FIFO Full
1/0	Rx FIFO full status flag (read only)

ROR[1]	Packet ID Enable
1/0	Rx FIFO overrun error status flag (read only)

RUR[1]	Packet ID Enable
1/0	Rx FIFO underrun error status flag (read only)

TSIF Channel Status Register (TSCHS) **0xF053B800**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TSIF0		TSIF1	

TSIF0[3:2]	TSIF0 Port Select
00	GPIO_F Select
01	GPIO_E Select
10	GPIO_D Select
11	GPIO_C Select

TSIF1[1:0]	TSIF1 Port Select
00	GPIO_F Select
01	GPIO_E Select
10	GPIO_D Select
11	GPIO_C Select

19 Remote Control Interface

19.1 Functional Description

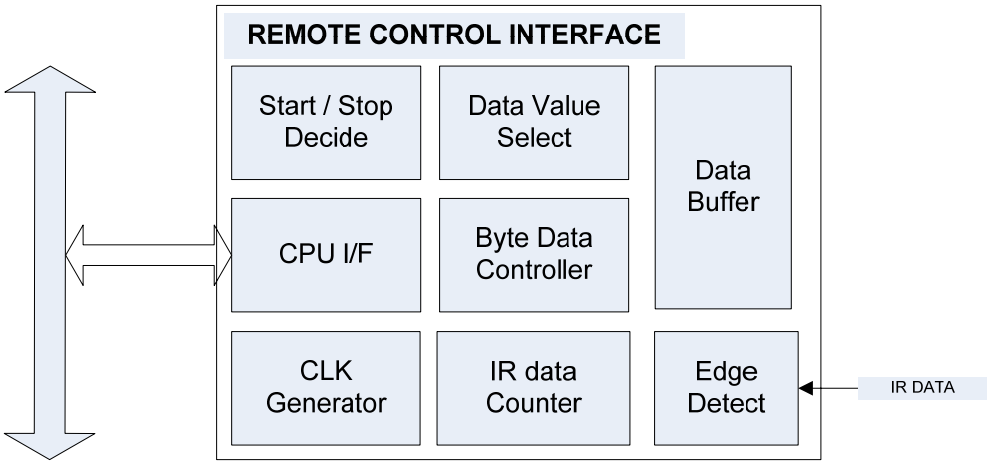


Figure 19.1 Remote Control Interface Block Diagram

The Remote Control Interface stores raw input samples of Infra-Red signal, generated from IR transmitter such as NOKIA, SONY, PHILLIPS, JVC, SHARP, etc. It does not support digital decoding like PWM, NRZ, NRZI but just counts runs of high and low level from IR input

Figure 19.2 shows the example of received IR data.

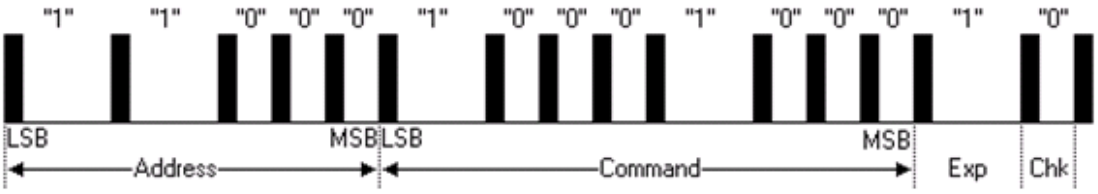


Figure 19.2 IR data input example

When the rising edge of IR data is detected, high level run counting is started. If IR input level changes to low, it stops counting and stores the count number into the FIFO. And at the same time, low level counting is started.

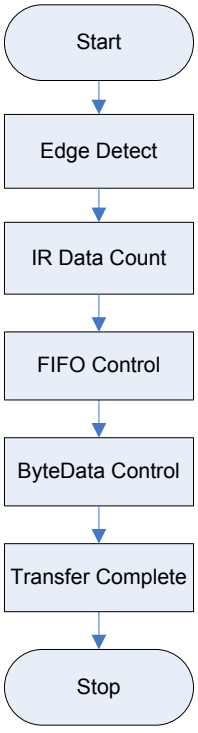


Figure 19.3 The overall flow of IR data capture

The high/low level run counts are written to internal FIFO in the order, shown in Figure 19.4.

Second IR data low dur_count value	Second IR data high dur_count value	First IR data low dur_cont value	First IR data high dur_count value
8bit	8bit	8bit	8bit

Figure 19.4 Data write format in FIFO

Note: The dur_count is 8-bit counter(ranges 0~255) and four count numbers are written at one time in the FIFO.

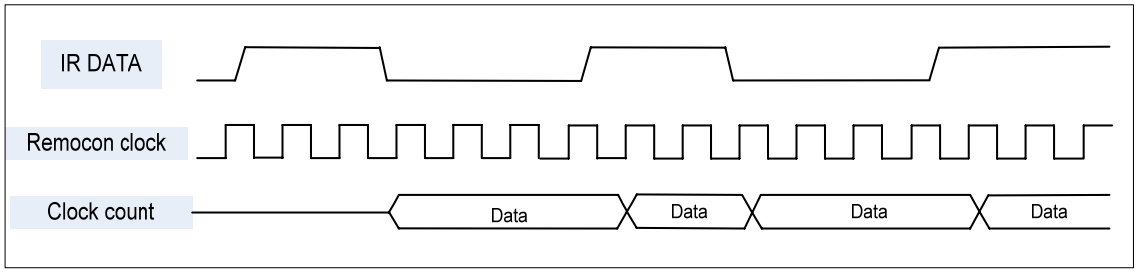


Figure 19.5 The timing diagram about internal counter clock and dur_count value

The sampling clock frequency can be changed according to clock divider register and be adjusted so that the count number does not exceed the counter range.

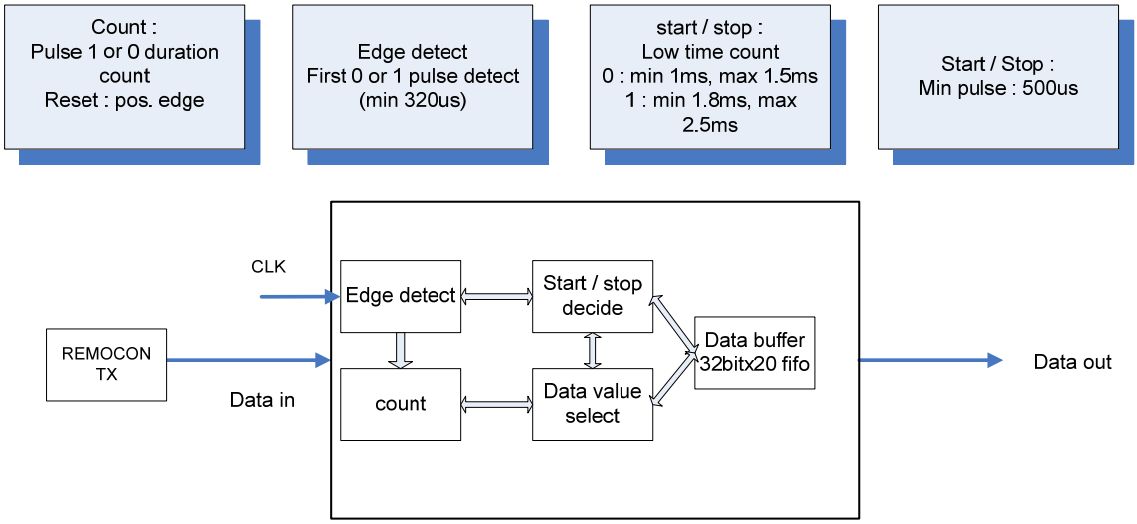


Figure 19.6 The minimum or maximum pulse width of IR signal

Figure 19.6 shows the minimum or maximum pulse width of IR signal required.

19.2 Register Description

Table 19.1 Remocon Register Map (Base Address = 0xF05F3000)

Name	Addr. Offset	Type	Reset	Description
RDATA	0x00	RO	0x0000	IR read Data
CMD	0x04	R/W	0x0000	Command Register
INPOL	0x08	R/W	0x0000	Input Polarity Inversion Register
STA	0x0C	W	0x0000	Status Register
CLKDIV	0x40	R	0x0000	Clock Divide Register

IR Read Data Register (RDATA) 0xF05F3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDATA															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA															

When remocon interrupt arises, host processor can repeatedly reads this register only 18-times(the maximum number of IR signal bit is restricted to 36), and analyzes all IR input. Also the initialization of read FIFO would be taken like this.

Command Register (CMD) 0xF05F3004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FWEN	DEN	WS	TH									CLEAR	EN	FF

FF [0]	Interrupt Acknowledge
0	Disabled
1	Enabled

EN [1]	Remocon Core enable bit
0	Disabled
1	Enabled

CLEAR [2]	Duration Count Clear
0	Disabled
1	Enabled

TH [3~11]	FIFO Threshold
Value	Threshold

WS [12]	IR CLK Wait Control
0	Disabled
1	Enabled

DEN [13]	Duration Count Enable
0	Disabled
1	Enabled

FWEN [14]	FIFO Write Enable
0	Disabled
1	Enabled

Input Polarity Inversion Register (INPOL) 0xF05F3008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														INV	

INV[1]	Input Polarity Inversion
0	Disabled
1	Enabled

Status Register (STA) 0xF05F300C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			OF	ICF											

ICF [0~11]	Interrupt Configure
0	Disabled
1	Enabled

OF [12]	FIFO Over Flow
0	Disabled
1	Enabled

CLK DIVIDER Register (CLKDIV) 0xF05F3040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLK_DIV															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								END_CNT							

END_CNT [0~7]	Transfer end Zero Count
0	Disabled
1	Enabled

CLK_DIV [14~31]	Clock Divider
0	Disabled
1	Enabled

20 I2C Controller

20.1 Functional Description

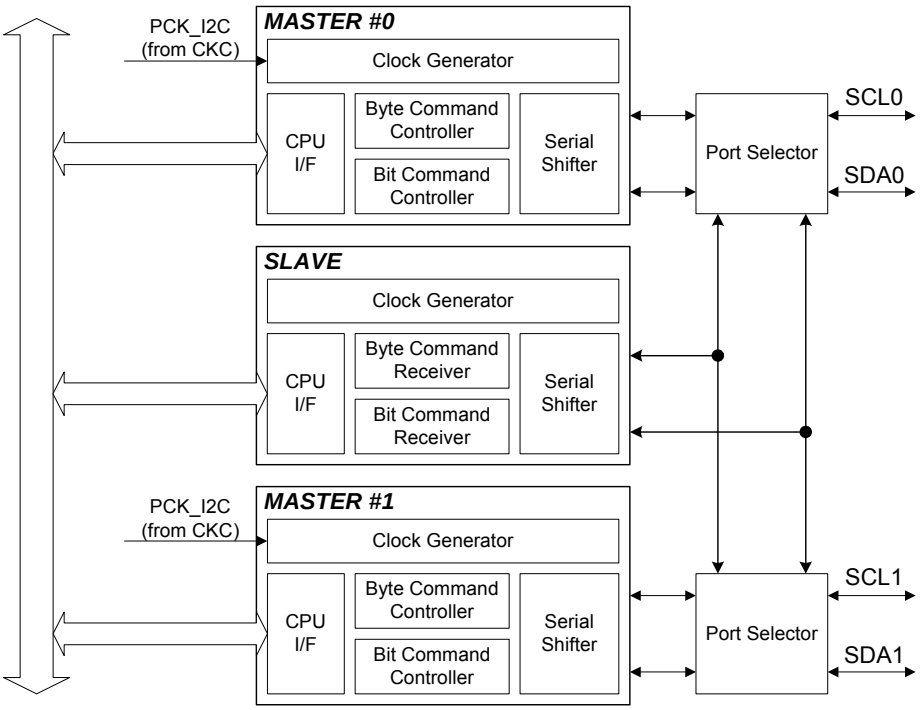


Figure 20.1 I2C Block Diagram

The I2C controller in the TCC8900 has two masters and one slave controller. The each master can control the dedicated I2C bus and the slave controller can select the one of two I2C buses. For each I2C bus can be selected by the register.

20.2 I2C Slave Function

- Supports Fast mode (400Kbps)
- Supports wait state by clock stretching
- 4byte TX FIFO, 4byte RX FIFO and 8byte Buffer

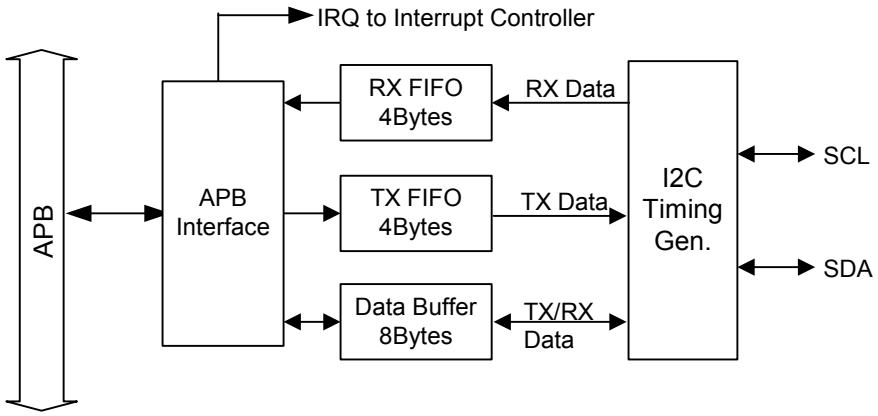
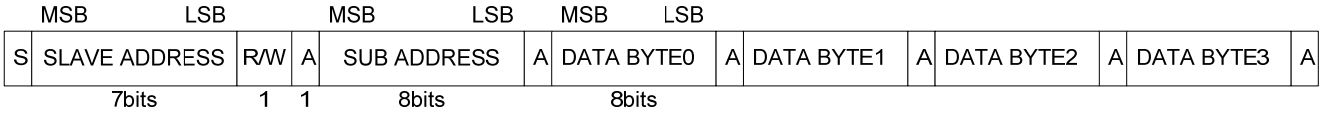


Figure 20.2 I2C Slave Block Diagram

The following figure and table show an example format for accessing Buffers and FIFOs in the I2C slave core.



S: Start Condition
R/W: Read (1) or Write(0)
A: Acknowledge from/to the core

SUB ADDRESS[7:0]	DATA[7:0] Source / Destination	Note
0x00 ~ 0x03	MB0 (Data Buffer 0) Byte 0 ~ 3	
0x04 ~ 0x07	MB1 (Data Buffer 1) Byte 0 ~ 3	
0x08 ~ 0x7F	Reserved	NACK is sent to a master
0x80 ~ 0xFF	RX/TX FIFO	All addresses are directed to the RX/TX FIFO

20.3 Related Blocks

After the signals are enabled, PCK_I2C (the main clock of I2C) must be enabled and configured to the proper frequency.

For internal synchronization, the APB clock frequency must be faster than the PCK_I2C frequency.

$f_{PCK\_I2C} \leq f_{HCLK} / 4.0$

20.4 Register Description

Table 20.1 I2C Register Map (Base Address = 0xF0530000)

Ch	Name	Addr. Offset	Type	Reset	Description
Master 0	PRES	0x00	R/W	0xFFFF	Clock Prescale register
	CTRL	0x04	R/W	0x0000	Control Register
	TXR	0x08	W	0x0000	Transmit Register
	CMD	0x0C	W	0x0000	Command Register
	RXR	0x10	R	0x0000	Receive Register
	SR	0x14	R	0x0000	Status Register
	TIME	0x18	R/W	0x0000	Timing Control Register
Master 1	PRES	0x40	R/W	0xFFFF	Clock Prescale register
	CTRL	0x44	R/W	0x0000	Control Register
	TXR	0x48	W	0x0000	Transmit Register
	CMD	0x4C	W	0x0000	Command Register
	RXR	0x50	R	0x0000	Receive Register
	SR	0x54	R	0x0000	Status Register
	TIME	0x58	R/W	0x0000	Timing Control Register
Slave	PORT	0x80	R/W	-	Data Access port (TX/RX FIFO)
	CTL	0x84	R/W	0x00000000	Control register
	ADDR	0x88	W	0x00000000	Address register
	INT	0x8C	W	0x00000000	Interrupt Enable Register
	STAT	0x90	R	0x00000000	Status Register
	MBF	0x9C	R/W	0x00000000	Buffer Valid Flag
	MB0	0xA0	R/W	0x00000000	Data Buffer 0 (Byte 3 ~ 0)
	MB1	0xA4	R/W	0x00000000	Data Buffer 1 (Byte 7 ~ 4)
Status	IRQSTR	0xC0	R	0x00000000	IRQ Status Register

Prescale Register (PRES) 0xF0530000, 0xF0530040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Clock Prescale data															

This register is used to prescale the SCL clock line. Due to the structure of the I2C interface, the core uses a 5*SCL clock internally. The prescale register must be programmed to this 5*SCL frequency (minus 1). Change the value of the prescale register only when 'EN' bit is cleared.

Example :

CLK Input frequency = 8MHz , Desired SCL frequency = 100KHz

Prescale = (8MHz / 100KHz) – 1 = 15

Control Register (CTR) 0xF0530004, 0xF0530044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								EN	IEN	MOD	RESERVED				

EN [7]	I2C Core enable bit
0	Disabled
1	Enabled

IEN [6]	I2C Core interrupt enable bit
0	Disabled
1	Enabled

MOD [5]	I2C Data Width
0	8bit Mode
1	16bit Mode

Transmit Register (TXR) 0xF0530008, 0xF0530048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Transmit Data															

When CTRL[5] is set, in case of 16Bit Mode is selected, Transmit Data bit width become 16 bit. Default mode is 8bit mode.

Command Register (CMD) **0xF053000C, 0xF053004C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								STA	STO	RD	WR	ACK	RESERVE		IACK

STA [7]	Start Condition Generation
0	Disabled
1	Enabled

STO [6]	Stop Condition Generation.
0	Disabled
1	Enabled

RD [5]	Read From Slave
0	Disabled
1	Enabled

WR [4]	Write to Slave
0	Disabled
1	Enabled

ACK [3]	Sent ACK
0	Enabled
1	Disabled

IACK [0]	Interrupt Acknowledge
0	-
1	Clear a pending interrupt

Receive Register (RXR) **0xF0530010, 0xF0530050**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Receive Data															

When CTRL[5] is set, in case of 16Bit Mode is selected, Transmit Data bit width become 16 bit. Default mode is 8bit mode.

Status Register (SR) 0xF0530014, 0xF0530054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								RxACK	BUSY	AL				TIP	IF

RxACK [7]	Received acknowledge from slave
0	Acknowledge received
1	No Acknowledge received

BUSY [6]	I2C Bus Busy
0	'0' after STOP signal detected
1	'1' after START signal detected

AL[5]	Arbitration lost
0	The core does not lose arbitration
1	The core loses arbitration

Arbitration is lost when a STOP signal is detected, but non requested master drives SDA high, but SDA is low

TIP [1]	Transfer in progress
0	Transfer Complete
1	Transferring Data

IF [0]	Interrupt Flag
0	-
1	Interrupt is pending

Timing Register (TR) 0xF0530018, 0xF0530058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RC								-		CKSEL	FC				

RC [15:8]	Recovery time counter load value
value	"0" disables recovery time counter. The recovery time counter is enabled and loaded with RC[7:0] whenever a STOP condition is issued by the core. Execution of a new command written to CMD register is delayed until the counter is expired. Actual wait time = (I2CCLK period) * PRES[15:0] * 5 * RC[7:0]

CKSEL [5]	Clock Source Select
0	I2CCLK from Clock controller
1	PCLK (HCLK) divided by 2

Recommended if PCLK is not variable during system operation.

FC[4:0]	Noise filter counter load value
value	"0" disables noise filter. SCL and SDA inputs are checked for stability until the counter is expired.

Data Port (DPORT) 0xF0530080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-					TXVC			-					RXVC		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-								PORT							

TXVC [26:24]	TX buffer valid entry count
value	TX buffer valid entry count

RXVC [18:16]	RX buffer valid entry count
value	RX buffer valid entry count

PORT[7:0]	TX/RX FIFO access port
value	TX / RX FIFO access port

Master Write Cycle to RX FIFO

	MSB LSB			MSB LSB		MSB LSB		MSB LSB		MSB LSB		MSB LSB	
S	WRITE SLAVE ADDRESS	W(0)	A	WRITE 1XXXXXXXb	A	WRITE DATA BYTE0	A	WRITE DATA BYTE1	A	WRITE DATA BYTE2	A	WRITE DATA BYTE3	A
	7bits	1	1	8bits		8bits							

Master Read Cycle to TX FIFO

	MSB LSB			MSB LSB		MSB LSB		MSB LSB		MSB LSB		MSB LSB	
S	WRITE SLAVE ADDRESS	R(1)	A	always read as 11111111b	A	READ DATA BYTE0	A	READ DATA BYTE1	A	READ DATA BYTE2	A	READ DATA BYTE3	A
	7bits	1	1	8bits		8bits							

The first byte is always read as 0xFF. The first byte must be discarded by the master

Control Register (CTL) 0xF0530084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV		-					FC					DRQEN			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-		TXTH		-		RXTH		--		RCLR	WS	SDA	CLR	-	EN

SLV [31:30]	Pin Configuration for Slave Core
00/11	2 master operation I2C Channel 0 : Master 0 I2C Channel 1 : Master 1
01	1 master / 1 slave I2C Channel 0 : Slave I2C Channel 1 : Master 1
10	1 master / 1 slave I2C Channel 0 : Master 0 I2C Channel 1 : Slave

.

FC[24:20]	Filter Counter Load Value
value	"0" disables noise filter. SCL and SDA inputs are checked for stability until the counter is expired.

DRQEN [19:16]	DMA Request Enable
value	DMA request enable

TXTH [13:12]	TX FIFO threshold for DMA Request
value	TX FIFO threshold value

.

RXTH [9:8]	RX FIFO threshold for DMA Request
value	RX FIFO threshold value

RCLR [5]	Clear Interrupt Status at read cycle
0	-
1	Clear

WS [4]	Wait Status Control by SCL Stretching
0	Enable
1	Disable

SDA [3]	Reserved for test
0	-
1	-

CLR [2]	Clear FIFO
0	-
1	Clear

EN [0]	Enable for this slave core
0	Disable
1	Enable

Address Register (ADDR) **0xF0530088**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								ADDR							-

ADDR [7:1]	Device Address
value	Slave address

Interrupt Register (INT) **0xF053008C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-				IRQSTAT											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-				IRQEN											

IRQST[bit]	Interrupt Status
27th bit	All bytes of data buffer have been read by a master
26th bit	All bytes of data buffer have been written by a master
25th bit	Data buffer has been read by a master
24th bit	Data buffer has been written by a master
23rd bit	TX FIFO under run
22nd bit	RX FIFO over run
21st bit	TX bus cycle started with TX FIFO empty
20th bit	RX FIFO full
19th bit	TX FIFO empty
18th bit	RX FIFO not empty
17th bit	TX FIFO Level (TXVC <= TXTH)
16th bit	RX FIFO Level (RXVC <= RXTH)

.

IRQEN[bit]	Interrupt Enable
11th bit	All byte of data buffer has been read by a master
10th bit	All byte of data buffer has been written by a master
9th bit	Data buffer has been read by a master
8th bit	Data buffer has been written by a master
7th bit	TX FIFO under run
6th bit	RX FIFO over run
5th bit	TX bus cycle started with TX FIFO empty
4th bit	RX FIFO full
3rd bit	TX FIFO empty
2nd bit	RX FIFO not empty
1st bit	TX FIFO Level (TXVC <= TXTH)
0th bit	RX FIFO Level (RXVC <= RXTH)

Note: When RCLR bit of Control Register is “1”, IRQSTAT bits are cleared at the end of read cycle.

When RCLR bit of Control Register is “0”, IRQSTAT bits are cleared by writing “1”.

Status Register (STAT) **0xF0530090**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-								DDIR	-						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-								SADR							

DDIR [3]	Data Direction
0	RX
1	TX

SADR[7:0]	Slave address received
value	Slave address received from address cycle.

Buffer Valid Flag Register (MBF) **0xF053009C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-								MBFT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-								MBFR							

MBFT [23:16]	Buffer TX Flag
value	Buffer TX flag for byte 7 ~ 0. When the data buffer 0/1 is read by an external master, the corresponding bit is set. MBFT[3:0] is cleared when the CPU writes to MB0 and MBFT[7:4] is cleared when MB1 is written by the CPU

MBFR [7:0]	Buffer RX Flag
value	Buffer RX flag for byte 7 ~ 0. When the data buffer 0/1 is read by an external master, the corresponding bit is set. MBFT[3:0] is cleared when the CPU writes to MB0 and MBFT[7:4] is cleared when MB1 is written by the CPU

Data buffer 0(MB0) **0xF05300A0**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Data byte 3								Data byte 2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data byte 1								Data byte 0							

Data buffer 1(MB1) **0xF05300A4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Data byte 7								Data byte 6							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data byte 5								Data byte 4							

NOTE: Deviation from the standard.

The MB0 and MB1 registers can not be read with the standard I2C read timing sequence. Always TX FIFO is accessed with the standard read timing. To read MB0 or MB1, the sequence below must be followed. If a master is not capable of this non-standard sequence, MB0 and MB1 can not be read.

	MSB LSB			MSB LSB		MSB LSB									
S	WRITE SLAVE ADDRESS	R(1)	A	WRITE 00000XXXb	A	READ DATA BYTE0	A	READ DATA BYTE1	A	READ DATA BYTE2	A	READ DATA BYTE3	A		
	7bits	1	1	8bits		8bits									

IRQSTR **0xF05300C0**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0													ST2	ST1	ST0

ST2	IRQ Status of I2C Slave Controller
0/1	IRQ Status Flag for I2C Slave Controller This would be read as '1' when the IRQ status of I2C slave controller is activated.

ST1	IRQ Status of I2C Master Controller Channel 1
0/1	IRQ Status Flag for I2C Master Controller of Channel 1. This would be read as '1' when the IRQ status of I2C master controller channel 1 is activated.

ST0	IRQ Status of I2C Master Controller Channel 0
0/1	IRQ Status Flag for I2C Master Controller of Channel 0. This would be read as '1' when the IRQ status of I2C master controller channel 0 is activated.

21 UART

21.1 Overview

The TCC8900 has the 6-channel UART that can be used in programming the system software, IrDA interfacing or high speed serial communication. Additionally, these channels can be used as the smart card interface.

The block diagram of UART is shown in Figure 21.1.

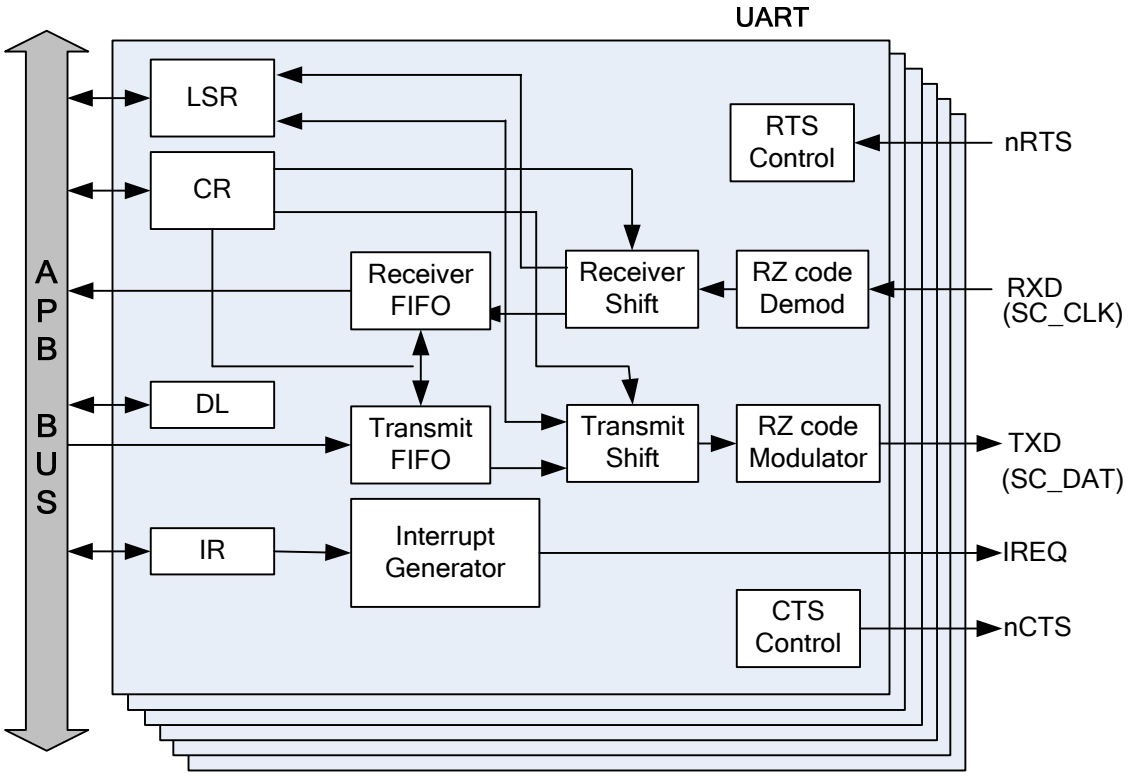
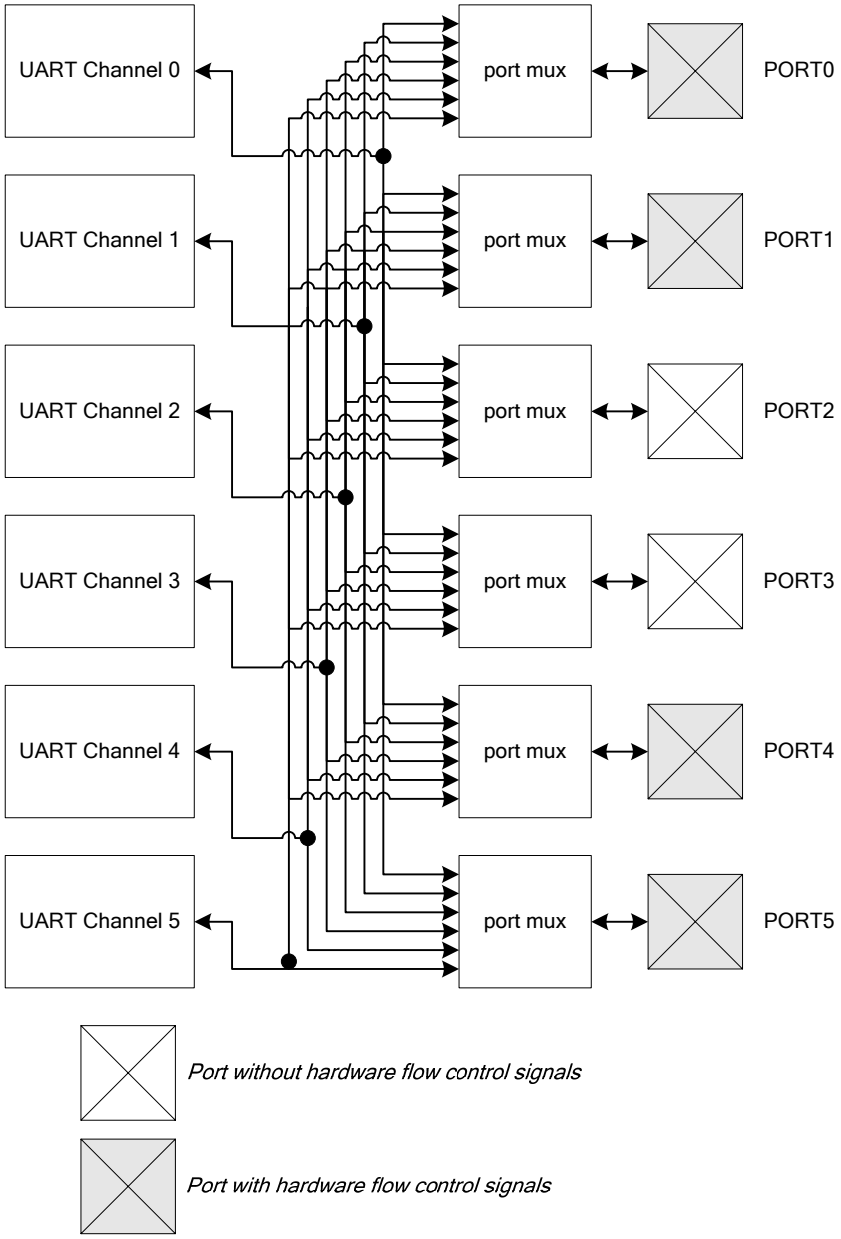


Figure 21.1 UART Block Diagram

Each channel can be connected to the one of 6 UART ports. UART port 0, port 1, port 4 and port 5 have flow-control signals, but UART port 2, and port 3 does not have them. Although all channels have the hardware flow control function, channels should be mapped to UART port 0, port 1, port 4, or port 5 to use the hardware flow control function.

For smart card interface, RXD signal is used as the smart card clock (SC_CLK) and TXD signal is used as the smart card data (SC_DAT). All UART ports and channels are available for smart card interface.


UART Channel 0
UART Channel 1
UART Channel 2
UART Channel 3
UART Channel 4
UART Channel 5
port mux
port mux
port mux
port mux
port mux
port mux
PORT0
PORT1
PORT2
PORT3
PORT4
PORT5
Port without hardware flow control signals
Port with hardware flow control signals

21.2 Operation Modes

21.2.1 AFC (Auto Flow Control) in RX Operation

Register Set Value
AFE (MCR[4]): Auto Flow Control Enable Bit = 1
DTL (AFT[3:0]) : nRTS Deassert Trigger Level = 15
ATL (AFT[7:4]) : nRTS Assert Trigger Level = 15
RXT (FCR[7:6]) : Rx Available FiFo Trigger Level = 0

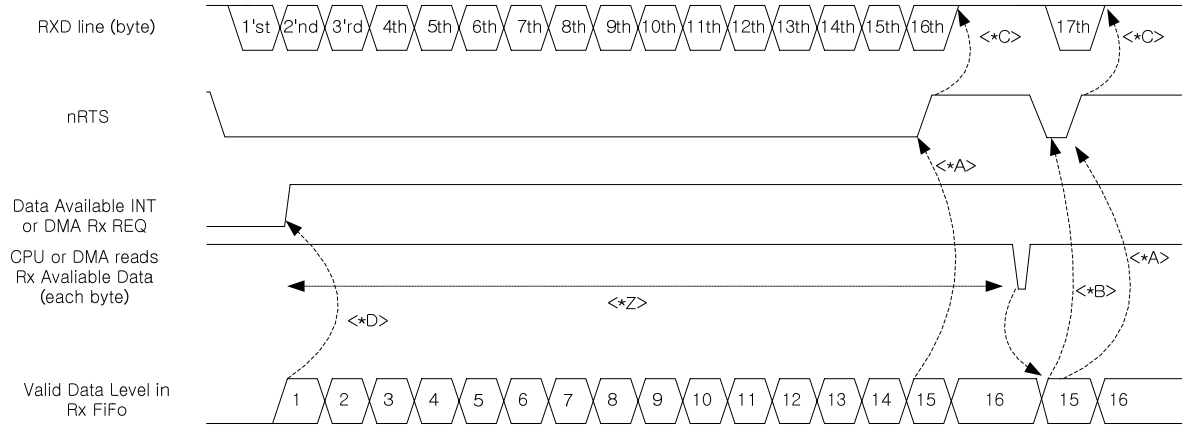


Figure 21.2 nRTS operation - Case: CPU/DMA is slow

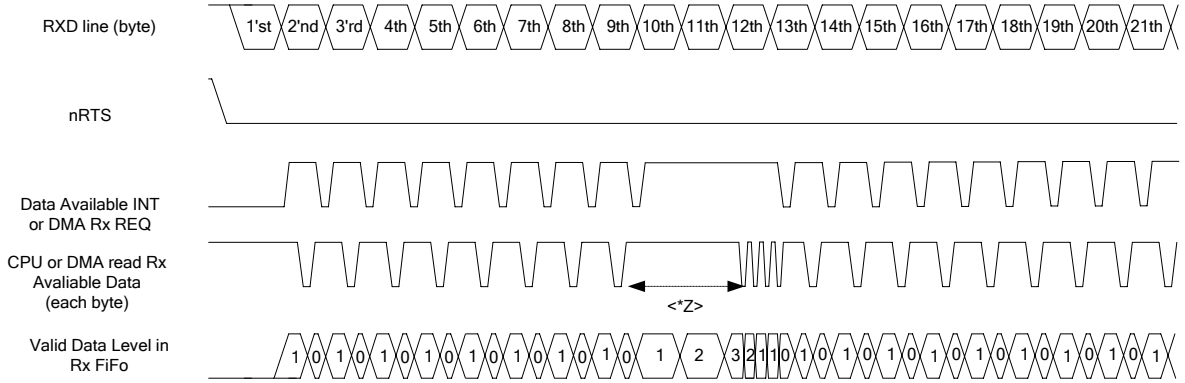


Figure 21.3 nRTS operation - Case: CPU/DMA is fast

<*Z>: When CPU can not read Rx available data due to other jobs or when DMA access can not be granted.
<*A>: Valida data is 15EA (AFT[3:0] = 15. According to RS(MCR[6]) value, nRTS is deasserted (nRTS = HIGH) whether under RXD Start Condition (RS=1) or under RXD Stop Condition (RS=0).
<*B>: When valid data decreases below 15(AFT [7:4]) and nRTS is deasserted, the nRTS wil be asserted.
<*C>: The uart counterpart communicating with TCC8900 does not send data when nRTS is deasserted.
<*D>: nRTS is asserted according to Rx Avaialble FiFo Trigger Level.
- AFT [7:0] should be set in such a way that latency to process is considered by detecting nRTS of the uart counterpart communicating with TCC8900.

21.2.2 AFC (Auto Flow Control) in TX Operation

Register Set Value :
AFE (MCR[4]): Auto Flow Control Enable Bit = 1 and RTS(MCR[1]) Bit = 1

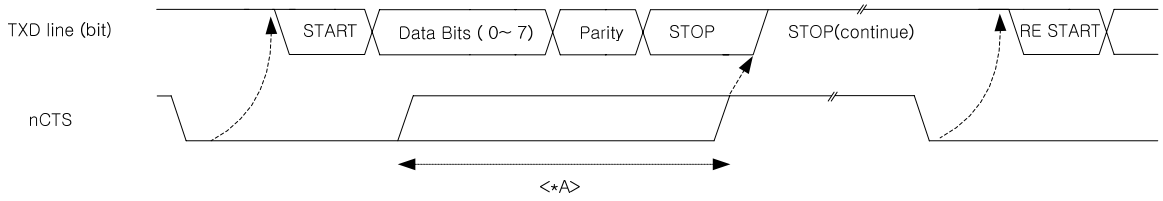


Figure 21.4 nCTS operation

<*A> : Even if nCTS is deasserted in the middle of TX transfer, current byte will be transferred.

21.2.3 Operation with General DMA

In UART channel 0~3, RX/TX data transmission through GDMA is available. Since in Channel 4 and 5, GDMA can not be used with hardware control style, in order to use it, make sure to use UART channel 0~3.

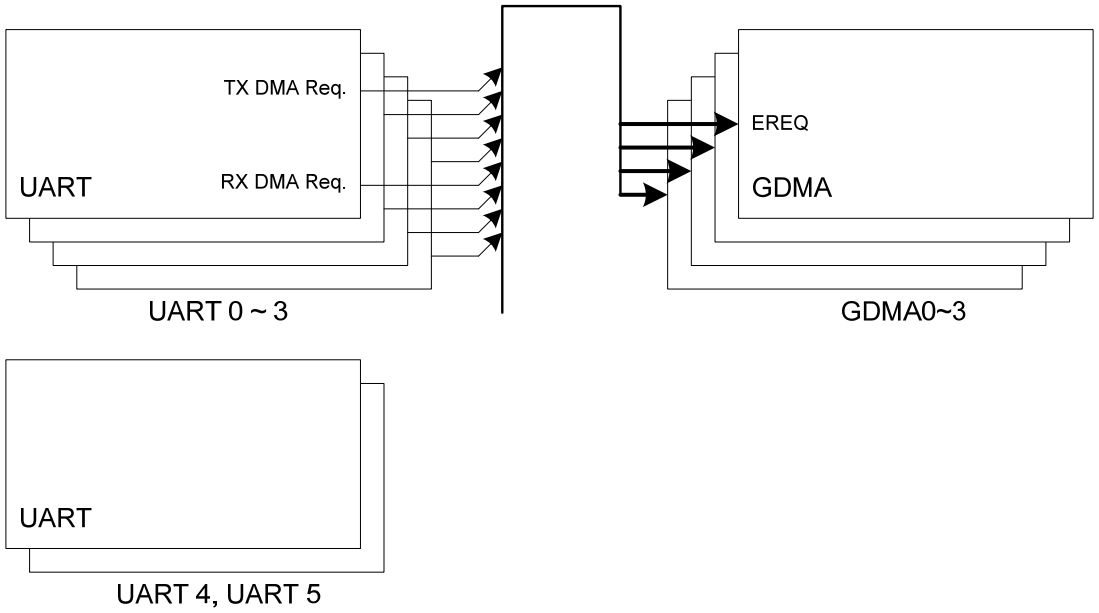


Figure 21.5 Operation with GDMA

21.2.4 RX Interrupt in DMA Transfer

Register Set Value :
FE(FCR[0]) : FiFo Enable Bit =1
RXT(FCR[7:6]) :
The Interrupt (or Rx Request) is asserted when
<*A> 2'b00 : 1 byte has been received. (CPU or DMA can read 1byte)
<*B> 2'b01 : 4 bytes has been received. (CPU or DMA can read 4bytes)
<*C> 2'b10 : 8 bytes has been received. (CPU or DMA can read 8bytes)
<*D> 3'b11 : 12 bytes has been received. (CPU or DMA can read 12bytes)

Cf. DMA Rx Request is asserted when RxDE(UCR[1]: Rx DMA Enable Bit)is Enabled.

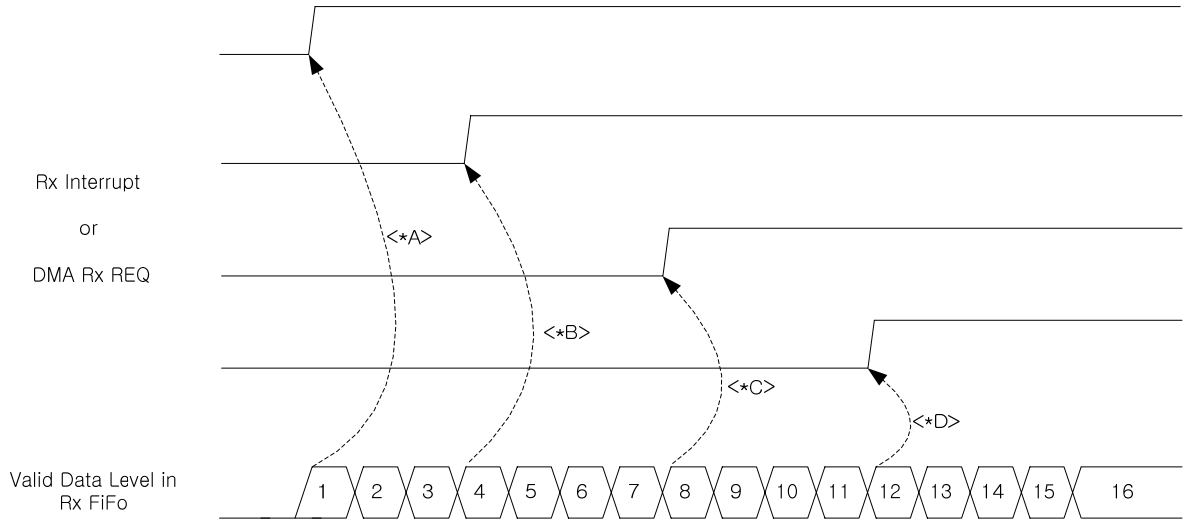


Figure 21.6 DMA operation – RX interrupt

21.2.4.1 TX Interrupt in DMA Transfer

Register Set Value :
FE(FCR[0]) : FiFo Enable Bit =1
TXT(FCR[5:4]) :
When Interrupt(or DMA Request) is asserted, The CPU (or DMA) is
<*A> 2'b00 : possible to write 16bytes at Tx FiFo (Empty)
<*B> 2'b01 : possible to write 8bytes at Tx FiFo
<*C> 2'b10 : possible to write 4bytes at Tx FiFo
<*D> 3'b11 : possible to write 1byte at Tx_FiFo

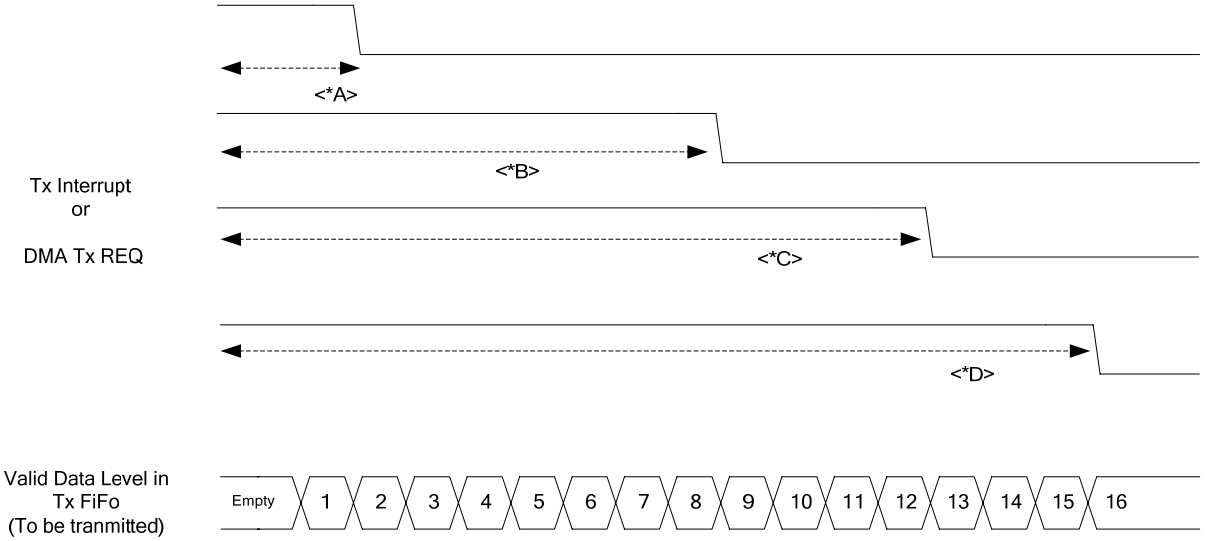
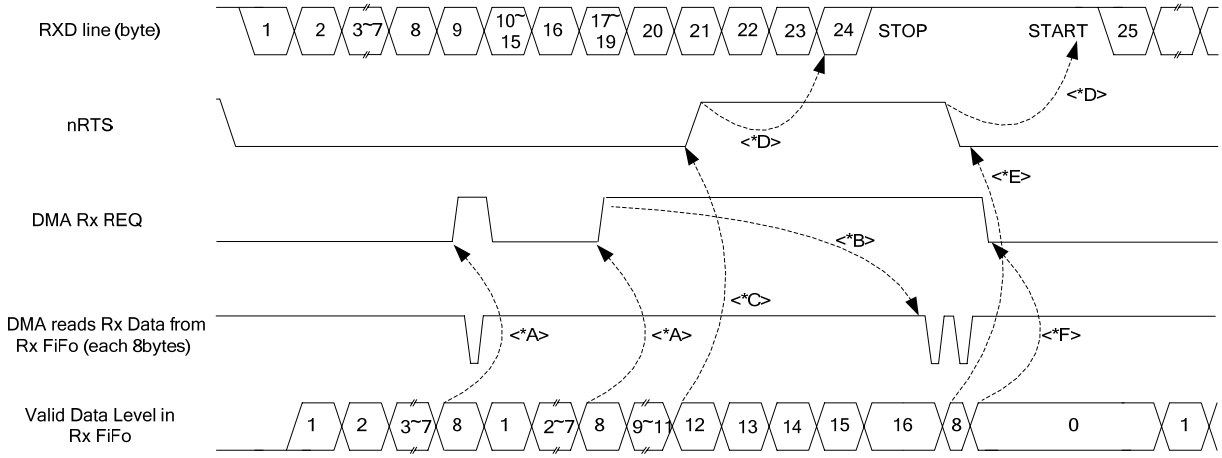


Figure 21.7 DMA operation – TX interrupt

21.2.5 RX Operation with DMA and AFC

UART Register Set Value
FE (FCR[0]) : FiFo Enable Bit = 1
AFE (MCR[4]): Auto Flow Control Enable Bit = 1
RxDE(UCR[1]: Rx DMA Enable bit = 1
DTL (AFT[3:0]) : nRTS Deassert Trigger Level = 12
ATL (AFT[7:4]) : nRTS Assert Trigger Level = 11
RXT (FCR[7:6]) : Rx Available FiFo Trigger Level = 8

DMA Register Set Value
TYPE(CHCTRL[9:8]) : H/W transfer with level sensitive = 2'b11
SYNC(CHCTRL[13]) : Syncronize External Request = 1
BSIZE(CHCTRL[7:6] : 8Read / 8Write = 2'b11 (depend on RXT)



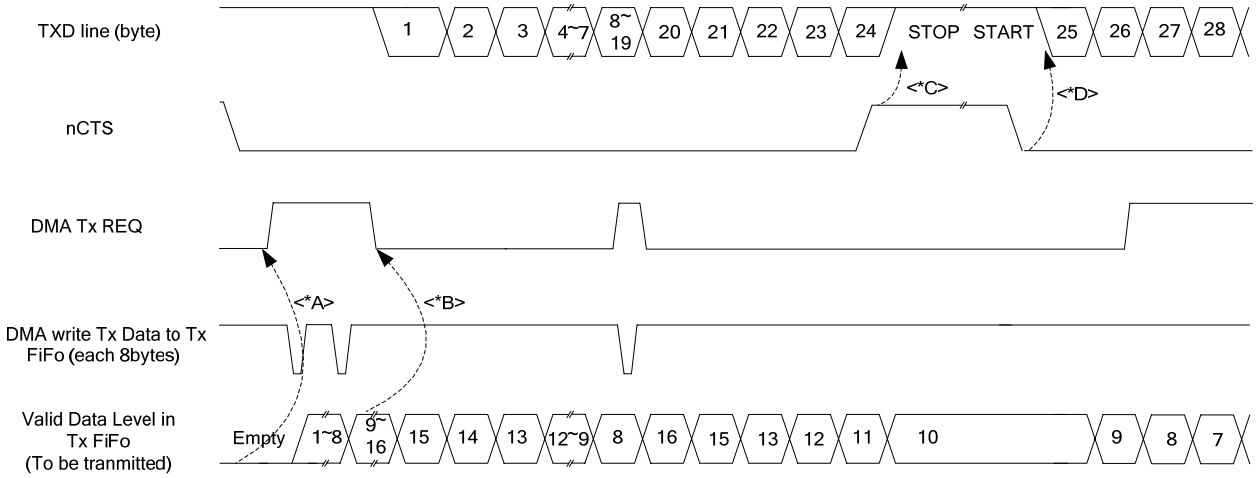
<*A> : DMA Rx Request is asserted by RXT
<*B> : DMA detects Rx Request but DMA can't transfer data (Other master higer than DMA occupies the bus)
<*C> : nRTS is reached by DTL
<*D> : The nRTS latency is between External UART communicated with TCCxx and TCCxxx.
The nRTS latency should be considered. if not, overrun error will occur
<*E> : nRTS is asserted by ATL
<*F> : DMA Rx Request is Deasserted By RxT

Figure 21.8 RX operation with DMA and AFC

21.2.6 TX Operation with DMA and AFC

UART Register Set Value
FE (FCR[0]) : FiFo Enable Bit = 1
TxDE(UCR[0] : Tx DMA Enable bit = 1
TXT (FCR[5:4]) : Tx FiFo Trigger Level = 8

DMA Register Set Value
TYPE(CHCTRL[9:8]) : H/W transfer with level sensitive = 2'b11
SYNC(CHCTRL[13]) : Syncronize External Request = 1
BSIZE(CHCTRL[7:6] : 8Read / 8Write = 2'b11 (depend on TXT)



<*A> : DMA Tx Request is asserted by TXT
<*B> : DMA Tx Request is deasserted by TXT

Figure 21.9 TX operation with DMA and AFC

21.3 Register Description

The base address of UART channel 0, 1, 2, 3, 4, and 5 are 0xF0532000, 0xF0532100, 0xF0532200, 0xF0532300, 0xF0532400, and 0xF0532500 respectively.

Table 21.1 UART Register Map

Name	Address	Type	Reset	Description
RBR	0x00	R	Unknown	Receiver Buffer Register(DLAB = 0)
THR	0x00	W	0x00	Transmitter Holding Register (DLAB=0)
DLL	0x00	R/W	0x00	Divisor Latch (LSB) (DLAB=1)
IER	0x04	R/W	0x00	Interrupt Enable Register (DLAB=0)
DLM	0x04	R/W	0x00	Divisor Latch (MSB) (DLAB=1)
IIR	0x08	R	Unknown	Interrupt Ident. Register (DLAB=0)
FCR	0x08	W	0xC0	FIFO Control Register (DLAB=1)
LCR	0x0C	R/W	0x03	Line Control Register
MCR	0x10	R/W	0x00	MODEM Control Register
LSR	0x14	R	Unknown	Line Status Register
MSR	0x18	R	Unknown	MODEM Status Register
SCR	0x1C	R/W	0x00	Scratch Register
AFT	0x20	R/W	0x00	AFC Trigger Level Register
UCR	0x24	R/W	0x00	UART Control Register
SRBR	0x40	R	Unknown	Rx Buffer Register
STHR	0x44	W	0x00	Transmitter Holding Register
SDLL	0x48	R/W	0x00	Divisor Latch (LSB)
SDLM	0x4C	R/W	0x00	Divisor Latch (MSB)
SIER	0x50	R/W	0x00	Interrupt Enable Register
SCCR	0x60	R/W	0x00	Smart Card Control Register
STC	0x64	R/W	0x00	Smart Card TX Count Register
IRCFG	0x80	R/W	0x00	IRDA Configuration Register

The base address of “Port Mux Registers” is 0xF0532600.

Table 21.2 UART Port Mux Register

Name	Address	Type	Reset	Description
CHSEL	0x00	R/W	0x3210	Channel Selection Register
CHST	0x00	R	0x0000	Channel Status Register

21.3.1 UART Controller Register

Receiver Buffer Register (RBR) **0xF0532n[28]00(DLAB=0)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Received Data (when reading)							

The RBR is actually the16byte FIFO, a received data from external device is stored in the RBR and CPU(or DMA) can read this register by Rx interrupt(or Rx DMA Request).

Transmitter Holding Register (THR) **0xF0532n00(DLAB=0)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Transmitting Data (when writing)							

The THR is actually the 16byte FIFO. To transmit data to external device, CPU(or DMA) should write data to the THR.

Divisor Latch Register (DLL) **0xF0532n00(DLAB=1)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Divisor Latch LSB							

Divisor Latch Register (DLM) **0xF0532n04(DLAB=1)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Divisor Latch MSB							

This is for generation of the desired baud rate clock.

The value can be calculated as follows.

{DLM, DLL} = fUART / (16 * desired baud rate)

For example,

If UART clock frequency is 48MHz (from CKC) and the baud-rate you want is 115,200 bps, the divisor value should be 26(48M/ (115200 x16)) in decimal.

In UART interface with general DMA, the UART clock frequency (from CKC) and configuration should be set by the following description.

- f_{UART} frequency
- CHCTRL.BST of GDMA = 0 (DMA Arbitration Mode Enable).

f_{UART} x 4 >= f_{BUS}

- CHCTRL.BST of GDMA = 1 (DMA Arbitration Mode Disable).

f_{UART} x 3 >= f_{BUS}

The following setting value(DMA/UART setting value) is example of RX/TX operation with DMA for the TCC8900.

- GDMA Setting Value

CHCTRL.BSIZE = 0 : 1Read/1Write (every UART request).

CHCTRL.WSIZE = 0 : 8bit data (1 byte).

[28] n = Channel Number 0 ~ 5.

- UART Setting Value

FCR.TXT = 0 : TX Empty Condition in TX FIFO.

FCR.RXT = 0 : Rx Available data Condition (minimum 1 character available data is received in Rx FIFO).

Interrupt Register (IER) **0xF0532n[29]04(DLAB=0)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												EMSI	ELSI	ETXI	ERXI

EMSI [3]	Type	Interrupt Enable Bit
0	R/W	Disable MODEM status interrupt
1	R/W	Enable MODEM status interrupt

ELSI [2]	Type	Interrupt Enable Bit
0	R/W	Disable receiver line status interrupt
1	R/W	Enable receiver line status interrupt

ETXI [1]	Type	Interrupt Enable Bit
0	R/W	Disable transmitter holding register empty interrupt
1	R/W	Enable transmitter holding register empty interrupt

ERXI [0]	Type	Interrupt Enable Bit
0	R/W	Disable received data available interrupt
1	R/W	Enable received data available interrupt

[29] n = Channel Number 0 ~5.

Interrupt Ident. Register (IIR) 0xF0532n[30]08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				STF											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0								0	0	0	0	IID[2:0]			IPF

STF [27]	Type	Smart Card TX done Flag
1	R	It represents that TX is done.

IID [3:1]	Type	Interrupt ID
011	R	Receiver line status
010	R	Received data available
110	R	Character timeout indication
001	R	Transmitter holding register empty
000	R	MODEM status

IID [3:1]	Type	Interrupt Source
011	R	Overrun/parity error/framing error/break error
010	R	Receiver data available or trigger level reached
110	R	While the last 4 characters are received, no characters have been removed from or input to the RX FIFO and there is at least 1 character in the RX FIFO.
001	R	Transmitter holding register empty
000	R	Clear to send data set ready or ring indicator or data carrier detect

IID [3:1]	Type	Interrupt Reset
011	R	Reading the line status register
010	R	Reading the receiver buffer register or the FIFO drops below the trigger level
110	R	Reading the receiver buffer register
001	R	Reading the IIR register(if source of interrupt) or writing into the transmitter holding register
000	R	Reading the MODEM status register

IPF [0]	Type	Interrupt Flag
0	R	Interrupt pending
1	R	Interrupt has not generated

[30] n = Channel Number 0 ~ 5.

FIFO Control Register (FCR) 0xF0532n[31]08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								RXT[1:0]		TXT[1:0]		DRTE	TXFR	RXFR	FE

RXT [7:6]	Type	RX FIFO Trigger Level(bytes)
00	W	1 byte
01	W	4 bytes
10	W	8 bytes
11	W	14 bytes

TXT [5:4]	Type	TX FIFO Trigger Level(bytes)
00	W	Possible to Write 16 byte (EMPTY)
01	W	Possible to Write 8 byte
10	W	Possible to Write 4 byte
11	W	Possible to Write 1 byte

These bits are used to configure the writable amount at FIFO when the TX interrupt or GDMA TX request is generated.

DRTE [3]	Type	DMA Mode Select
0	W	DMA transfer is depend on RxDE or TxDE bit.
1	W	Enable both Rx & Tx DMA transfer (Regardless of RxDE or TxDE status)

TXFR [2]	Type	TX FIFO Reset
1	W	Reset TX FIFO counter and FIFO data

RXFR [1]	Type	RX FIFO Reset
1	W	Reset RX FIFO counter and FIFO data

FE [0]	Type	FIFO Enable
1	W	Enable TX FIFO and RX FIFO.

[31] n = Channel Number 0 ~ 5.

Line Control Register (LCR) 0xF0532n[32]0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								DLAB	SB	SP	EPS	PEN	STB	WLS[1:0]	

DLAB [7]	Type	Divisor Latch Access
1	R/W	Access the divisor latches of the baud generator
0	R/W	Access the receiver buff, the transmitter holding register, or the interrupt enable register.

SB [6]	Type	Set Break
1	R/W	The serial output is forced to the spacing(logic 0) state
0	R/W	Disable the break

SP [5]	Type	Stick Parity
1	R/W	When PEN[3], EPS[4], and SP[5] are set to 1, the parity bit is transmitted and checked as a logic 0. If PEN[3] and SP[5] are set to 1 and EPS[4] is set to 0, then the parity bit is transmitted and checked as a logic 1
0	R/W	Disable stick parity

EPS [4]	Type	Even Parity Select
1	R/W	Generate or check even parity
0	R/W	Generate or check odd parity

PEN [3]	Type	Parity Enable
1	R/W	A parity bit is generated(TX) or checked(RX)

STB [2]	Type	Number of Stop Bits
1	R/W	One stop bit is generated in the transmitted data
0	R/W	When 5-bit word length is selected, one and a half stop bits are generated. When either 6, 7, or 8-bit word length is selected, two stop bits are generated.

WLS [1:0]	Type	Word Length Select
00	R/W	5 bits
01	R/W	6 bits
11	R/W	7 bits
11	R/W	8 bits

[32] n = Channel Number 0 ~ 5.

Modem Control Register (MCR) 0xF0532n[33]10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0									RS	AFE	LOOP	-	-	RTS	-

RS [6]	Type	RTS Deassert Condition Control Bit
0	R/W	nRTS is de-asserted at the Rx Stop Condition
1	R/W	nRTS is de-asserted at the Rx Start Condition (Reset value)

AFE [5]	Type	Hardware Auto Flow Control Enable Bit
0	R/W	Disable Automatic Flow Control
1	R/W	Enable Automatic Flow Control

LOOP [4]	Type	Loop Back Mode Enable
0	R/W	Disable local loop back feature
1	R/W	Enable local loop back feature

RTS [1]	Type	Request To Send
0	R/W	Set the nRTS line to high state
1	R/W	Reset the nRTS line to low state

AFE	RTS	Flow Control Configuration
1	1	Both nCTS and nRTS automatic flow control is activated
1	0	Only nCTS automatic flow control is activated
0	X	Both nCTS and nRTS automatic flow control is de-activated (Same as TCC77x series, only support for S/W Flow Control)

[33] n = Channel Number 0 ~ 5.

Line Status Register (LSR) 0xF0532n[34]14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								ERF	TEMT	THRE	BI	FE	PE	OE	DR

ERF [7]	Type	Divisor Latch Access
1	R	In the FIFO mode this bit is set when there is at least one parity error, framing error or break indication in the FIFO
0	R	In the 16450 mode

TEMT [6]	Type	Transmitter Empty
1	R	Transmitter holding register and the transmitter shift register are both empty

THRE [5]	Type	Transmitter Holding Register Empty
1	R	UART is ready to accept a new char for transmission

BI [4]	Type	Break Interrupt
1	R	The received data input is held in the spacing(logic 0) state for longer than a full word transmission time

FE [3]	Type	Framing Error
1	R	The received character did not have a valid stop bit

PE [2]	Type	Parity Error
1	R	The received data character does not have the correct even or odd parity

OE [1]	Type	Overrun Error
1	R	The receiver buffer register was not read by the CPU before the next character was transferred into the receiver buffer register.

DR [0]	Type	Data Ready
1	R	The receiver data ready

[34] n = Channel Number 0 ~ 5.

Modem Status Register(MSR) 0xF0532n[35]18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								reserved			CTS	reserved			DCTS

CTS [4]	Type	Clear to Send
	R	This bit is the complement of the Clear to Send input

DCTS [0]	Type	Delta Clear to Send
	R	This bit is the Delta Clear to Send indicator.

Scratch Register (SCR) 0xF0532n1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Scratch Data[7:0]							

This 8-bit Read/Write Register does not control the UART in anyway. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.

AFC Trigger Level Register (AFT) 0xF0532n20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								ATL[3:0]				DTL[3:0]			

ATL	Type	[3:0]	nRTS Assert Trigger Level
N	R/W	nRTS assert trigger level	

DTL	Type	[3:0]	nRTS Deassert Trigger Level
N	R/W	nRTS de-assert trigger level	

[35] n = Channel Number 0 ~ 5.

Control Register (UCR) 0xF0532n[36]24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												RWA	TWA	RxDE	TxDE

RWA	Type	[3]	Rx Word Access
0	R/W	Rx FIFO access to byte	
1	R/W	Rx FIFO access to word (4 bytes)	

TWA	Type	[2]	Tx Word Access
0	R/W	Tx FIFO access to byte	
1	R/W	Tx FIFO access to word (4 bytes)	

RxDE	Type	[1]	Rx DMA Enable
0	R/W	Rx DMA disable	
1	R/W	Rx DMA enable	

TxDE	Type	[0]	Tx DMA Enable
0	R/W	Tx DMA disable	
1	R/W	Tx DMA enable	

[36] n = Channel Number 0 ~ 5.

Receiver Buffer Register (SRBR) **0xF0532n[37]40**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Received Data (Read Only)							

Transmitter Holding Register (STHR) **0xF0532n44**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Transmitting Data (Write Only)							

Divisor Latch Register (SDLL) **0xF0532n48**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Divisor Latch LSB							

Divisor Latch Register (SDLM) **0xF0532n4C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Divisor Latch MSB							

Interrupt Register (SIER) **0xF0532n50**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												EMSI	ELSI	ETXI	ERXI

SRBR, STHR, SDLL, SDLM, SIER registers are copy of the RBR, THR, DLL, DLM, IER registers. These registers can be accessed without concern of DLAB state.

[37] n = Channel Number 0 ~ 5.

SmartCard Configuration Register (SCCR) 0xF0532n[38]60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEN	STEN	DEN	P	STF	STE										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIV [15:0]															

SEN	Type	[31]	Smart Card Interface Enable
1	R/W	The smart card interface is enabled	

STEN	Type	[30]	SmartCard Tx Enable
1	R/W	Enable	

DEN	Type	[29]	SmartCard Clock Divider Enable
1	R/W	Enable	

P	Type	[28]	SmartCard Clock Post Divider Enable
1	R/W	Enable	

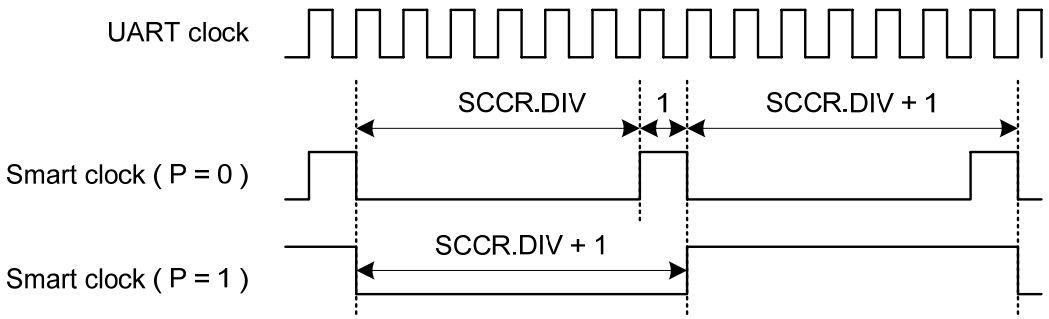
STF	Type	[27]	SmartCard Tx done Flag
1	R	It represents that all characters specified by S_CNT are transmitted. It can be TX done interrupt source.	
1	W	When '1' is written to it, it is cleared.	

STE	Type	[26]	SmartCard Tx done Intrrupt Enable
1	R	Enable Tx done Interrupt	

DIV	Type	[15:0]	SmartCard Clock Divisor
1	R/W	Fout = Fuart/((DIV+1)*2^P)	

Fout and Fuart can not be the same. Therefore, In the case that DIV = 0 or DEN = 0, Smart clock operates only under the condition that P is set to 1 (P = 1).

DEN	DIV	P	Smart clock
0	x	0	Disabled
0	x	1	Fout = Fuart / 2
1	0	0	Disabled
1	0	1	Fout = Fuart / 2
1	N (!= 0)	P	Fout = Fuart / ((N + 1) * 2 ^ P)



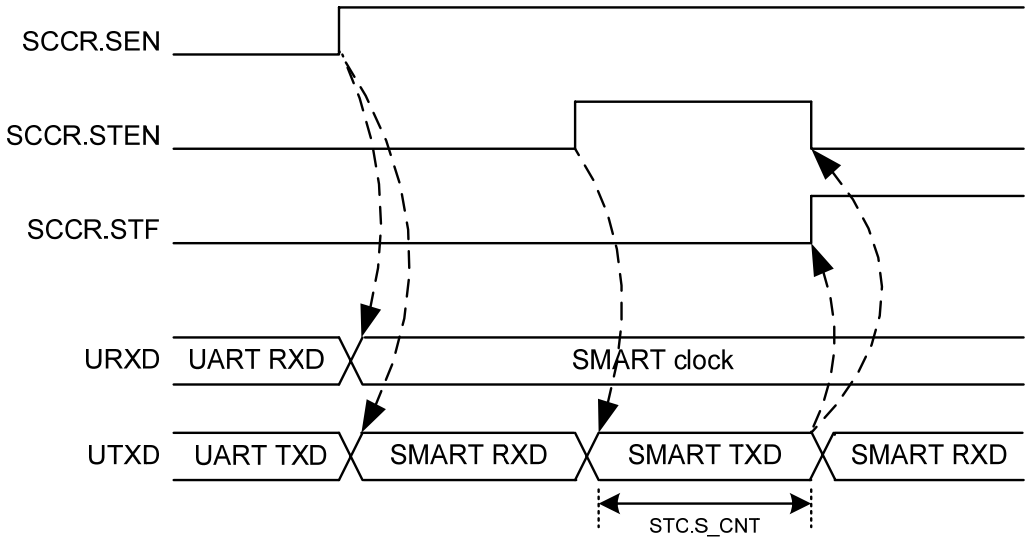
[38] n = Channel Number 0 ~ 5.

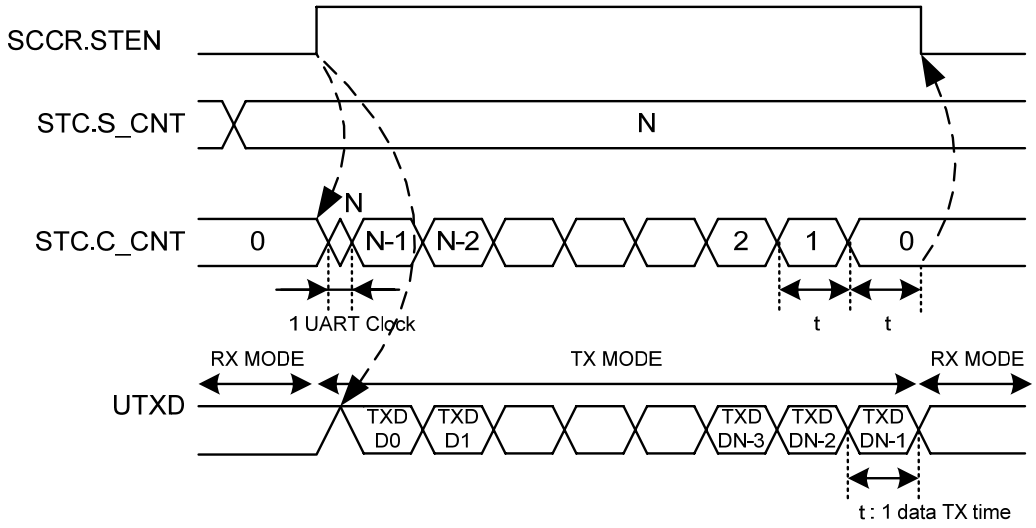
SmartCard Tx Count (STC) **0xF0532n64**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C_CNT[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S_CNT[15:0]															

C_CNT	Type	[31:16]	Current Tx Count
N	R	It represents current TX count when smart card interface is enabled	

S_CNT	Type	[15:0]	Start Tx Count
N	R/W	When smart card interface is enabled, it specifies TX Count	





IRDA Configuration Register (IRCFG) 0xF0532n[39]80

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												RXP	TXP	RXE	TXE

In IrDA Mode, each zero bit of TXD has a pulse width of 3/16 of a bit time.

RXP	Type	[3]	RXD polarity
1	R/W	Invert polarity of IrDA-RXD signal	

TXP	Type	[2]	TXD polarity
1	R/W	Invert polarity of IrDA-TXD signal	

RXE	Type	[1]	Rx Enable
1	R/W	Enable IrDA-Rx	

TXE	Type	[0]	TX Enable
1	R/W	Enable IrDA-Tx	

[39] n = Channel Number 0 ~ 5.

21.3.2 Port Mux Register

TCC8900 has 6 UART ports, and each port can be allocated for one of UART channels. Base address is 0xF0055400

Channel Selection Register (CHSEL) **0xF0532600**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				0						CH5		0		CH4	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0		CH3		0		CH2		0		CH1		0		CH0	

The value of CH0, CH1, CH2, CH3, CH4, and CH 5 is the corresponding port number. Therefore, when CH0 is set to 3, UART channel0 is connected to UART port 3. Note that the value for each channel should be different.

Channel Status Register **0xF0532604**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
										CH5	CH4	CH3	CH2	CH1	CH0

When one or more UART channels are issuing an interrupt request, the corresponding bit is set to 1.

22 DMA CONTROLLER

22.1 Overview

The TCC8900 has four 3-channel general DMA (GDMA) controllers for data transfer. The block diagram of GDMA controller is in the following figure.

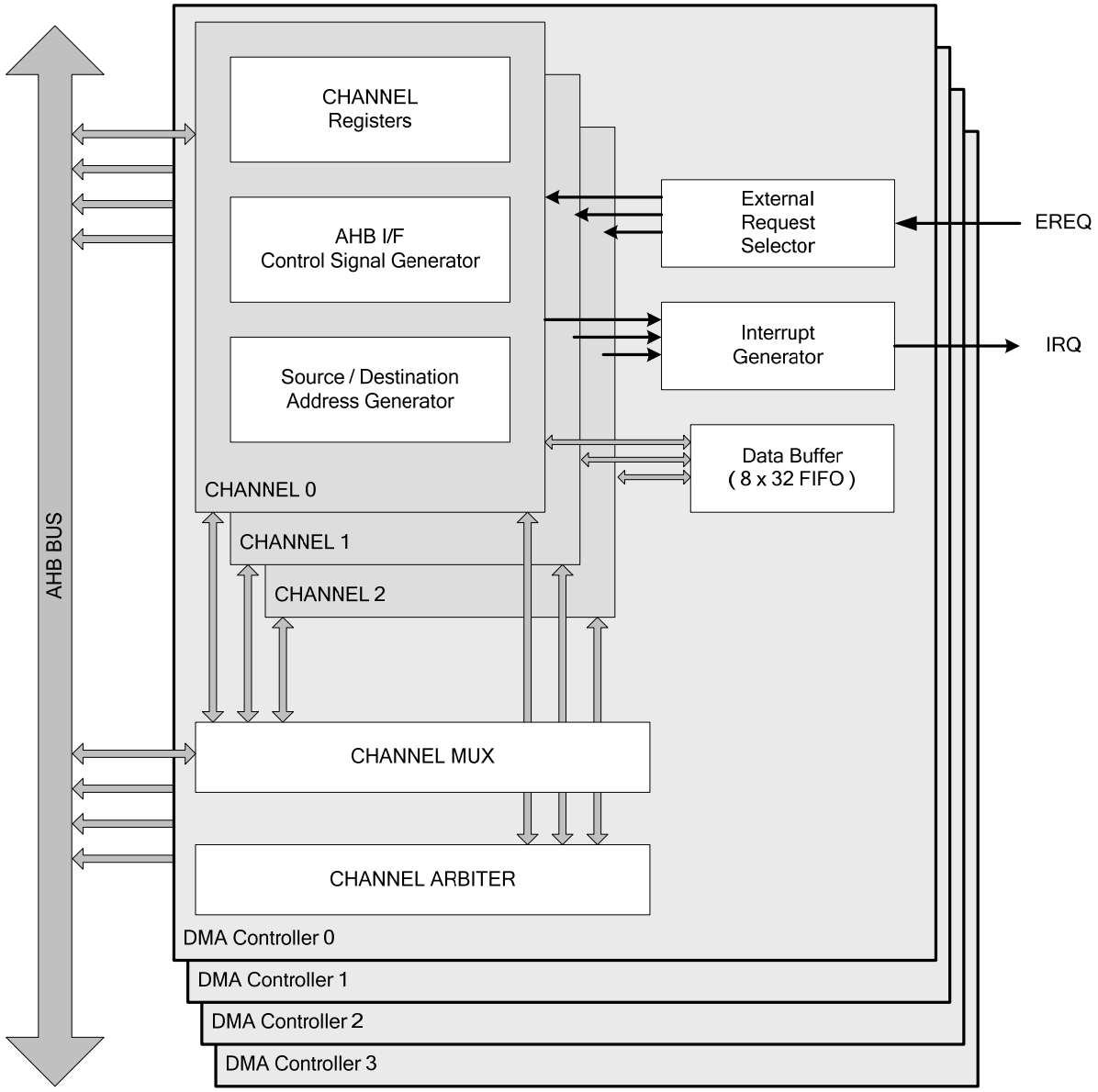


Figure 22.1 GDMA Controller Block Diagram

22.2 Register Description

Table 22.1 General DMA Controller Register Map(DMA Base Address = 0xF0540n00, n=0,1,2,3)

Name		offset	Type	Reset	Description
CHANNEL 0	ST_SADR0	0x00	R/W	0x00000000	Start Address of Source Block
	SPARAM0	0x04	R/W	0x00000000	Parameter of Source Block
	C_SADR0	0x0C	R	0x00000000	Current Address of Source Block
	ST_DADR0	0x10	R/W	0x00000000	Start Address of Destination Block
	DPARAM0	0x14	R/W	0x00000000	Parameter of Destination Block
	C_DADR0	0x1C	R	0x00000000	Current Address of Destination Block
	HCOUNT0	0x20	R/W	0x00000000	Initial and Current Hop count
	CHCTRL0	0x24	R/W	0x00000000	Channel Control Register
	RPTCTRL0	0x28	R/W	0x00000000	Repeat Control Register
	EXTREQ0	0x2C	R/W	0x00000000	External DMA Request Register
CHANNEL 1	ST_SADR1	0x30	R/W	0x00000000	Start Address of Source Block
	SPARAM1	0x34	R/W	0x00000000	Parameter of Source Block
	C_SADR1	0x3C	R	0x00000000	Current Address of Source Block
	ST_DADR1	0x40	R/W	0x00000000	Start Address of Destination Block
	DPARAM1	0x44	R/W	0x00000000	Parameter of Destination Block
	C_DADR1	0x4C	R	0x00000000	Current Address of Destination Block
	HCOUNT1	0x50	R/W	0x00000000	Initial and Current Hop count
	CHCTRL1	0x54	R/W	0x00000000	Channel Control Register
	RPTCTRL1	0x58	R/W	0x00000000	Repeat Control Register
	EXTREQ1	0x5C	R/W	0x00000000	External DMA Request Register
CHANNEL 2	ST_SADR2	0x60	R/W	0x00000000	Start Address of Source Block
	SPARAM2	0x64/0x68	R/W	0x00000000	Parameter of Source Block
	C_SADR2	0x6C	R	0x00000000	Current Address of Source Block
	ST_DADR2	0x70	R/W	0x00000000	Start Address of Destination Block
	DPARAM2	0x74/0x78	R/W	0x00000000	Parameter of Destination Block
	C_DADR2	0x7C	R	0x00000000	Current Address of Destination Block
	HCOUNT2	0x80	R/W	0x00000000	Initial and Current Hop count
	CHCTRL2	0x84	R/W	0x00000000	Channel Control Register
	RPTCTRL2	0x88	R/W	0x00000000	Repeat Control Register
	EXTREQ2	0x8C	R/W	0x00000000	External DMA Request Register
CHCONFIG		0x90	R/W	0x00000000	Channel Configuration Register

The GDMA registers are listed in Table 22.1. The TCC8900 has 4 GDMA, which are GDMA0, GDMA1, GDMA2, and GDMA3. And their base addresses are 0xF0540000, 0xF0540100, 0xF0540200 and 0xF0540300 respectively. One GDMA has 3 channels, which are Channel 0, Channel 1 and Channel 2. Channel offset is 0x00, 0x30, and, 0x60 as shown in Table 22.1 and Table 22.2

Table 22.2 BASE Address(DMA_BASE) of All GDMA Channles

GDMA NAME	GDMA BASE	CHANNEL #	BASE Address
GDMA0	0xF0540000	0	0xF0540000
		1	0xF0540030
		2	0xF0540060
GDMA1	0xF0540100	0	0xF0540100
		1	0xF0540130
		2	0xF0540160
GDMA2	0xF0540200	0	0xF0540200
		1	0xF0540230
		2	0xF0540260
GDMA3	0xF0540300	0	0xF0540300
		1	0xF0540330
		2	0xF0540360

Start Source Address Register (ST_SADR) DMA_BASE+0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST_SADR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ST_SADR[15:0]															

This register contains the start address of source memory block for DMA transfer. The transfer begins reading data from this address.

Source Block Parameter Register (SPARAM) DMA_BASE + 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMASK[23:8]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SMASK[7:0]								SINC[7:0]							

SMASK[23:0]	[31:8]	Source Address Mask Register
0	non-masked	
1	Masked so that source address bit doesn't be changed during DMA transfer	

Each bit field controls the dedicated bit of source address field. That is, if SMASK[23] is set to 1, the 28th bit of source address is masked, and if SMASK[22] is set to 1, the 27th bit of source address is masked, and so on. If a bit is masked, a corresponding bit of address bus is not changed during DMA transfer. This function can be used to generate circular buffer address.

SINC[7:0]	[7:0]	Source Address Increment Register
sinc	Source address is added by amount of sinc at every write cycles. sinc is represented as 2's complement, so if SINC[7] is 1, the source address is decremented.	

The addresses of DMA transfer are 32bit wide, but the upper 4bit of them are not affected during DMA transfer. If the source or destination address reaches its maximum address space like 0x7FFFFFFF or 0x2FFFFFFF, the next transfer is starting from 0x70000000 or 0x20000000 not from 0x80000000 or 0x30000000.

Current Source Address Register (C_SADR) DMA_BASE + 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C_SADR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C_SADR[15:0]															

This register contains the current source address of DMA transfer. It represents that the current transfer read data from this address. This is read only register.

Start Destination Address Register (ST_DADR) DMA_BASE + 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST_DADR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ST_DADR[15:0]															

This register contains the start address of destination memory block for DMA transfer.

Destination Block Parameter Register (DPARAM) DMA_BASE + 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMASK[23:8]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMASK[7:0]								DINC[7:0]							

DMASK[28:0]	[31:8]	Destination Address Mask Register
0	non-masked	
1	Masked so that destination address bit doesn't be changed during DMA transfer	

Each bit field controls the corresponding bit of source address field. That is, if DMASK[23] is set to 1, the 28th bit of source address is masked, and if DMASK[22] is set to 1, the 27th bit of source address is masked, and so on. If a bit is masked, a corresponding bit of address bus is not changed during DMA transfer. This function can be used to generate circular buffer address.

DINC[7:0]	[7:0]	Destination Address Increment Register
dinc	Destination address is added by amount of dinc at every write cycles. dinc is represented as 2's complement, so if DINC[7] is 1, the destination address is decremented.	

The addresses of DMA transfer have 32bit wide, but the upper 4bit of them are not affected during DMA transfer. If the source or destination address reaches its maximum address space like 0x7FFFFFFF or 0x2FFFFFFF, the next transfer is starting from 0x70000000 or 0x20000000 not from 0x80000000 or 0x30000000.

Current Destination Address Register (C_DADR) DMA_BASE + 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C_DADR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C_DADR[15:0]															

This register contains current destination address of DMA transfer. It represents that the current transfer write data to this address. This is read only register.

HOP Count Register (HCOUNT) DMA_BASE + 0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C_HCOUNT[15:0]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ST_HCOUNT[15:0]															

C_HCNT[15:0]	[31:16]	Current Hop Count
cn	Represent cn number of Hop transfer remains	

ST_HCNT[15:0]	[15:0]	Start Hop Count
sn	DMA transfers data by amount of sn Hop transfers	

At the beginning of transfer, the C_HCNT is updated by ST_HCNT register. At the end of every hop transfer, this is decremented by 1 until it reaches to zero. When this reaches to zero, the DMA finishes its transfer and may or may not generate its interrupt according to IEN flag of CHCTRL register.

Channel Control Register (CHCTRL) **DMA_BASE + 0x24**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CONT	DTM	SYN	HRD	LOCK	BST	TYPE		BSIZE		WSIZE		FLG	IEN	REP	EN

CONT	[15]	Issue Continuous Transfer
0	DMA transfer begins from ST_SADR / ST_DADR address	
1	DMA transfer begins from C_SADR / C_DADR address It must be used after the former transfer has been executed, so that C_SADR and C_DADR contain a meaningful value.	

DTM	[14]	Differential Transfer Mode
0	Differential Transfer Mode Disable	
1	Differential Transfer Mode Enable for WSIZE = 10 and BSIZE = 11: 32 bit-to- 16bit transfer 4 Read(Word Unit) / 8 Write(Half-Word Unit) for WSIZE = 11 and BSIZE = 11 : 16 bit-to- 32bit transfer. 8 Read(Half-Word Unit) / 4 Write(Word Unit)	

SYNC	[13]	Hardware Request Synchronization
0	Do not Synchronize Hardware Request.	
1	Synchronize Hardware Request.	

HRD	[12]	Hardware Request Direction
0	ACK/EOT signals are issued when DMA-Read Operation.	
1	ACK/EOT signals are issued When DMA-Write Operation.	

LOCK	[11]	Issue Locked Transfer
1	DMA transfer executed with lock transfer	

Lock field controls the LOCK signal (refer to AHB specification). When the LOCK is set to 1, the DMA transfer is not bothered by other AHB masters like LCD controller, ARM etc. This field is only meaningful in case of non-burst type transfers.

BST	[10]	BURST Transfer
0	DMA transfer executed with arbitration.	
1	DMA transfer executed with no arbitration. (burst operation)	

Arbitration means that at the end of every HOP transfer, the AHB bus is released from DMA channel so other master can occupy the bus when that master has requested the bus.

Burst means that once the DMA request occurs, all of transfers are executed without further DMA requests.

TYPE[1:0]	[9:8]	Transfer Type
00	SINGLE transfer with edge-triggered detection	
11	SINGLE transfer with level-sensitive detection	
01	HW transfer	
10	SW transfer	

In SINGLE Type, after one Hop data transferring DMA checks External DMA Request (DREQ) and then if its bit is active, DMA transfers next hop data. DREQ is detected level-sensitive or edge-triggered by SINGLE transfer TYPE.

The 1 Hop of transfer means 1 burst of read followed by 1 burst of write. 1 burst means 1, 2 or 4 consecutive read or write-cycles defined by BSIZE field of CHCTRL register. The Figure 22.2 illustrates the relation among the above transfers.

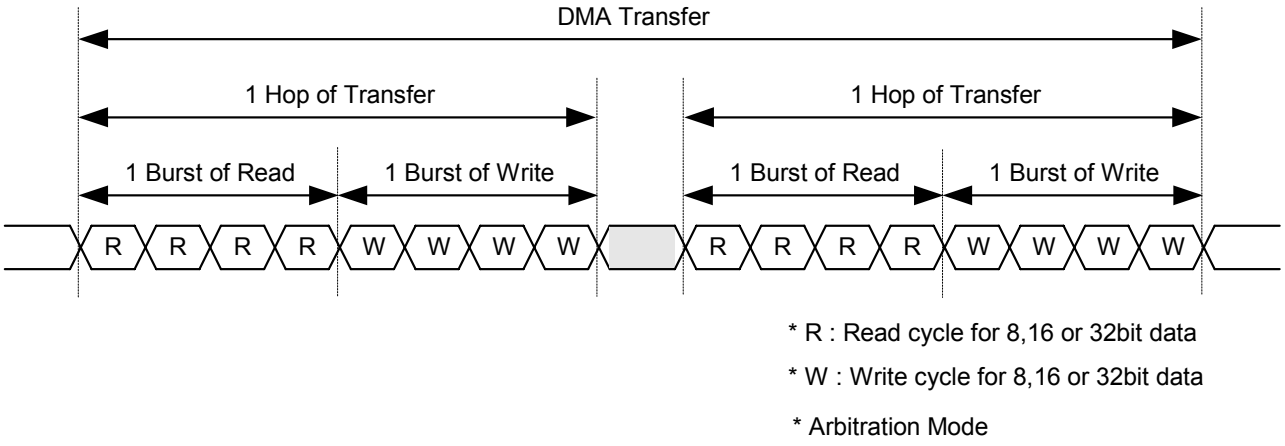


Figure 22.2 Relation between Hop and Burst Transfers (If burst size is 4.)

Hardware type transfer means that the DMA transfer is triggered by external or internal hardware blocks selected by DMASEL field in CHCTRL register.

Software type transfer means that the DMA transfer is triggered by EN bit of CHCTRL register. When this is set to 1, transfer request signal is generated internally and then the transfer begins immediately.

Hardware demand type transfer (HW_DEMAND) means that once the DMA request occurs,

DMA checks request signal each hope transfer, and if request signal is set, DMA transfer one hope's data. After transferring all hope's data, DMA operation will be finished.

Figure 22.3 shows the example of various types of transfer.

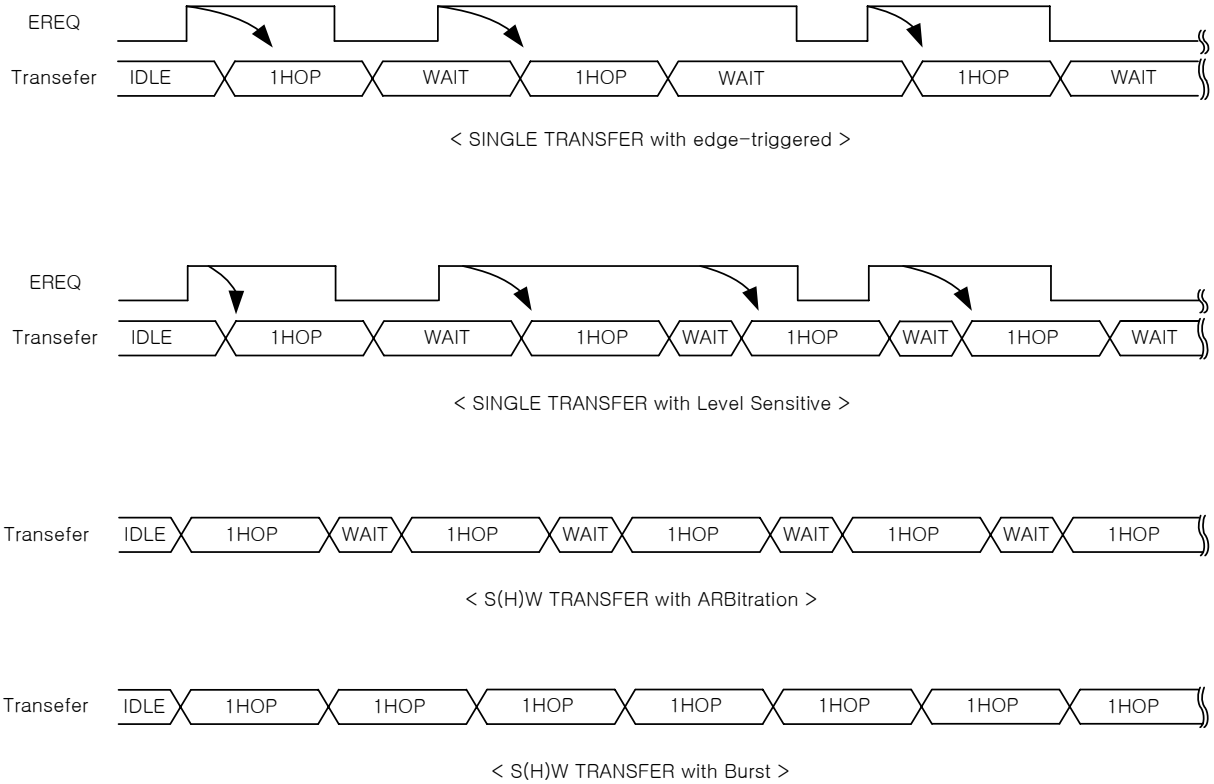


Figure 22.3 The Example Of Various Types of Transfer.

BSIZE[1:0]	[7:6]	Burst Size
0	1 Burst transfer consists of 1 read or write cycle.	
1	1 Burst transfer consists of 2 read or write cycles	
2	1 Burst transfer consists of 4 read or write cycles	
3	1 Burst transfer consists of 8 read or write cycles	

WSIZE[1:0]	[5:4]	Word Size
0	Each cycle read or write 8bit data	
1	Each cycle read or write 16bit data	
2, 3	Each cycle read or write 32bit data	

FLAG	[3]	DMA Done Flag
1	Represents that all hop of transfers are fulfilled. When writing 1 to this bit, it is cleared to 0	

It is not automatically cleared by starting another transfer, so before starting any other DMA transfer, user must clear this flag to 0 for checking DMA status correctly.

IEN	[2]	Interrupt Enable
1	At the same time the FLAG goes to 1, DMA interrupt request is generated.	

To generate IRQ or FIQ interrupt, the DMA flag of IEN register in the interrupt controller must be set to 1 ahead.

REP	[1]	Repeat Mode Control
0	After all of hop transfer has executed, the DMA channel is disabled	
1	The DMA channel remains enabled. When another DMA request has occurred, the DMA channel start transfer data again with the same manner (type, address, increment, mask) as the latest transfer of that channel.	

EN	[0]	DMA Channel Enable
0	DMA channel is terminated and disabled. It does not affect the HCOUNT register, so if the current hop counter is not zero when channel is disabled, it is possible that the transfer illegally starts right after channel is re-enabled. Make sure that HCOUNT is zero not to continue transfer after channel is re-enabled.	
1	DMA channel is enabled. If software type transfer is selected, this bit generates DMA request directly, or if hardware type transfer is used, the selected interrupt request flag generate DMA request.	

Repeat Control Register (RPTCTRL) DMA_BASE + 0x28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRI	EOT	0						RPTCNT[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RPTCNT[15:0]															

DRI	[31]	Disable Repeat Interrupt
0	DMA Interrupt is occurred when the end of each Repeated DMA operation.	
1	DMA Interrupt occur is occurred when the last DMA Repeated DMA operation	

This bit is meaningful when Repeat Mode is enabled.

EOT	[30]	EOT
0	EOT signal is occurred when the end of each Repeated DMA operation in HW(including Single) transfer Mode.	
1	EOT Signal is occurred when the last Repeated DMA operation in HW(including Single) transfer Mode.	

This bit is meaningful when Repeat Mode is enabled.

RPTCNT[23:0]	[23:0]	Repeat Count
0	DMA transfer data endlessly.	
None zero	DMA transfer the number of (N + 1) * HCOUNT data	

This bit is meaningful when Repeat Mode is enabled. When this bit is cleared in repeat mode,

DMA will run endlessly. To exit endless repeat mode, clear EN bit of DMACTRL or disable Repeat Mode. It's possible to circular transfer using repeat count.

External DMA Request Register (EXTREQn) **DMA_BASE + 0x2C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMASEL[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMASEL[15:0]															

DMASEL	Connected Hardware
[0]	GPSB Channel 3 TX
[1]	GPSB Channel 4 TX
[2]	GPSB Channel 5 TX
[3]	TS I/F 0
[4]	GPSB Channel 3 RX
[5]	GPSB Channel 4 RX
[6]	GPSB Channel 5 RX
[7]	TS I/F 1
[8]	UART Channel 2 Transmitter
[9]	UART Channel 2 Receiver
[10]	UART Channel 3 Transmitter
[11]	UART Channel 3 Receiver
[12]	TS I/F PID0
[13]	SATA RX
[14]	SATA TX
[15]	Test Block
[16]	Memory Stick 0
[17]	Memory Stick 1
[18]	NAND Flash Controller
[19]	I2C Channel 0
[20]	SPDIF Transmitter - Packet(Audio) Data
[21]	SPDIF Transmitter - User Data
[22]	CD I/F
[23]	DAI Transmitter
[24]	DAI Receiver
[25]	I2C Channel 1
[26]	UART Channel 0 Transmitter
[27]	UART Channel 0 Receiver
[28]	SPDIF Receiver
[29]	UART Channel 1 Transmitter
[30]	UART Channel 1 Receiver
[31]	TS I/F PID1

Channel Configuration Register (CHCONFIG) DMA_BASE +0x90

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0									IS2	IS1	IS0	0	MIS2	MIS1	MIS0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0					SWP2	SWP1	SWP0	0	PRI[2:0]			0			FIX

IS2	[22]	Channel 2 Alternate interrupt status
0	No interrupt in the corresponding channel	
1	The corresponding channel interrupt is occurred	

Without regard to Interrupt enable bit(IEN) of the channel2, this bit indicates the channel2 interrupt status.

This bit is automatically cleared when FLAG bit of the corresponding channel is cleared. This bit is read only.

IS1	[21]	Channel 1 Alternate interrupt status
0	No interrupt in channel 1	
1	Channel1 Interrupt is occurred	

Except for channel difference, this bit is the same as IS2 bit.

IS0	[20]	Channel 0 Alternate interrupt status
0	No interrupt in channel 0	
1	Channel1 Interrupt is occurred	

Except for channel difference, this bit is the same as IS2 bit.

MIS2	[18]	Channel2 Masked Interrupt Status
0	Masked interrupt is not occurred in channel 2	
1	Channel2 Masked Interrupt is occurred	

This bit is set when the channel2 interrupt occurs and interrupt enable bit (IEN) of channel2 is set. This bit is automatically cleared when FLAG bit of the channel2 is cleared. This bit is read only.

MIS1	[17]	Channel1 Masked Interrupt Status
0	Masked interrupt is not occurred in channel 1	
1	Channel1 Masked Interrupt is occurred	

Except for channel difference, this bit is the same as MIS2 bit.

MIS0	[16]	Channel0 Masked Interrupt Status
0	Masked interrupt is not occurred in channel 0	
1	Channel0 Masked Interrupt is occurred	

Except for channel difference, this bit is the same as MIS1 bit.

SWP2	[10]	Channel2 SWAP Enable bit
0	Do not swap Data.	
1	Swap Channel Data.	

When this bit is set, data to be written to destination address will be swapped.

For example, the 32bit source data which consists of 4bytes {D3, D2, D1, D0} will be stored {D0, D1, D2, D3} in destination address. The 16bit source data which consists of 2bytes {D1, D0} will be stored {D0,D1} in destination address.

SWP1	[9]	Channel1 SWAP Enable bit
0	Do not Swap Channel1 Data.	
1	Swap Channel1 Data.	

Except for channel difference, the function controlled by this bit is the same as its SWP2 bit.

SWP0	[8]	Channel0 SWAP Enable bit
0	Do not Swap Channel0 Data.	
1	Swap Channel0 Data.	

Except for channel difference, the function controlled by this bit is the same as its SWP2 bit.

PRI[2:0]	[5:4]	Channel Priority
000	CH0 > CH1 > CH2	
001	CH0 > CH2 > CH1	
010	CH1 > CH0 > CH2	
011	CH1 > CH2 > CH0	
100	CH2 > CH1 > CH0	
101	CH2 > CH0 > CH1	

PRI bits is meaningful when fix[0] bit is enabled.

FIX	[0]	Fixed Priority Operation
0	Round-Robin (Cyclic) Mode.	
1	Fixed Priority Mode.	

In round-robin mode, each channel is enabled one by one every one hop transferring.

In Fixed mode, according to PRI bit, the highest channel is serviced first and lower priority channel is serviced after higher priority channel operation is finished. See Figure 22.4 for more information.

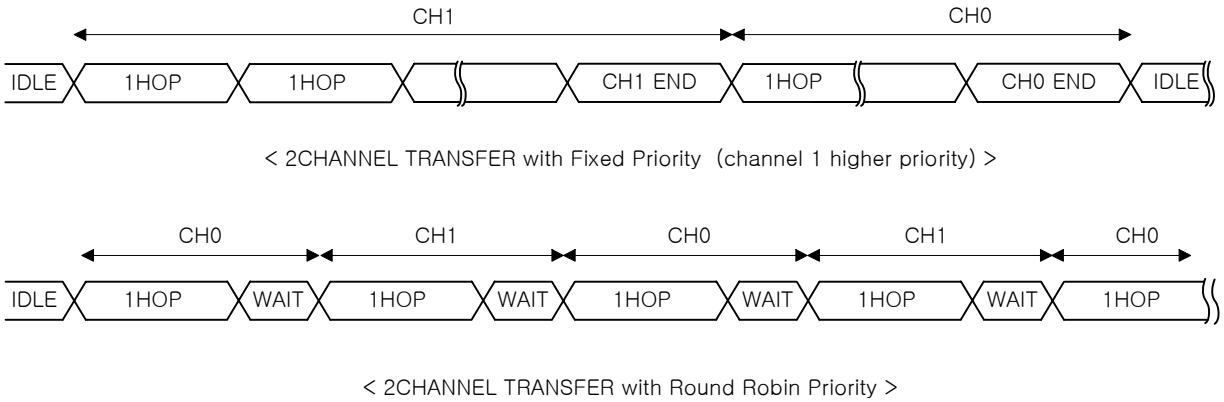


Figure 22.4 Data transfer when Channel0 and Channel1 are enabled

23 RTC (Real-Time Clock)

23.1 Overview

The Real Time Clock (RTC) unit can be operated by the backup battery if the system power is turned off. The RTC provides time keeping data to CPU as BCD (binary coded decimal) values. The data includes second, minute, hour, date, day of the week, month, and year. The RTC unit also can perform the alarm function. The block diagram is shown in Figure 16.1.

Features

- Clock and calendar functions (BCD display): seconds, minutes, hours, date, day of week, month, year
- Leap year generator
- Wake-up (PMWKUP) signal generation: support on the power down mode (PWDN)
- Alarm interrupt (ALMINT) in normal operation mode
- Supports 32.768kHz or 4.194304MHz XTAL input
- Power Supply Voltage: System power supply(3.0V, 1.2V), Backup battery (3.0V)

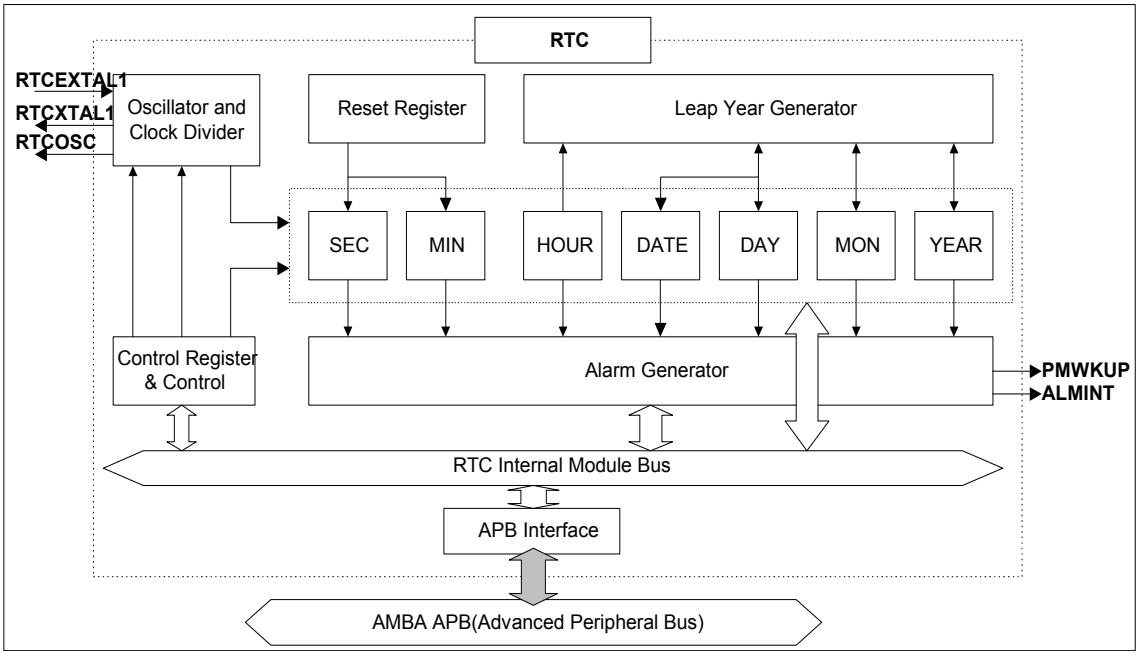


Figure 23.1 RTC Block Diagram

23.2 Function Description

23.2.1 System Clock Frequency Control

The leap year generator calculates whether the last date of each month is 28, 29, 30 or 31 based on data from BCDDAY, BCDMON, and BCDYEAR. This also considers leap years in deciding the last date. A 16 bit counter can just represent four BCD digits, so it can decide whether any year is a leap year or not.

23.2.2 System Power Operation

It is required to set bit 1 of the RTCCON register for interfacing between CPU and RTC logic. An one second error can occur when the CPU reads or writes data into BCD counters and this can cause the change of the higher time units. When the CPU reads/writes data to/from the BCD counters, another time unit may be changed if BCDSEC register is overflowed. To avoid this problem, the CPU should reset BCDSEC register to 00h. The reading sequence of the BCD counters is BCDYEAR, BCDMON, BCDDATE, BCDDAY, BCDHOUR, BCDMIN, and BCDSEC. It is required to read it again from BCDYEAR to BCDSEC if BCDSEC is zero.

23.2.3 Backup Battery Operation

The RTC logic is driven by backup battery if the system power turns off. The interfaces of the CPU and RTC logic are blocked and the battery drives the oscillation circuit, clock divider (DIVIDER), and BCD counters (RTCFCN) to minimize power dissipation.

23.2.4 Alarm Function

The RTC generates alarm signal at specified time in the power down mode or normal operation mode. In normal operation mode, the alarm interrupt (ALMINT) is activated and in the power down mode the power management wake up (PMWKUP) signal is activated. The RTC alarm register, RTCALM, determines the alarm enable and the condition of the alarm time setting.

23.3 RTC Operation

23.3.1 Boot-Up Sequence

The figure below shows how to initialize register of RTCINT block (RTCALM, RTCIM, RTCPEND).

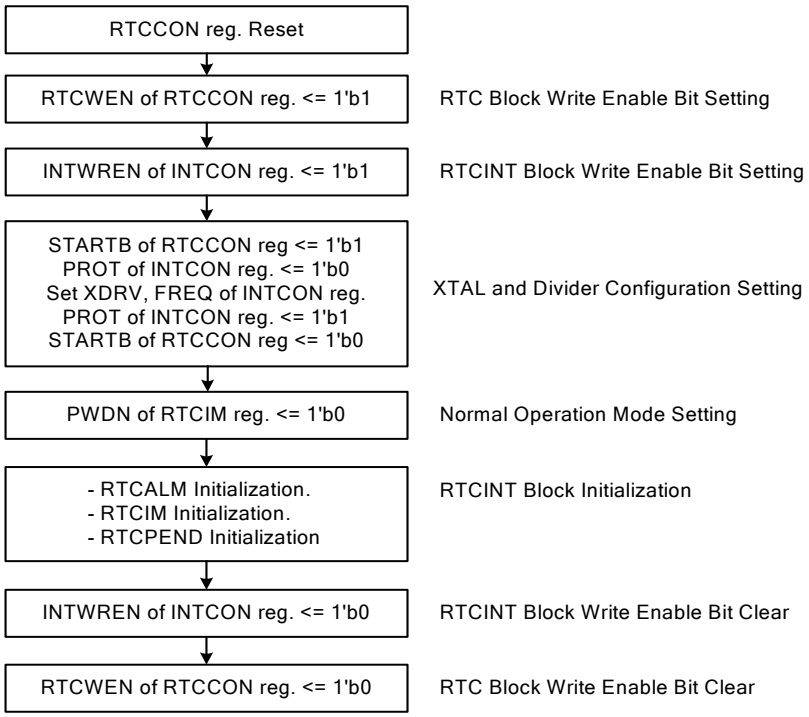


Figure 23.2 Boot-Up Sequence

23.3.2 RTC Time Setting

The following figure shows how to set the time when clock is stopped. This works when the entire calendar or clock is to be set.

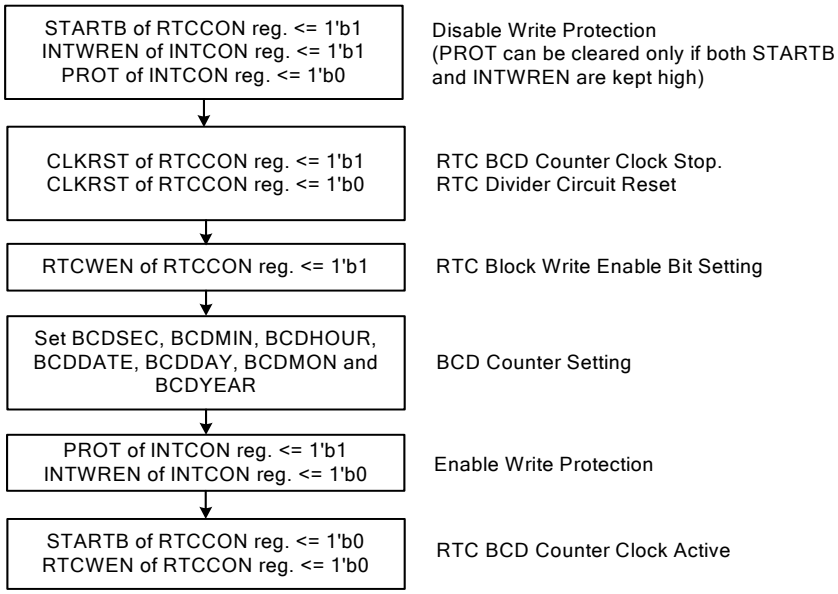


Figure 23.3 The RTC Time Setting Sequence

23.3.3 RTC Alarm Time Setting

The following figure shows how to use the alarm function. Alarms can be generated using the seconds, minutes, hours, days of week, date, month, year or any combination of these. Set the ALMEN bit (bit 7) in the register on which the alarm is

placed to "1", and then set the alarm time. Clear the ALMEN bit in the register on which the alarm is placed to "0".

When the INTMODE bit of RTCIM register is high, and the clock and alarm times match, "1" is set in the PEND bit of RTCPEND register. The detection of alarm can be checked with reading the PEND bit.

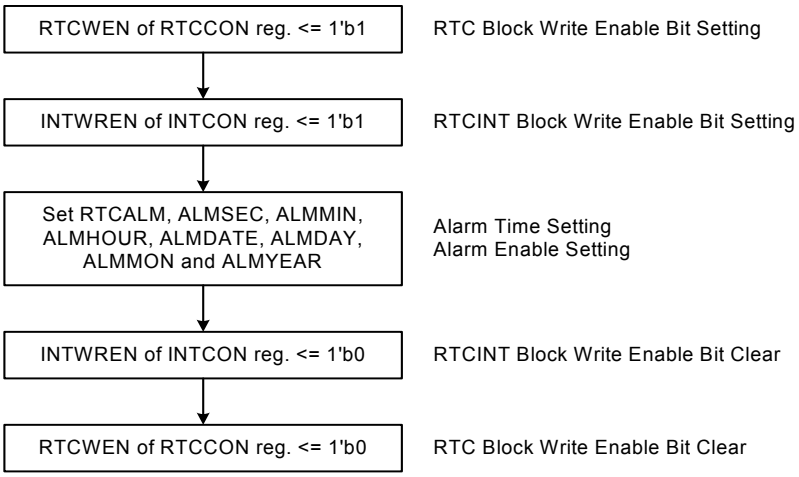


Figure 23.4 RTC Alarm Time Setting Sequence

Note: To generate the alarm interrupt, you must set INTWREN (INTCON[0])

23.3.4 RTCPEND Clear

The following figure shows how to Clear RTC interrupt

There are two types of interrupt.

- Alarm Interrupt (ALMINT).
- Wake-Up Interrupt (WKUPINT).

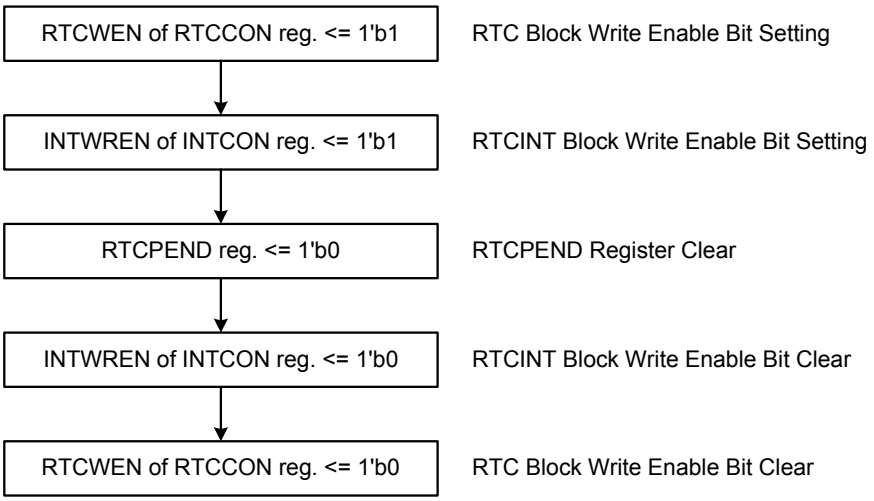


Figure 23.5 PEND Clear Sequence

23.3.5 RTC Operation

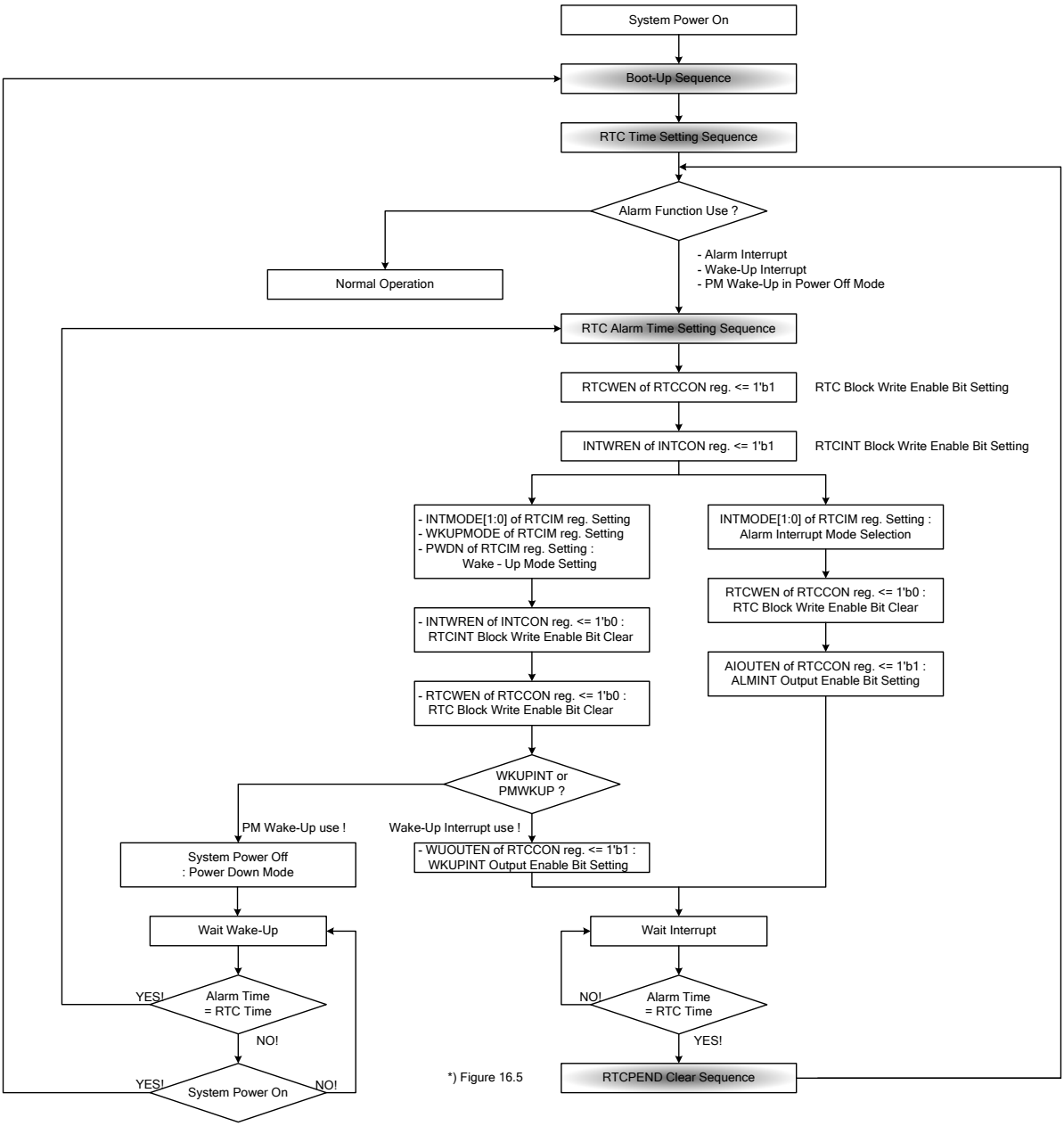


Figure 23.6 RTC Operation Process Flow Chart.

23.3.6 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in the following table and the RTC crystal oscillator circuit in the following figure.

Table 23.1 Recommended Oscillator Circuit Constants

fosc	C1	C2	Rf
32.768 kHz	from 10 pF to 22 pF	from 10 pF to 22 pF	> 5MΩ
4.194304 MHz	TBD	TBD	TBD

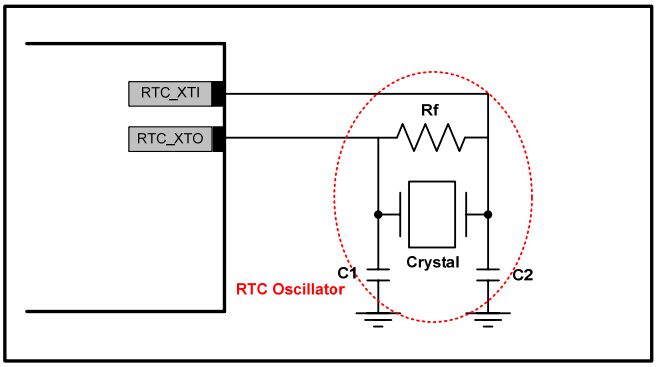


Figure 23.7 Example of Crystal Oscillator Circuit Connection

23.4 Programmer's model

23.4.1 Register memory map

Table 23.2 RTC Register Map (Base Address = 0xF05F2000)

Register	Address	R/W	Reset value	Description
RTCCON	0x00	R/W	0x00	RTC Control Register
INTCON	0x04	R/W	-	RTC Interrupt Control Register
RTCALM	0x08	R/W	-	RTC Alarm Control Register
ALMSEC	0x0C	R/W	-	Alarm Second Data Register
ALMMIN	0x10	R/W	-	Alarm Minute Data Register
ALMHOUR	0x14	R/W	-	Alarm Hour Data Register
ALMDATE	0x18	R/W	-	Alarm Date Data Register
ALMDAY	0x1C	R/W	-	Alarm Day of Week Data Register
ALMMON	0x20	R/W	-	Alarm Month Data Register
ALMYEAR	0x24	R/W	-	Alarm Year Data Register
BCDSEC	0x28	R/W	-	BCD Second Register
BCDMIN	0x2C	R/W	-	BCD Minute Register
BCDHOUR	0x30	R/W	-	BCD Hour Register
BCDDATE	0x34	R/W	-	BCD Date Register
BCDDAY	0x38	R/W	-	BCD Day of Week Register
BCDMON	0x3C	R/W	-	BCD Month Register
BCDYEAR	0x40	R/W	-	BCD Year Register
RTCIM	0x44	R/W	-	RTC Interrupt Mode Register
RTCPEND	0x48	R/W	-	RTC Interrupt Pending Register
RTCSTR	0x4C	R/W	0x00	RTC Interrupt Status Register

23.4.2 Register Description

RTC Control Register (RTCCON) **0xF05F2000**

RTCCON register consists of 8-bits such as STARTB that controls of running the normal counters, RTCWEN that controls the read/write enable of the BCD registers, CLKSEL, CNTSEL, and CLKRST for BCD counters testing.

RTCWEN bit controls all interfaces between the CPU and the RTC, so it should be set to '1' in an initialization routine to enable data transfer after a system reset. Instead of working BCD with 1Hz, CLKSEL bit enables the operation of BCD counters with an external clock which is applied through the pin EXTAL1 to the test BCD counters. CNTSEL bit converts the dependent

The operation of BCD counters into independent counters for the test. CLKRST resets the frequency divided logic in the RTC.

OSCEN bit controls the path from input of Crystal to the output of divider logic. If this bit is high, the output of divider is 1 Hz clock. This bit is implemented to test the oscillator circuit and divider block.

INTWREN bit controls the path from the RTCIF Block to the RTCINT Block.

NOTE: CLKRST, CNTSEL and CLKSEL are affected by PROT and INTWERN of INTCON register. They are valid only if (~PROT & INTWREN & STARTB).

Table 23.3 RTC Control Register (RTCCON)

RTCCON	Bit	Initial State	Description
STARTB	[0]	0	RTC start bit 0: RUN 1: Halt
RTCWEN	[1]	0	RTC write enable bit 0: Disable 1: Enable
CLKSEL	[2]	0	BCD counter test clock set bit 0: EXTAL1 divided clock (1 Hz) 1: Reserved (EXTAL1: 32.768 kHz)
CNTSEL	[3]	0	BCD count test type set bit 0: Merge BCD counters 1: Reserved (Separate BCD counters)
CLKRST	[4]	0	RTC clock count set bit 0: No reset 1: reset

OSCEN	[5]	0	Oscillator and Divider circuit test enable bit 0: Disable 1: Enable
AIOUTEN	[6]	0	Alarm Interrupt Output Enable 0: Disable 1: Enable
WUOUTEN	[7]	0	Wake Up Output Enable 0: Disable 1: Enable

Note: If PROT bit of INTCON register is set as "1", the BCD counters start normal operation with 1Hz clock regardless of STARTB status.

RTC Interrupt Control Register (INTCON) 0xF05F2004

Table 23.4 RTC Interrupt Control Register (INTCON)

INTCON	Bit	Initial State	Description
INTWREN	[0]	Undef.	Interrupt Block Write Enable bit 0: Disable 1: Enable
STATUS	[2:1]	Undef.	User Define Status Register
FSEL	[10:8]	Undef.	Divider Output Select 3'b000: 32.768kHz XTAL 3'b111: 4.194304MHz XTAL
XDRV	[13:12]	Undef.	XTAL I/O Driver Strength Select 2'b00: 32.768KHz XTAL @ 1.8V 2'b01: 32.768KHz XTAL @ 3.3V 2'b10: 4.194304MHz XTAL @ 1.8V 2'b11: 4.194304MHz XTAL @ 3.3V
PROT	[15]	Undef.	Protection Enable Bit 0: Disable. 1: Enable. If enabled, the following fields are affected. - XDRV, FSEL of INTCON register are write protected - All the BCD Data Registers are write protected - CLKRST, CNTSEL and CLKSEL bits of RTCCON reigister are made non-functional To clear this bit, RTCWEN, STARTB and INTWREN must be set first. To enable this bit, RTCWEN and STARTB must be set first.

To write INTCON register, you must set RTCWEN (RTCCON[1]).

FSEL, and XDRV bits can be written only if (RTCWEN & STARTB & INTWREN & ~PROT)

To enable PROT bit, RTCWEN and STARTB must be set first.

To clear PROT bit, RTCWEN, STARTB and INTWREN must be set first.

RTC Alarm Control Register (RTCALM) 0xF05F2008

RTCALM register determines the alarm enable and the condition of the alarm time setting. Note that RTCALM register generates the alarm signal through ALMINT in normal operation mode.

Alarms can be generated using the seconds, minutes, hours, day of week, date, month, year or any combination of these. Set the ALMEN bit (bit 7) in the register on which the alarm is placed to "1", and then set the alarm time. Clear the ALMEN bit in the register on which the alarm is placed to "0".

Table 23.5 RTC Alarm Control Register (RTCALM)

RTCALM	Bit	Initial State	Description
SECEN	[0]	Undef.	Second alarm enable bit 0 : Disable 1 : Enable
MINEN	[1]	Undef.	Minute alarm enable bit 0 : Disable 1 : Enable
HOUREN	[2]	Undef.	Hour alarm enable bit 0 : Disable 1 : Enable
DATEEN	[3]	Undef.	Date alarm enable bit 0 : Disable 1 : Enable
DAYEN	[4]	Undef.	Day of week alarm enable bit 0 : Disable 1 : Enable
MONEN	[5]	Undef	Mon alarm enable bit 0 : Disable 1 : Enable
YEAREN	[6]	Undef	Year alarm enable bit 0 : Disable 1 : Enable
ALMEN	[7]	Undef.	Alarm global enable bit 0: Disable 1 : Enable

Note: To Write RTCALM register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read RTCALM register, you must set INTWREN (INTCON[0]).

Alarm Second Data Register (ALMSEC) 0xF05F200C

Table 23.6 Alarm Second Data Register (ALMSEC)

ALMSEC	Bit	Initial State	Description
SECDATA	[6:0]	Undef.	BCD value for alarm second bits [3:0] bit is from 0 to 9 [6:4] bit is from 0 to 5

Note: To Write ALMSEC register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMSEC register, you must set INTWREN (INTCON[0]).

Alarm Minute Data Register (ALMMIN) 0xF05F2010

Table 23.7 Alarm Minute Data Register (ALMMIN)

ALMMIN	Bit	Initial State	Description
MINDATA	[6:0]	Undef.	BCD value for alarm second bits [3:0] bit is from 0 to 9 [6:4] bit is from 0 to 5

Note: To Write ALMMIN register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMMIN register, you must set INTWREN (INTCON[0]).

Alarm Hour Data Register (ALMHOUR) 0xF05F2014

Table 23.8 Alarm Hour Data Register (ALMHOUR)

ALMHOUR	Bit	Initial State	Description
HOURDATA	[5:0]	Undef.	BCD value for alarm hour bits [3:0] bit is from 0 to 9 [5:4] bit is from 0 to 2

Note: To Write ALMHOUR register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMHOUR register, you must set INTWREN (INTCON[0]).

Alarm Date Data Register (ALMDATE) 0xF05F2018

Table 23.9 Alarm Date Data Register (ALMDATE)

ALMDATE	Bit	Initial State	Description
DATEDATA	[5:0]	Undef.	BCD value for alarm date, from 0 to 28, 29, 30, 31 (decimal: 01 ~ 31) [3:0] bit is from 0 to 9 [5:4] bit is from 0 to 3

Note: To Write ALMDATE register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMDATE register, you must set INTWREN (INTCON[0]).

Alarm Day of Week Data Register (ALMDAY) 0xF05F201C

Table 23.10 Alarm Day of Week Data Register (ALMDAY)

ALMDAY	Bit	Initial State	Description
DAYDATA	[2:0]	Undef.	BCD value for alarm day bits [2:0] bit is from 0 to 6 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday

			110: Saturday

Note: To Write ALMDAY register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMDAY register, you must set INTWREN (INTCON[0]).

Alarm Month Data Register (ALMMON) 0xF05F2020

Table 23.11 Alarm Month Data Register (ALMMON)

ALMMON	Bit	Initial State	Description
MONDATA	[4:0]	Undef.	BCD value for alarm month bits [3:0] bit is from 0 to 9 [4] bit is from 0 to 1

Note: To Write ALMMON register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMMON register, you must set INTWREN (INTCON[0]).

Alarm Year Data Register (ALMYEAR) 0xF05F2024

Table 23.12 Alarm Year Data Register (ALMYEAR)

ALMYEAR	Bit	Initial State	Description
YEARDATA	[15:0]	Undef.	BCD value for alarm year bits [7:0] bit is from 0 to 99 [15:8] bit is from 0 to 99

Note: To Write ALMYEAR register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMYEAR register, you must set INTWREN (INTCON[0]).

BCD Second Data Register (BCDSEC) 0xF05F2028

Table 23.13 BCD Second Data Register (BCDSEC)

BCDSEC	Bit	Initial State	Description
SECDATA	[6:0]	Undef.	BCD value for second bits [3:0] bit is from 0 to 9 [6:4] bit is from 0 to 5

Note: To Write BCDSEC register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]) and clear PROT(INTCON[15]).

To Read BCDSEC register, you must set INTWREN (INTCON[0]).

BCD Minute Data Register (BCDMIN) 0xF05F202C

Table 23.14 BCD Minute Data Register (BCDMIN)

BCDMIN	Bit	Initial State	Description
MINDATA	[6:0]	Undef.	BCD value for minute bits [3:0] bit is from 0 to 9 [6:4] bit is from 0 to 5

Note: To Write BCDMIN register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]) and clear PROT(INTCON[15]).

To Read BCDMIN register, you must set INTWREN (INTCON[0]).

BCD Hour Data Register (BCDHOUR) 0xF05F2030

Table 23.15 BCD Hour Data Register (BCDHOUR)

BCDHOUR	Bit	Initial State	Description
HOURDATA	[5:0]	Undef.	BCD value for hour bits [3:0] bit is from 0 to 9 [5:4] bit is from 0 to 2

Note: To Write BCDHOUR register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]) and clear PROT(INTCON[15]).

To Read BCDHOUR register, you must set INTWREN (INTCON[0]).

BCD Date Data Register (BCDDATE) 0xF05F2034

Table 23.16 BCD Date Data Register (BCDDATE)

BCDDATE	Bit	Initial State	Description
DATEDATA	[5:0]	Undef.	BCD value for date bits(decimal : 01 ~ 31) [3:0] bit is from 0 to 9 [5:4] bit is from 0 to 3

Note: To Write BCDDATE register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]) and clear PROT(INTCON[15]).

To Read BCDDATE register, you must set INTWREN (INTCON[0]).

BCD Day of Week Data Register (BCDDAY) 0xF05F2038

Table 23.17 BCD Day of Week Data Register (BCDDAY)

BCDDAY	Bit	Initial State	Description
DATEDAY	[2:0]	Undef.	BCD value for date bits [2:0] bit is from 0 to 6 000 : Sunday 001 : Monday 010 : Tuesday 011 : Wednesday 100 : Thursday 101 : Friday 110 : Saturday

Note: To Write BCDDAY register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]) and clear PROT(INTCON[15]).

To Read BCDDAY register, you must set INTWREN (INTCON[0]).

BCD Month Data Register (BCDMON) 0xF05F203C

Table 23.18 BCD Month Data Register (BCDMON)

BCDMON	Bit	Initial State	Description
MONDATA	[4:0]	Undef.	BCD value for month bits [3:0] bit is from 0 to 9 [4] bit is from 0 to 1

Note: To Write BCDMON register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]) and clear PROT(INTCON[15]).

To Read BCDMON register, you must set INTWREN (INTCON[0]).

BCD Year Data Register (BCDYEAR) 0xF05F2040

Table 23.19 BCD Year Data Register (BCDYEAR)

BCDYEAR	Bit	Initial State	Description
YEARDATA	[15:0]	Undef.	BCD value for year bits [7:0] bit is from 0 to 99 [15:8] bit is from 0 to 99

Note: To Write BCDYEAR register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]) and clear PROT(INTCON[15]).

To Read BCDYEAR register, you must set INTWREN (INTCON[0]).

RTC Interrupt Mode Register (RTCIM) 0xF05F2044

When the INTMODE bit of RTCIM register is high, and the clock and alarm times match, "1" is set in the PEND bit of RTCPEND register. The detection of alarm can be checked with reading the PEND bit.

Table 23.20 RTC Interrupt Mode Register (RTCIM)

RTCIM	Bit	Initial State	Description
INTMODE	[1:0]	0	Interrupt mode selection bit x0: Disable alarm interrupt mode. x1: Enable alarm interrupt mode. 01: Supports on the edge alarm interrupt. 11: Supports on the level alarm interrupt.
WKUPMODE	[2]	1	Wakeup mode selection bit 0: PMWKUP active low 1: PMWKUP active high
PWDN	[3]	0	Operation mode selection bit 0: Normal Operation Mode 1: Power Down Mode

Note: To Write RTCIM register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read RTCIM register, you must set INTWREN (INTCON[0]).

To generate the alarm interrupt, you must be set INTMODE (RTCIM[0] = 1), PWDN (RTCIM [3] = 0) and RTCALM register should be set properly. To generate PMWKUP signal, PWDN register should be set to power down mode.

RTC Interrupt Pending Register (RTCPEND) 0xF05F2048

Table 23.21 RTC Interrupt Pending Register (RTCPEND)

RTCPEND	Bit	Initial State	Description
PEND	[0]	0	Interrupt pending enable bit 0 : PEND bit is cleared. 1 : PEND bit is pending.

Note: To Clear RTCPEND register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])
To Read RTCPEND register, you must set INTWREN (INTCON[0]).

RTC Interrupt Status Register (RTCSTR) 0xF05F204C

Table 23.22 RTC Interrupt Status Register (RTCSTR)

RTCSTR	Bit	Initial State	Description
CNT	[3:0]	0	Count value of alarm/wake-up interrupt
ALM	[6]	0	RTC Alarm interrupt pending 1 : RTC Alarm interrupt was generated To clear this bit, write ‘1’
PRI	[7]	0	RTC wake-up interrupt pending 1 : RTC wake-up interrupt was generated To clear this bit, write ‘1’

NOTE: RTC Interrupt = ALM | PRI

24 TSADC Interface

24.1 Overview

The TCC8900 has 8 channel general purpose low-power ADC for battery level detection, key detection, remote control interface, touch screen interface, etc. It is a CMOS type 10/12-bit A/D converter with 8-channel analog input multiplexer.It converts the analog input signal into 10/12-bit binary digital codes at a maximum conversion rate of 1 MSPS with 5MHz A/D converter clock(CKIN). A/D converter operates with on-chip sample-and-hold function. The power down mode is supported. Touch-screen interface controls input pads(XP,XM,YP,YM) to obtain X/Y-position on the 4-wire touch screen device.

- Resolution : 10-bit / 12-bit
- Maximum Conversion Rate : 1MSPS
- Main Clock : 5MHz (Max.)
- Input Range : 0.0V ~ VDDA_ADC
- Normal Conversion mode
- Separate X/Y Conversion mode
- Auto(Sequential) X/Y Conversion mode
- Waiting for Interrupt mode
- STOP mode wakeup source

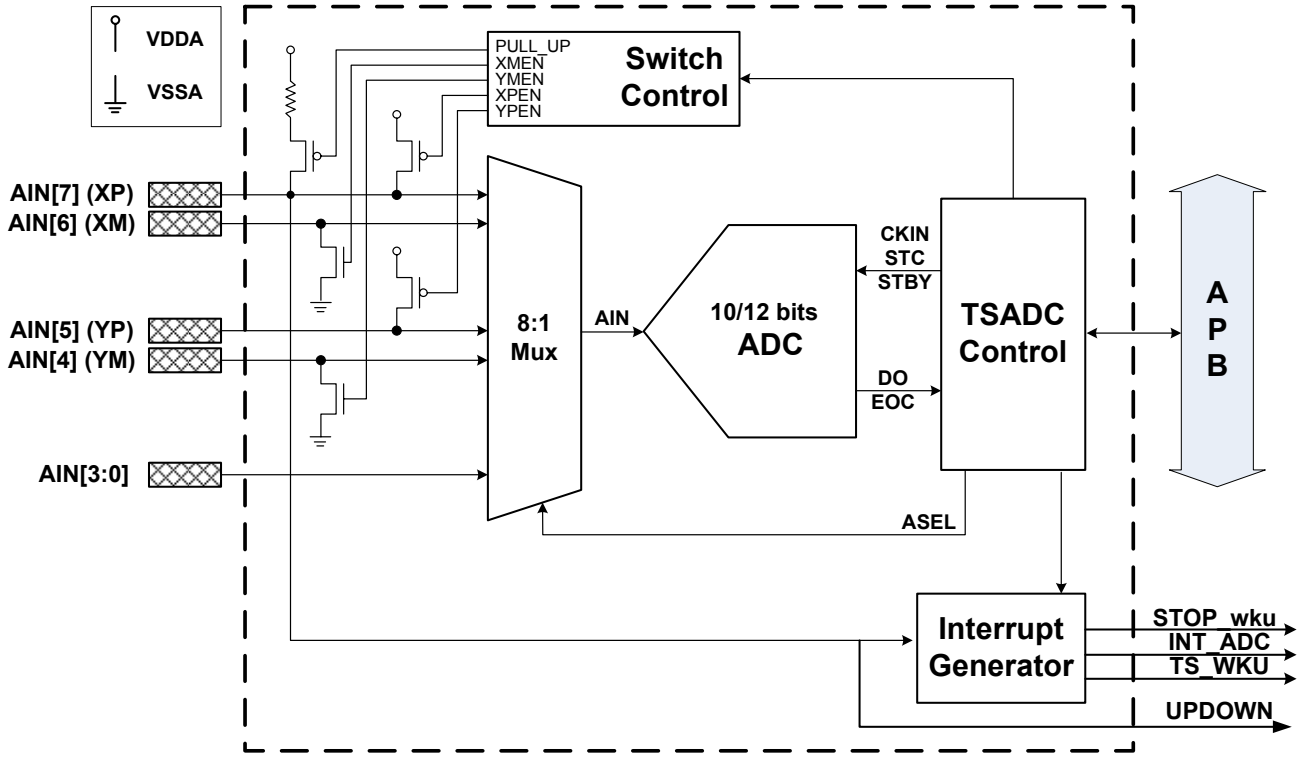


Figure 24.1 ADC Controller Block Diagram

TSADC interface have touch screen interface(4-wire) and 4 normal analog input. If touch screen is not used, touch screen interface is used as normal analog input. The maximum operating frequency of APB interface(PCLK) is 50 MHz. ADC clock(CKIN) is generated by PCLK. The maximum operating frequency of CKIN is 5 MHz(1 MSPS).

(CKIN should be set less than PCLK by 5 times)

When UDEN (ADCCON) is 0, interrupt source is pen UP signal. When UDEN (ADCCON) is 1, interrupt source is pen DOWN signal. Because UPDOWN interrupt signal keep alive on power down mode, it is used of power wake-up signal on power down mode.

X-position conversion data is inserted to ADCDAT0, Y-position data is inserted to ADCDAT1.

Also TSADC control signal is controlled directly by GPIO.

24.1.1 A/D Conversion Time

When the PCLK freq. is 50 MHz and the prescaler value is 49, total 10-bit/12-bit conversion time is as follows.

ADC freq = 50 MHz / (49 + 1) = 1 MHz

Conversion time = 1 / (1 MHz / 5 cycles) = 1 / 200 KHz = 5 us

Note 1: Maximum freq of PCLK is 50MHz.

Note 2: This A/D converter was designed to operate at maximum 5 MHz clock, so the conversion rate can go up to 1 MSPS.

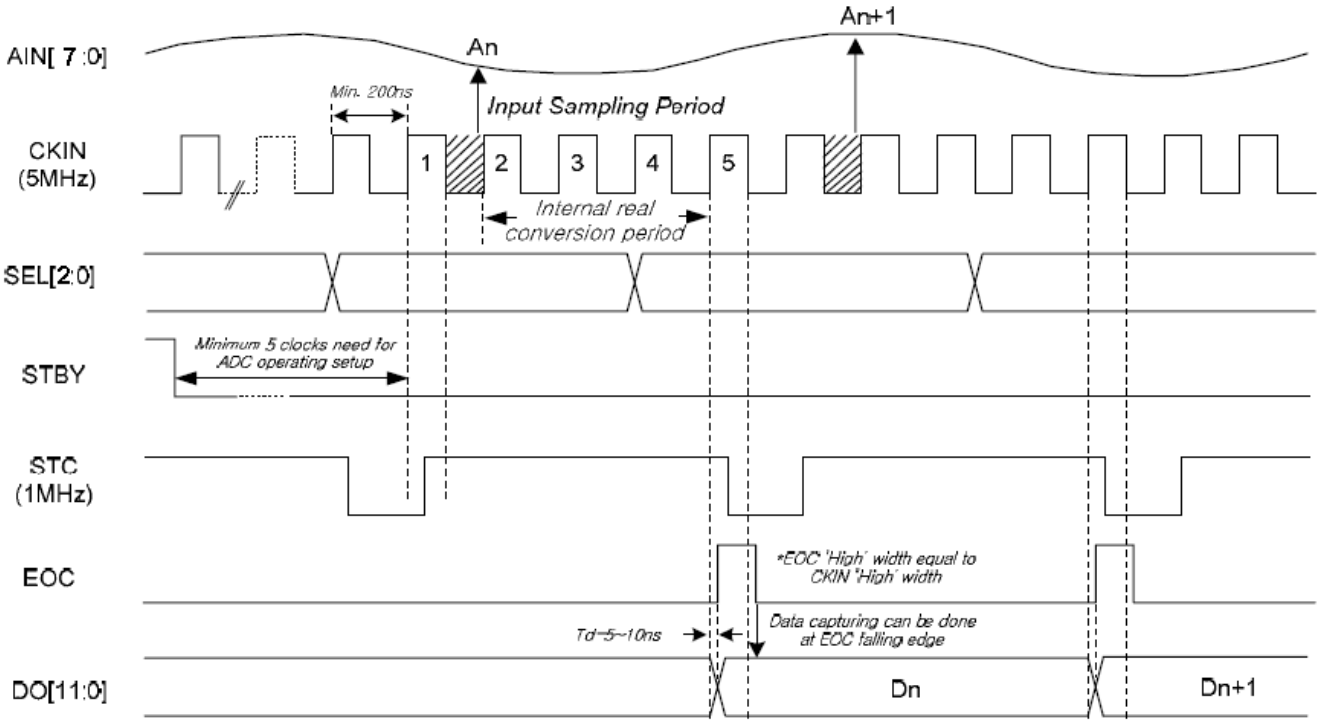


Figure 24.2 Main waveform

24.1.2 Touch Screen Interface Mode

24.1.2.1 Normal Conversion Mode (AUTO = 0, XY_PST = 0)

The operation of this mode is identical with AIN[0] ~ AIN[3]. It can be initialized by setting the ADCCON and ADCTSC. All of the switches and pull-up resister should be turned off(reset value 0x58 makes switches turn-off). The converted data can be read out from ADCDAT0 (A/D Conversion data 0 register).

24.1.2.2 Separate X/Y position Conversion Mode (AUTO = 0, XY_PST = control)

This mode consists of two states. One is X-position(X-pos.) measurement and the other is Y-position(Y-pos.) measurement state.

X-pos. conversion state is operated as the following way. Set XY_PST is '10b' and read out the converted data from ADCDAT0. When XY_PST is '01b', XP and XM switch is automatically on and ADC channel selection bits are automatically changed to '101b'. The end of X-pos. conversion can be notified by interrupt(INT_ADC).

Y-pos. conversion state is operated as the following way. Set XY_PST is '10b' and read out the converted data from ADCDAT1. When XY_PST is '10b', YP and YM switch is automatically on and ADC channel selection bits are automatically changed to '111b'. The end of Y-pos. conversion can be notified by interrupt(INT_ADC).

24.1.2.3 Auto(Sequential) X/Y position Conversion Mode (AUTO = 1, XY_PST = 0)

Auto(Sequential) X/Y Conversion mode is operated in the following method. Touch screen controller sequentially converts X-pos. and Y-pos. that is touched. After Touch Screen controller writes X-pos. data to ADCDAT0 and writes Y-pos. data to ADCDAT1, Touch screen interface generates Interrupt(INT_ADC). The conversion states are automatically changed. When X-pos is detected, XP and XM switch is automatically on and ADC channel selection bits are automatically changed to '101b'. After auto X/Y position conversion, mode is changed for pull-up interrupt detection(ADCTSC=0x173).

24.1.2.4 Waiting Interrupt Mode (ADCTSC = 0x00D3)

Touch screen controller generates an interrupt signal(INT_UPDN) when the stylus pen is down or up. The value of ADCTSC is '0x00D3' , P_UP is '0', XPEN is '1', XMEN is '0', YPEN is '1' and YMEN is '1'.

After touch screen controller generates interrupt signal, waiting interrupt mode must be cleared. (XY_PST sets to the No operation mode)

24.1.2.5 Stand-By Mode

Standby mode is activated when ADCCON[2] (STBY) is set to '1', In this mode, A/D Conversion operation is halted and ADCDAT0,ADCDAT1 register contains the previous converted data.

24.2 TSADC Controller Register Description

Table 24.1 TSADC Controller Register Map (Base Address = 0xF05F4000)

Name	Address	Type	Reset	Description
ADCCON	0x00	R/W	0x00003FC4	ADC Control Register
ADCTSC	0x04	R/W	0x00000058	ADC Touch Screen Control Register
ADCDLY	0x08	R/W	0x000000FF	ADC Start or Interval Delay Register
ADCDAT0	0x0C	R	-	ADC Conversion Data Register
ADCDAT1	0x10	R	-	ADC Conversion Data Register
ADCUPDN	0x14	R/W	0x00000000	Stylus Up or Down Interrupt Register
ADCCLRINT	0x18	W	-	Clear ADC Interrrupt
Reserved	0x1C	-	-	Reserved
ADCCLRUPDN	0x20	W	-	Clear Pen UP/DOWN Interrupt

ADC Control Register (ADCCON) 0xF05F4000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															RES
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
E_FLG	PS_EN	PS_VAL								ASEL			STBY	RD_ST	EN_ST

Field	Name	RW	Reset	Description
16	RES	RW	0x0	ADC Resolution Selection 0 = 10 bits A/D Conversion 1 = 12 bits A/D Conversion
15	E_FLG	R	0x0	End of Conversion(EOC) Flag 0 = A/D Conversion in Progress 1 = End of A/D Conversion
14	PS_EN	RW	0x0	Enable Prescaler 0 = Disable 1 = Enable
13 ~ 6	PS_VAL	RW	0xFF	Value of Prescaler Data value : 4 ~ 255 Note that division factor is (N+1) when the prescaler value is N. Note: ADC frequency should be set less than PCLK by 5 times. (ex) PCLK=10MHz, ADC Freq<2MHz)
5 ~ 3	ASEL	RW	0x0	Analog Input Channel Select 3'b000 = AIN0 3'b001 = AIN1 3'b010 = AIN2 3'b011 = AIN3 3'b100 = AIN4(YM) 3'b101 = AIN5(YP) 3'b110 = AIN6(XM) 3'b111 = AIN7(XP)
2	STBY	RW	0x1	Stand-By mode selection 0 = Operation mode 1 = Stand-By mode
1	RD_ST	RW	0x0	A/D Conversion Start by data read 0 = Disable 1 = Enable
0	EN_ST	RW	0x0	A/D Conversion Start by Eanble signal 0 = Disable 1 = Enable

ADC Touch Screen Register (ADCTSC) 0xF05F4004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							UDEN	YMEN	YPEN	XMEN	XPEN	P_UP	AUTO	XY_PST	

While waiting for stylus up/down interrupt, XPEN bit must be set to '1', namely 'XP output disable' and P_UP bit must be set to '0', namely 'XP Pull-Up Enable'.

AUTO_PST bit should be set '1' only in Automatic and Sequential X/Y Position Conversion.

If you don't use touch screen, you should not tie AIN[6](XM) and AIN[4](YM) to VSSA. Because Pull-Up switch, XP and YP switch are enable in SLEEP mode.

Field	Name	RW	Reset	Description
8	UDEN	RW	0x0	Detect Stylus UP / DOWN status 0 = Detect Stylus-DOWN status 1 = Detect Stylus-UP status
7	YMEN	RW	0x0	YM to GND Switch Enable 0 = Disable (YM = AIN4, Hi-z) 1 = Enable (YM = VSSA)
6	YPEN	RW	0x1	YP to VDD Switch Enable 0 = Enable (YP = VDDA) 1 = Disable (YP = AIN5, Hi-z)
5	XMEN	RW	0x0	XM to GND Switch Enable 0 = Disable (XM = AIN6, Hi-z) 1 = Enable (XM = VSSA)
4	XPEN	RW	0x1	XP to VDD Switch Enable 0 = Enable (XP = VDDA) 1 = Disable (XP = AIN7, Hi-z)
3	P_UP	RW	0x1	Pull-Up Switch Enable 0 = XP Pull-up Enable 1 = XP Pull-up Disable
2	AUTO	RW	0x0	Automatic(sequencial) conversion of X-pos. and Y-pos 0 = Normal A/D conversion 1 = Automatic(sequencial) conversion of X-pos. and Y-pos
1 ~ 0	XYPST	RW	0x0	Select conversion of X-pos or Y-pos 2'b00 = No operation 2'b01 = X-pos conversion 2'b10 = Y-pos conversion 2'b11 = Waiting Interrupt mode

ADC Start Delay Register (ADCDLY) 0xF05F4008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															CLKsrc
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DELAY															

Before A/D Conversion, touch screen uses X-tal clock(3.68 MHz)

During A/D Conversion, PCLK(Max, 50 MHz) is used.

Field	Name	RW	Reset	Description
16	CLKsrc	R/W	0	ADCDLY clock source In waiting for interrupt mode, CLKsrc is used as delay filter clock source. 0 = External input clock 1 = RTC clock
15 ~ 0	DELAY	R/W	0xFF	1. In case of A/D conversion mode (normal, separate, auto) : A/D conversion is delayed by counting clock is PCLK. -> A/D conversion delay value 2. In case of waiting interrupt mode : when stylus down occurs in waiting interrupt mode, it generates interrupt signal(INT_PENUPDN) at interval of several ms for AUTO X/Y-pos conversion. If this interrupt occurs in STOP mode, it generates Wake-Up signal, having interval (several ms), for exiting STOP MODE.

ADC Data0 Register (ADCDAT0) 0xF05F400C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UPDN	AUTO	XYPST		XPDATA_12		XPDATA									

Field	Name	RW	Reset	Description
15	UPDN	R	-	UP/DOWN state of Stylus in Waiting Interrupt Mode. 0 = Stylus-DOWN 1 = Stylus-UP
14	AUTO	R	-	Automatic sequencing conversion of X-pos. and Y-pos. 0 = Normal A/D Conversion 1 = Automatic (sequencial) conversion of X-pos, Y-pos
13 ~ 12	XYPST	R	-	Select conversion of X-pos. or Y-pos. 2'b00 = No operation. 2'b01 = X-pos conversion. 2'b10 = Y-pos conversion. 2'b11 = Waiting Interrupt mode
11 ~ 10	XPDATA_12	R	-	When A/D resolution is 12bits, this is X-pos. conversion data[11:0] value.
9 ~ 0	XPDATA	R	-	X-pos. conversion data[9:0] value (Includes normal A/D Conversion data value) Data value : 0x00 ~ 0x3FF

ADC Data1 Register (ADCDAT1) 0xF05F4010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UPDN	AUTO	XY_PST		XPDATA_12		XPDATA									

Field	Name	RW	Reset	Description
15	UPDN	R	-	Select UP/DOWN state of Stylus in Waiting Interrupt mode. 0 = Stylus-DOWN 1 = Stylus-UP
14	AUTO	R	-	Automatic(sequencial) conversion of X-pos. and Y-pos. 0 = Normal A/D conversion 1 = Sequencial conversion of X-pos, Y-pos

13 ~ 12	XY_PST	R	-	Select conversion of X-pos. or Y-pos. 2'b00 = No operation 2'b01 = X-pos conversion 2'b10 = Y-pos conversion 2'b11 = Waiting Interrupt mode
11 ~ 10	XPDATA_12	R	-	When A/D resolution is 12bits, this is X-pos conversion data[11:0] value.
9 ~ 0	XPDATA	R	-	X-pos. conversion data[9:0] value (Includes normal A/D conversion data value) Data value : 0x00 ~ 0x3FF

ADC Touch Screen UP/DOWN Register (ADCUPDN) **0xF05F4014**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														UP	DOWN

Field	Name	RW	Reset	Description
1	UP	RW	0x0	Stylus-UP Interrupt history (***After check, this bit must be cleared manually***) 0 = No Stylus-UP Interrupt. 1 = Stylus-UP Interrupt has been occurred.
0	DOWN	RW	0x0	Stylus-DOWN Interrupt history. (***After check, this bit should be cleared manually***) 0 = No Stylus-DOWN Interrupt 1 = Stylus-DOWN Interrupt has been occurred.

ADC Touch Screen Interrupt Clear Register (ADCCLRINT) **0xF05F4018**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															INT_CLR

ISR(Interrupt Service Routine) is responsible for clearing interrupts after the interruptservice is completed. Writing any values on this register will clear up the relevant interrupts asserted.

Field	Name	RW	Reset	Description
0	INT_CLR	W	-	Clear the INT_ADC Interrupt

ADC Touch Screen Interrupt Clear Register (ADCCLRUPDN) **0xF05F4020**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UPDN_CLR

ISR(Interrupt Service Routine) is responsible for clearing interrupts after the interrupt service is completed. Writing any values on this register will clear up the relevant interrupts asserted.

Field	Name	RW	Reset	Description
0	UPDN_CLR	W	-	Clear the Stylus UP/DOWN Interrupt

24.3 Programing Note

The converted data can be accessed by means of interrupt or polling method. With interuupt method, the overall conversion time – from A/D converter start to convert data read – may be delayed because of the return time of interrupt service routine and data access time. With polling method, by checking the ADCCON[15] – end of conversion flag – bit, the read time from ADCDAT register can be determined.

A/D conversion can be activated in different way. After ADCCON[1] – A/D conversion start-by-read mode – is set to 1. A/D conversion starts simultaneously when converted data is read.

If pen up/down interrupt is used as an wake-up source in STOP mode, XY_PST bit(ADCTSC[1:0]) should be set to 2'b11(waiting interrupt mode). To choose stylus pen up/down wake-up, UD_SEN bit(ADCTSC[8]) is used.

[Input range]

The analog input is single-ended type and the range is from VREF to AGND. This analog input voltage follows reference voltage range fundamentally. So, if you want to alter into another input range, you should change the voltage value of VREF.

Table 24.2 I/O Chart

Index	AIN input	Digital output	YPEN
0	~ 0.0008056	0000 0000 0000	1 LSB = 1.805 mV VREF = 3.3 V AGND = 0.0 V (at 3.3 V typical supply voltage)
1	0.0008056 ~ 0.0016112	0000 0000 0001	
~	~	~	
2047	1.6491944 ~ 1.65000	0111 1111 1111	
2048	1.6500000 ~ 1.6508056	1000 0000 0000	
2049	1.6508056 ~ 1.6516112	1000 0000 0001	
~		~	
	3.2983888 ~ 3.2991944	1111 1111 1110	
	3.2991944 ~ 3.3	1111 1111 1111	

[Analog Input Selection Table for 4-wire touch screen panels]

Five different functions are defined as shown in the following table. These functions require control of the switches in ADC as well as the interrupt function. After ADC generates the interrupt signal(UPDOWN), waiting for interrupt mode must be cleared by setting to the No operation mode.

Table 24.3 Analog Input Selection Table for 4-wire touch screen panels

XY_PST	XPEN	XMEN	YPEN	YMEN	P_UP	SEL[2:0]
X position measurement	0	1	1	0	1	101(YP)
Y position measurement	1	0	0	1	1	111(XP)
No operation (Normal ADC)	1	0	1	0	1	AIN port number
Waiting Interrupt (for detecting pen up/down)	1	0	1	1	0	-
Auto measurement	auto	auto	auto	auto	auto	auto

For touch screen function, it is recommended to follow the below sequence.

1. Waiting for interrupt mode (for detecting pen up/down)
2. After receiving the interrupt signal, setting to no operation mode to clear waiting for interrupt mode
3. X position measurement (after several measurement, averaging over those results)
4. Y position measurement (the same method as X position measurement)
5. Waiting for interrupt mode (for detecting pen up/down)

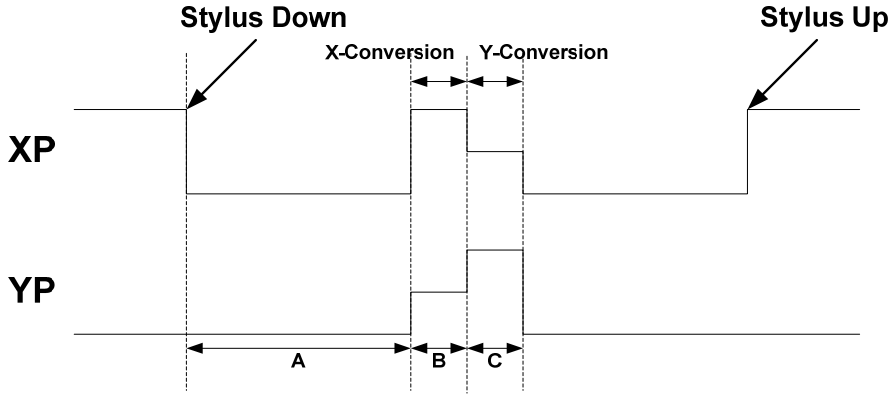


Figure 5.2 Operation Signal

A = D x (1/rtc clock) or D x (1/X-tal clock) or D x (1/EXTCLK clock)
B = D x (1/PCLK clock) + 5 x (1/PCLK clock) x (PRSCVL + 1)
C = D x (1/PCLK clock) + 5 x (1/PCLK clock) x (PRSCVL + 1)

D = Delay value of ADCDLY register

[Analog Input Selection Table for Pressure Measurement]

Touch pressure can also be measured with this IP. Since the contact resistance between the X and Y plate de-creases as the pressure increases, the pressure of the touch screen can be determined by measuring it. To cal-culate the contact resistance, the method requires knowing the X-plate resistance, measurement of the X-position, and two additional cross panel measurement of the touch screen. The following table shows the con-figuration for pressure measurement using a 4-wire touch screen.

Table 24.4 Analog Input Selection Table for Pressure Measurement

XY_PST	XPEN	XMEN	YPEN	YMEN	P_UP	SEL[3:0]
X position measurement	0	1	1	0	1	101(YP)
XP measurement	1	1	0	0	1	111(XP)
YM measurement	1	1	0	0	1	100(YM)

$$R_{contact} = R_{x-plate} \frac{X_{position}}{2^{12}} \left(\frac{YM_{measurement}}{XP_{measurement}} - 1 \right)$$

25 ECC

25.1 Functional Description

The ECC (Error Correction Code) is used to correct data error in storage device or various kind of communicating system. The TCC8900 has simple ECC generation and Error Correction Module. By enable ECC module, it consistently monitors internal bus activity and calculate ECC whenever there is read or write cycle from/to a predefined memory area. The area can be determined by special Register so this module can be used ECC calculation itself not only for specific storage device such as NAND flash.

The following figure represents block diagram including internal bus connection for ECC module.

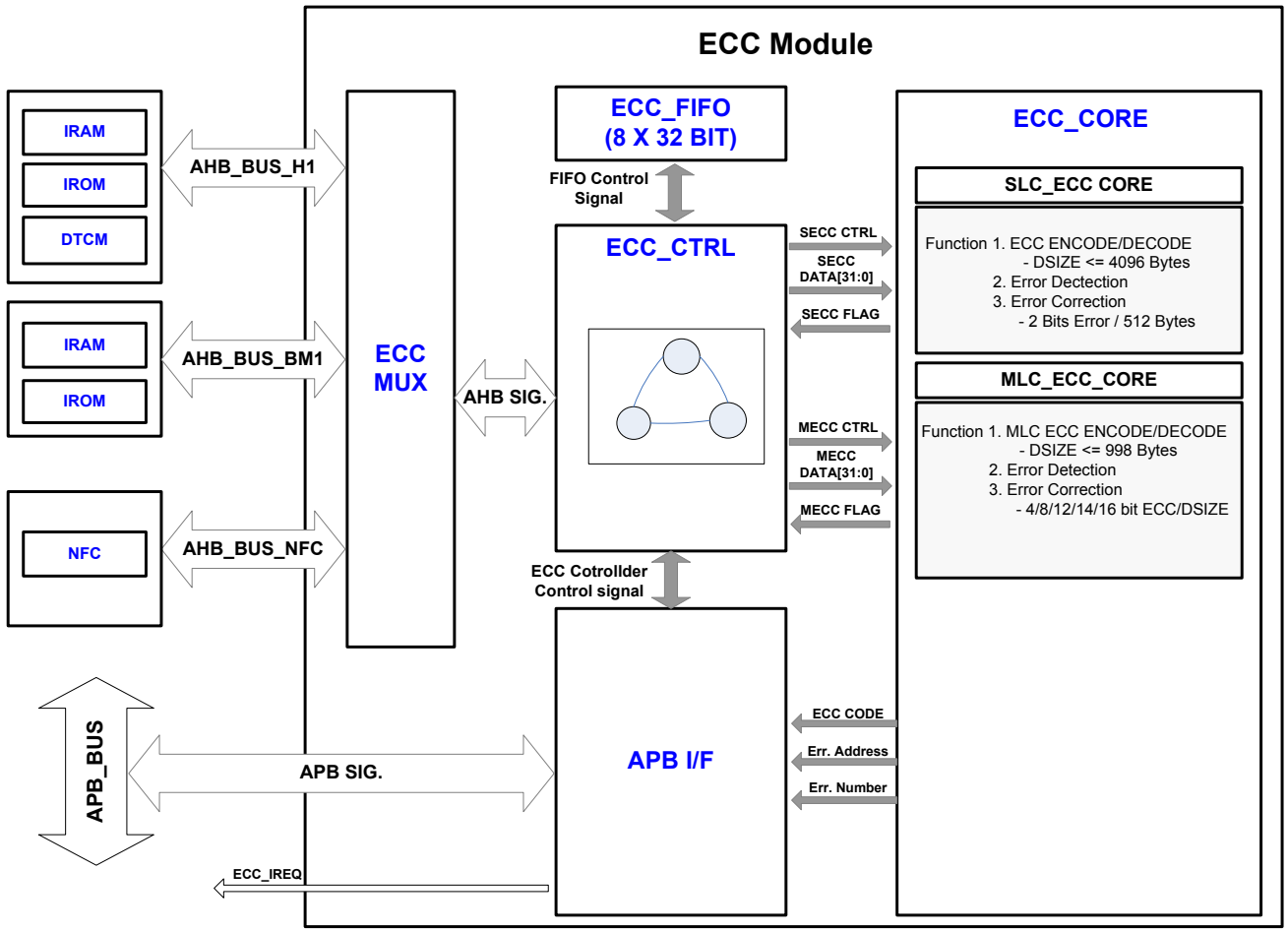


Figure 25.1 ECC Block Diagram

25.2 Register Description

Table 25.1 ECC Register Map (ECC BASE Address=0xF0539000)

Name	Address	Type	Reset	Description
ECC_CTRL	0x00	R/W	0x00000000	ECC Control Register
ECC_BASE	0x04	R/W	0x00000000	Base Address for ECC Calculation
ECC_MASK	0x08	R/W	0x00000000	Address mask for ECC area.
ECC_CLEAR	0x0C	R/W		ECC Clear
SLC ECC Code Register Set				
SECC_0	0x10	R/W	0x00000000	1st SLC ECC Code Register
SECC_1	0x14	R/W	0x00000000	2nd SLC ECC Code Register
SECC_2	0x18	R/W	0x00000000	3rd SLC ECC Code Register
SECC_3	0x1C	R/W	0x00000000	4th SLC ECC Code Register
SECC_4	0x20	R/W	0x00000000	5th SLC ECC Code Register
SECC_5	0x24	R/W	0x00000000	6th SLC ECC Code Register
SECC_6	0x28	R/W	0x00000000	7th SLC ECC Code Register
SECC_7	0x2C	R/W	0x00000000	8th SLC ECC Code Register
SECC_8	0x30	R/W	0x00000000	9th SLC ECC Code Register
SECC_9	0x34	R/W	0x00000000	10th SLC ECC Code Register
SECC_10	0x38	R/W	0x00000000	11th SLC ECC Code Register
SECC_11	0x3C	R/W	0x00000000	12th SLC ECC Code Register
SECC_12	0x40	R/W	0x00000000	13th SLC ECC Code Register
SECC_13	0x44	R/W	0x00000000	14th SLC ECC Code Register
SECC_14	0x48	R/W	0x00000000	15th SLC ECC Code Register
SECC_15	0x4C	R/W	0x00000000	16th SLC ECC Code Register
MLC ECC4 Code Register Set				
MECC4_0	0x10	R/W	0x00000000	1st MLC ECC4 Code Register
MECC4_1	0x14	R/W	0x00000000	2nd MLC ECC4 Code Register
MLC ECC8 Code Register Set				
MECC8_0	0x10	R/W	0x00000000	1st MLC ECC8 Code Register
MECC8_1	0x14	R/W	0x00000000	2nd MLC ECC8 Code Register
MECC8_2	0x18	R/W	0x00000000	3rd MLC ECC8 Code Register
MECC8_3	0x1C	R/W	0x00000000	4th MLC ECC8 Code Register
MLC ECC12 Code Register Set				
MECC12_0	0x10	R/W	0x00000000	1st MLC ECC12 Code Register
MECC12_1	0x14	R/W	0x00000000	2nd MLC ECC12 Code Register
MECC12_2	0x18	R/W	0x00000000	3rd MLC ECC12 Code Register
MECC12_3	0x1C	R/W	0x00000000	4th MLC ECC12 Code Register
MECC12_4	0x20	R/W	0x00000000	5th MLC ECC12 Code Register
MLC ECC14 Code Register Set				
MECC14_0	0x10	R/W	0x00000000	1st MLC ECC14 Code Register
MECC14_1	0x14	R/W	0x00000000	2nd MLC ECC14 Code Register
MECC14_2	0x18	R/W	0x00000000	3rd MLC ECC14 Code Register
MECC14_3	0x1C	R/W	0x00000000	4th MLC ECC14 Code Register
MECC14_4	0x20	R/W	0x00000000	5th MLC ECC14 Code Register
MECC14_5	0x24	R/W	0x00000000	6th MLC ECC14 Code Register
MLC ECC16 Code Register Set				
MECC16_0	0x10	R/W	0x00000000	1st MLC ECC16 Code Register
MECC16_1	0x14	R/W	0x00000000	2nd MLC ECC16 Code Register
MECC16_2	0x18	R/W	0x00000000	3rd MLC ECC16 Code Register
MECC16_3	0x1C	R/W	0x00000000	4th MLC ECC16 Code Register
MECC16_4	0x20	R/W	0x00000000	5th MLC ECC16 Code Register
MECC16_5	0x24	R/W	0x00000000	6th MLC ECC16 Code Register
MECC16_6	0x28	R/W	0x00000000	7th MLC ECC16 Code Register
SLC ECC Error Address Register Set				
SECC_EADDR0	0x50	R	0x00000000	SLC ECC Error Address Register0
SECC_EADDR1	0x54	R	0x00000000	SLC ECC Error Address Register1
SECC_EADDR2	0x58	R	0x00000000	SLC ECC Error Address Register2
SECC_EADDR3	0x5C	R	0x00000000	SLC ECC Error Address Register3
SECC_EADDR4	0x60	R	0x00000000	SLC ECC Error Address Register4
SECC_EADDR5	0x64	R	0x00000000	SLC ECC Error Address Register5
SECC_EADDR6	0x68	R	0x00000000	SLC ECC Error Address Register6
SECC_EADDR7	0x6C	R	0x00000000	SLC ECC Error Address Register7

SECC_EADDR8	0x70	R	0x00000000	SLC ECC Error Address Register8
SECC_EADDR9	0x74	R	0x00000000	SLC ECC Error Address Register9
SECC_EADDR10	0x78	R	0x00000000	SLC ECC Error Address Register10
SECC_EADDR11	0x7C	R	0x00000000	SLC ECC Error Address Register11
SECC_EADDR12	0x80	R	0x00000000	SLC ECC Error Address Register12
SECC_EADDR13	0x84	R	0x00000000	SLC ECC Error Address Register13
SECC_EADDR14	0x88	R	0x00000000	SLC ECC Error Address Register14
SECC_EADDR15	0x8C	R	0x00000000	SLC ECC Error Address Register15
MLC ECC4 Error Address Register Set				
MECC4_EADDR0	0x50	R	0x00000000	MLC ECC Error Address Register0
MECC4_EADDR1	0x54	R	0x00000000	MLC ECC Error Address Register1
MECC4_EADDR2	0x58	R	0x00000000	MLC ECC Error Address Register2
MECC4_EADDR3	0x5C	R	0x00000000	MLC ECC Error Address Register3
MLC ECC8 Error Address Register Set				
MECC8_EADDR0	0x50	R	0x00000000	MLC ECC8 Error Address Register0
MECC8_EADDR1	0x54	R	0x00000000	MLC ECC8 Error Address Register1
MECC8_EADDR2	0x58	R	0x00000000	MLC ECC8 Error Address Register2
MECC8_EADDR3	0x5C	R	0x00000000	MLC ECC8 Error Address Register3
MECC8_EADDR4	0x60	R	0x00000000	MLC ECC8 Error Address Register4
MECC8_EADDR5	0x64	R	0x00000000	MLC ECC8 Error Address Register5
MECC8_EADDR6	0x68	R	0x00000000	MLC ECC8 Error Address Register6
MECC8_EADDR7	0x6C	R	0x00000000	MLC ECC8 Error Address Register7
MLC ECC12 Error Address Register Set				
MECC12_EADDR0	0x50	R	0x00000000	MLC ECC12 Error Address Register0
MECC12_EADDR1	0x54	R	0x00000000	MLC ECC12 Error Address Register1
MECC12_EADDR2	0x58	R	0x00000000	MLC ECC12 Error Address Register2
MECC12_EADDR3	0x5C	R	0x00000000	MLC ECC12 Error Address Register3
MECC12_EADDR4	0x60	R	0x00000000	MLC ECC12 Error Address Register4
MECC12_EADDR5	0x64	R	0x00000000	MLC ECC12 Error Address Register5
MECC12_EADDR6	0x68	R	0x00000000	MLC ECC12 Error Address Register6
MECC12_EADDR7	0x6C	R	0x00000000	MLC ECC12 Error Address Register7
MECC12_EADDR8	0x70	R	0x00000000	MLC ECC12 Error Address Register8
MECC12_EADDR9	0x74	R	0x00000000	MLC ECC12 Error Address Register9
MECC12_EADDR10	0x78	R	0x00000000	MLC ECC12 Error Address Register10
MECC12_EADDR11	0x7C	R	0x00000000	MLC ECC12 Error Address Register11
MLC ECC14 Error Address Register Set				
MECC14_EADDR0	0x50	R	0x00000000	MLC ECC14 Error Address Register0
MECC14_EADDR1	0x54	R	0x00000000	MLC ECC14 Error Address Register1
MECC14_EADDR2	0x58	R	0x00000000	MLC ECC14 Error Address Register2
MECC14_EADDR3	0x5C	R	0x00000000	MLC ECC14 Error Address Register3
MECC14_EADDR4	0x60	R	0x00000000	MLC ECC14 Error Address Register4
MECC14_EADDR5	0x64	R	0x00000000	MLC ECC14 Error Address Register5
MECC14_EADDR6	0x68	R	0x00000000	MLC ECC14 Error Address Register6
MECC14_EADDR7	0x6C	R	0x00000000	MLC ECC14 Error Address Register7
MECC14_EADDR8	0x70	R	0x00000000	MLC ECC14 Error Address Register8
MECC14_EADDR9	0x74	R	0x00000000	MLC ECC14 Error Address Register9
MECC14_EADDR10	0x78	R	0x00000000	MLC ECC14 Error Address Register10
MECC14_EADDR11	0x7C	R	0x00000000	MLC ECC14 Error Address Register11
MECC14_EADDR12	0x80	R	0x00000000	MLC ECC14 Error Address Register12
MECC14_EADDR13	0x84	R	0x00000000	MLC ECC14 Error Address Register13
MLC ECC16 Error Address Register Set				
MECC16_EADDR0	0x50	R	0x00000000	MLC ECC16 Error Address Register0
MECC16_EADDR1	0x54	R	0x00000000	MLC ECC16 Error Address Register1
MECC16_EADDR2	0x58	R	0x00000000	MLC ECC16 Error Address Register2
MECC16_EADDR3	0x5C	R	0x00000000	MLC ECC16 Error Address Register3
MECC16_EADDR4	0x60	R	0x00000000	MLC ECC16 Error Address Register4
MECC16_EADDR5	0x64	R	0x00000000	MLC ECC16 Error Address Register5
MECC16_EADDR6	0x68	R	0x00000000	MLC ECC16 Error Address Register6
MECC16_EADDR7	0x6C	R	0x00000000	MLC ECC16 Error Address Register7
MECC16_EADDR8	0x70	R	0x00000000	MLC ECC16 Error Address Register8
MECC16_EADDR9	0x74	R	0x00000000	MLC ECC16 Error Address Register9
MECC16_EADDR10	0x78	R	0x00000000	MLC ECC16 Error Address Register10
MECC16_EADDR11	0x7C	R	0x00000000	MLC ECC16 Error Address Register11
MECC16_EADDR12	0x80	R	0x00000000	MLC ECC16 Error Address Register12
MECC16_EADDR13	0x84	R	0x00000000	MLC ECC16 Error Address Register13

MECC16_EADDR14	0x88	R	0x00000000	MLC ECC16 Error Address Register14
MECC16_EADDR15	0x8C	R	0x00000000	MLC ECC16 Error Address Register15
ERRNUM	0x90	R	0x00000000	ECC Error Number
ECC_IREQ	0x94	R/W	0x00000000	ECC Interrupt Control Register

ECC Control Register (ECC_CTRL) 0xF0539000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											DIEN16	DIEN14	DIEN12	DIEN8	DIEN4
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	DSIZE[10:00]											EN[3:0]			

DIEN16	[20]	MLC ECC16 Decoding Interrupt Enable
0	MLC ECC16 Decoding Interrupt Disable	
1	MLC ECC16 Decoding Interrupt Enable	

DIEN14	[19]	MLC ECC14 Decoding Interrupt Enable
0	MLC ECC14 Decoding Interrupt Disable	
1	MLC ECC14 Decoding Interrupt Enable	

DIEN12	[18]	MLC ECC12 Decoding Interrupt Enable
0	MLC ECC12 Decoding Interrupt Disable	
1	MLC ECC12 Decoding Interrupt Enable	

DIEN8	[17]	MLC ECC8 Decoding Interrupt Enable
0	MLC ECC8 Decoding Interrupt Disable	
1	MLC ECC8 Decoding Interrupt Enable	

DIEN4	[16]	MLC ECC4 Decoding Interrupt Enable
0	MLC ECC8 Decoding Interrupt Disable	
1	MLC ECC8 Decoding Interrupt Enable	

Note: After MLC ECC4/8/12/14/16 decoding interrupt, ERRNUM Register can be used to detect data error, and it is able to correct data error by ERRADDRx.

If a decoding interrupt enable bit of ECC_CTRL register is set and MLC ECC decoding operation is finished, ECC interrupt is really generated and user can clear interrupt request register by writing “1” into FLG field or IREQ field.

DATASIZE	[14:04]	ECC Data Size
N	SLC ECC	0 < N <= 1024 (word size) & N=multiples of 64
N	MLC ECC	0 < N <= 998 (byte size)

Note: In the case of SLC ECC, N is written with Word Data Size and it can be set up to 4096 bytes in multiples of 256 byte unit.

Note: In the case of MLC ECC4/8/12/14/16, N is written with byte data size and it can be set up to 1004 bytes in 1 byte unit.

EN	[03:00]	ECC Enable Control
4’b00xx	ECC Disable	
4’b0100	SLC ECC Encoding Enable	
4’b0101	SLC ECC Decoding Enable	
4’b0110	MLC ECC4 Encoding Enable	
4’b0111	MLC ECC4 Decoding Enable	
4’b1000	MLC ECC8 Encoding Enable	
4’b1001	MLC ECC8 Decoding Enable	
4’b1010	MLC ECC12 Encoding Enable	
4’b1011	MLC ECC12 Decoding Enable	
4’b1100	MLC ECC14 Encoding Enable	
4’b1101	MLC ECC14 Decoding Enable	
4’b1110	MLC ECC16 Encoding Enable	
4’b1111	MLC ECC16 Decoding Enable	

ECC Base Address Register (ECC_BASE) 0xF0539004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECC_BASE[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ECC_BASE[15:2]														0	

ECC Address Mask Register (ECC_MASK) 0xF0539008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				ECC_MASK[27:16]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ECC_MASK[15:2]														0	

Note: The ECC is calculated whenever the specified region of memory is accessed. The region for ECC calculating is determined by ECC_BASE & ECC_MASK Register. The real base address is determined by following formula.

Real base address = ECC_BASE(0xF005B004) & ~(ECC_MASK[27:2] << 2)

(The real base address is assumed to be word aligned, so the least 2 bits are always 0.)

The size of region is also determined by ECC_MASK Register. If ECC_MASK Register has N concatenated 0 from LSB, the region size is set to 2^N bytes.

ECC Clear Register (ECC_CLR) 0xF053900C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Don’t care															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Don’t care															

Note: Whenever this Register is written by any value, SLC/MLC Block are cleared. Before ECC Encoding and Decoding, SLC/MLC Block must be cleared.

SLC ECC Code Register (SECC_n, n = 0,...,15) ECC BASE10 + n*4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								SLC_ECCn_0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLC_ECCn_1								SLC_ECCn_2							

Note: These Registers contain ECC output for SLC. It calculates ECC of SSFDC standard, and can contain up to 8 blocks of data.

For each output Register, there are a total of 22 bits of parity data (6 bits for column parity and 16 bits for line parity) as follows:

P1, P1', P2, P2', P4, P4', P8, P8', P16, P16', ……, P1024, P1024'

The parity data that have been generated are stored as follows.

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SLC_ECC0_0	P64	P64'	P32	P32'	P16	P16'	P8	P8'
SLC_ECC0_1	P1024	P1024'	P512	P512'	P256	P256'	P128	P128'
SLC_ECC0_2	P4	P4'	P2	P2'	P1	P1'	1	1

Note: To correct error for SLC ECC, writing original ECC data that read from nand flash to SLC_ECCx Register.

For SLC ECC

Address	512 Byte	…	2048 Byte	4096 Byte
0xF0539010	SECC[047:024]		SECC[191:168]	SECC[383:360]
0xF0539014	SECC[023:000]		SECC[167:144]	SECC[359:336]
0xF0539018			SECC[143:120]	SECC[335:312]
0xF053901C			SECC[119:096]	SECC[311:288]
0xF0539020			SECC[095:072]	SECC[287:264]
0xF0539024			SECC[071:048]	SECC[263:240]
0xF0539028			SECC[047:024]	SECC[239:216]
0xF053902C			SECC[023:000]	SECC[215:192]
0xF0539030				SECC[191:168]
0xF0539034				SECC[167:144]
0xF0539038				SECC[143:120]
0xF053903C				SECC[119:096]
0xF0539040				SECC[095:072]
0xF0539044				SECC[071:048]
0xF0539048				SECC[047:024]
0xF053904C				SECC[023:000]

MLC ECC4 Code Register 0(MECC4_0) **ECC BASE + 0x10**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC4_CODE1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC4_CODE0[15:00]															

MLC ECC4 Code Register 1(MECC4_1) **ECC BASE + 0x14**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0												MECC4_CODE4[51:48]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC4_CODE3[47:32]															

Note: Real ECC Code Size for MLC ECC4 mode is 52 bits, For the remainder bits of MLC ECC4 Code Register 1, zeros are padded.

Note: MLC ECC4 Code Register must be written and read by word access.

MLC ECC8 Code Register 0(MECC8_0) **ECC BASE + 0x10**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC8_CODE1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC8_CODE0[15:00]															

MLC ECC8 Code Register 1(MECC8_1) **ECC BASE + 0x14**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC8_CODE3[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC8_CODE2[47:32]															

MLC ECC8 Code Register 2(MECC8_2) **ECC BASE + 0x18**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC8_CODE6[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC8_CODE5[79:64]															

MLC ECC8 Code Register 3(MECC8_3) **ECC BASE + 0x1C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								MECC8_CODE7[104:96]							

Note: Real ECC Code Size for MLC ECC8 mode is 104 bits, For the remainder bits of MLC ECC8 Code Register 3, zeros are padded.

Noe: MLC ECC8 Code Register must be written and read by word access.

MLC ECC12 Code Register 0(MECC12_0) **ECC BASE + 0x10**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC12_CODE1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC12_CODE0[15:00]															

MLC ECC12 Code Register 1(MECC12_1) **ECC BASE + 0x14**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC12_CODE3[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC12_CODE2[47:32]															

MLC ECC12 Code Register 2(MECC12_2) **ECC BASE + 0x18**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC12_CODE5[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC12_CODE4[79:64]															

MLC ECC12 Code Register 3(MECC12_3) **ECC BASE + 0x1C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC12_CODE7[127:112]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC12_CODE6[111:96]															

MLC ECC12 Code Register 4(MECC12_4) **ECC BASE + 0x20**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			MECC12_CODE9[155:144]												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC12_CODE8[143:128]															

Note: Real ECC Code Size for MLC ECC12 mode is 156 bits, For the remainder bits of MLC ECC12 Code Register 4, zeros are padded.

Note: MLC ECC12 Code Register must be written and read by word access.

MLC ECC14 Code Register 0(MECC14_0) **ECC BASE + 0x10**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC14_CODE1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC14_CODE0[15:00]															

MLC ECC14 Code Register 1(MECC14_1) **ECC BASE + 0x14**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC14_CODE3[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC14_CODE2[47:32]															

MLC ECC14 Code Register 2(MECC14_2) **ECC BASE + 0x18**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC14_CODE5[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC14_CODE4[79:64]															

MLC ECC14 Code Register 3(MECC14_3) **ECC BASE + 0x1C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC14_CODE7[127:112]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC14_CODE6[111:96]															

MLC ECC14 Code Register 4(MECC14_4) **ECC BASE + 0x20**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MECC14_CODE9[159:144]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MECC14_CODE8[143:128]															

MLC ECC14 Code Register 5(MECC14_5) **ECC BASE + 0x24**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

0								MECC14_CODE11[183:176]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MECC14_CODE10[175:160]															

Note: Real ECC Code Size for MLC ECC14 mode is 184 bits, For the remainder bits of MLC ECC14 Code Register 5, zeros are padded.

Note: MLC ECC14 Code Register must be written and read by word access.

MLC ECC16 Code Register 0(MECC16_0) **ECC BASE + 0x10**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MECC16_CODE1[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MECC16_CODE0[15:00]															

MLC ECC16 Code Register 1(MECC16_1) **ECC BASE + 0x14**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MECC16_CODE3[63:48]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MECC16_CODE2[47:32]															

MLC ECC16 Code Register 2(MECC16_2) **ECC BASE + 0x18**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MECC16_CODE5[95:80]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MECC16_CODE4[79:64]															

MLC ECC16 Code Register 3(MECC16_3) **ECC BASE + 0x1C**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MECC16_CODE7[127:112]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MECC16_CODE6[111:96]															

MLC ECC16 Code Register 4(MECC16_4) **ECC BASE + 0x20**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MECC16_CODE9[159:144]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MECC16_CODE8[143:128]															

MLC ECC16 Code Register 5(MECC16_5) **ECC BASE + 0x24**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MECC16_CODE11[191:176]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MECC16_CODE10[175:160]															

MLC ECC16 Code Register 6(MECC16_6) **ECC BASE + 0x28**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MECC16_CODE12[207:192]															

Note: Real ECC Code Size for MLC ECC16 mode is 207 bits, For the remainder bits of MLC ECC16 Code Register 6, zeros are padded.

Note: MLC ECC16 Code Register must be written and read by word access.

SLC ECC Error Address (SECC_EADDRx, x = 0,…,15) **ECC BASE50 + n*4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SECC_EADDRx[14:0]														

Note:To correct errors for SLC ECC

- SECC_EADDR[14:3] : Byte Error Address

- SECC_EADDR[2:0] : Error Bit Plane

Corrected DATA = ADDR[SECC_EADDR[14:3]] ^ (1 << SECC_EADDR[2:0])
(Byte) (Byte)

MLC ECC4 Error Address Register (MECC4_EADDRn, n = 0,1,2,3) **ECC BASE50 + n*4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			EADDR[12:00]												

EADDR	[12:00]	MLC ECC Error Address
N		A bit address of ERROR

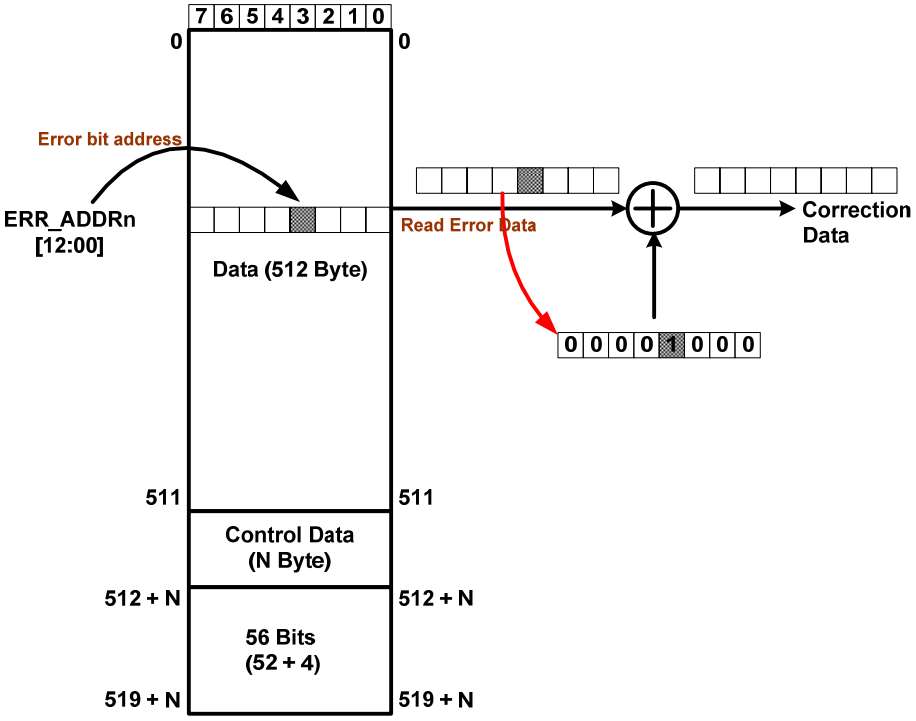


Figure 25.2 Example of MLC ECC4 error correction

MLC ECC8 Error Address Register (MECC8_EADDRn, n = 0,…,7) **ECC BASE50 + n*4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			EADDR[12:00]												

Note: Refer to MECC4_EADDRn

MLC ECC12 Error Address Register (MECC12_EADDRn, n = 0,…,11) **ECC BASE50 + n*4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			EADDR[12:00]												

Note: Refer to MECC4_EADDRn

MLC ECC14 Error Address Register (MECC14_EADDRn, n = 0,…,13) **ECC BASE50 + n*4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			EADDR[12:00]												

Note: Refer to MECC4_EADDRn

MLC ECC16 Error Address Register (MECC16_EADDRn, n = 0,…,15) **ECC BASE50 + n*4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			EADDR[12:00]												

Note: Refer to MECC4_EADDRn

ECC Error Number Register (ERRNUM) **0xF005B070**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERRNUM[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRNUM[15:0]															

ERRNUM	SLC ECC Status
[2*n+1:2*n]	DECODE
00	No Error
01	ECC Error
10	Correctable Error
11	Correction Impossible Error

(n = 0,…, 15)

For MLC ECC4/8/12/14/16

ERRNUM [4:0]	MLC ECC4 DECODE	MLC ECC8 DECODE	MLC ECC12 DECODE	MLC ECC14 DECODE	MLC ECC16 DECODE
00000	No Error	No Error	No Error	No Error	No Error
00001	1 BIT Error	1 BIT Error	1 BIT Error	1 BIT Error	1 BIT Error
00010	2 BIT Error	2 BIT Error	2 BIT Error	2 BIT Error	2 BIT Error
00011	3 BIT Error	3 BIT Error	3 BIT Error	3 BIT Error	3 BIT Error
00100	4 BIT Error	4 BIT Error	4 BIT Error	4 BIT Error	4 BIT Error
00101	-	5 BIT Error	5 BIT Error	5 BIT Error	5 BIT Error
00110		6 BIT Error	6 BIT Error	6 BIT Error	6 BIT Error
00111		7 BIT Error	7 BIT Error	7 BIT Error	7 BIT Error
01000		8 BIT Error	8 BIT Error	8 BIT Error	8 BIT Error
01001		-	9 BIT Error	9 BIT Error	9 BIT Error
01010			10 BIT Error	10 BIT Error	10 BIT Error
01011			11 BIT Error	11 BIT Error	11 BIT Error
01100			12 BIT Error	12 BIT Error	12 BIT Error
01101			-	13 BIT Error	13 BIT Error
01110				14 BIT Error	14 BIT Error
01111				-	15 BIT Error
10000					16 BIT Error
10001 ~ 11110					-
11111	correction impossible error				

ECC Interrupt Request Register (ECC_IREQ) 0xF005B094

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				M16EF	M16DF	M14EF	M14DF	M12EF	M12DF	M8EF	M8DF	M4EF	M4DF	SEF	SDF
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					M16DI		M14DI		M12DI		M8DI		M4DI		I

M16EF	[27]	MLC ECC16 Encoding Flag Register
0	Read	MLC ECC16 Encoding is not Completed
1	Read	MLC ECC16 Encoding is Completed
1	Write	MLC ECC16 Encoding Flag Clear

M16DF	[26]	MLC ECC16 Decoding Flag Register
0	Read	MLC ECC16 Decoding is not Completed
1	Read	MLC ECC16 Decoding is Completed
1	Write	MLC ECC16 Decoding Flag Clear

M14EF	[25]	MLC ECC14 Encoding Flag Register
0	Read	MLC ECC14 Encoding is not Completed
1	Read	MLC ECC14 Encoding is Completed
1	Write	MLC ECC14 Encoding Flag Clear

M14DF	[24]	MLC ECC14 Decoding Flag Register
0	Read	MLC ECC14 Decoding is not Completed
1	Read	MLC ECC14 Decoding is Completed
1	Write	MLC ECC14 Decoding Flag Clear

M12EF	[23]	MLC ECC12 Encoding Flag Register
0	Read	MLC ECC12 Encoding is not Completed
1	Read	MLC ECC12 Encoding is Completed
1	Write	MLC ECC12 Encoding Flag Clear

M12DF	[22]	MLC ECC12 Decoding Flag Register
0	Read	MLC ECC12 Decoding is not Completed
1	Read	MLC ECC12 Decoding is Completed
1	Write	MLC ECC12 Decoding Flag Clear

M8EF	[21]	MLC ECC8 Encoding Flag Register
0	Read	MLC ECC8 Encoding is not Completed
1	Read	MLC ECC8 Encoding is Completed
1	Write	MLC ECC8 Encoding Flag Clear

M8DF	[20]	MLC ECC8 Decoding Flag Register
0	Read	MLC ECC8 Decoding is not Completed
1	Read	MLC ECC8 Decoding is Completed
1	Write	MLC ECC8 Decoding Flag Clear

M4EF	[19]	MLC ECC4 Encoding Flag Register
0	Read	MLC ECC4 Encoding is not Completed
1	Read	MLC ECC4 Encoding is Completed
1	Write	MLC ECC4 Encoding Flag Clear

M4DF	[18]	MLC ECC4 Decoding Flag Register
0	Read	MLC ECC4 Decoding is not Completed
1	Read	MLC ECC4 Decoding is Completed
1	Write	MLC ECC4 Decoding Flag Clear

SEF	[17]	SLC ECC Encoding Flag Register
0	Read	SLC ECC Encoding is not Completed
1	Read	SLC ECC Encoding is Completed
1	Write	SLC ECC Encoding Flag Clear

SDF	[16]	SLC ECC Decoding Flag Register
0	Read	SLC ECC Decoding is not Completed

1	Read	SLC ECC Decoding is Completed
1	Write	SLC ECC Decoding Flag and Interrupt Request Clear

M16DI	[10]	MLC ECC16 Decoding Interrupt Request Register
0	Read	MLC ECC16 Decoding Interrupt Request is not Occurred
1	Read	MLC ECC16 Decoding Interrupt Request is Occurred
1	Write	MLC ECC16 Decoding Flag and Interrupt Request Clear

M14DI	[8]	MLC ECC14 Decoding Interrupt Request Register
0	Read	MLC ECC14 Decoding Interrupt Request is not Occurred
1	Read	MLC ECC14 Decoding Interrupt Request is Occurred
1	Write	MLC ECC14 Decoding Flag and Interrupt Request Clear

M12DI	[6]	MLC ECC12 Decoding Interrupt Request Register
0	Read	MLC ECC12 Decoding Interrupt Request is not Occurred
1	Read	MLC ECC12 Decoding Interrupt Request is Occurred
1	Write	MLC ECC12 Decoding Flag and Interrupt Request Clear

M8DI	[4]	MLC ECC8 Decoding Interrupt Request Register
0	Read	MLC ECC8 Decoding Interrupt Request is not Occurred
1	Read	MLC ECC8 Decoding Interrupt Request is Occurred
1	Write	MLC ECC8 Decoding Flag and Interrupt Request Clear

M4DI	[2]	MLC ECC4 Decoding Interrupt Request Register
0	Read	MLC ECC4 Decoding Interrupt Request is not Occurred.
1	Read	MLC ECC4 Decoding Interrupt Request is Occurred
	Write	MLC ECC4 Decoding Flag and Interrupt Request Clear

25.3 ECC Coding Sequence

25.3.1 SLC ECC Encoding Sequence

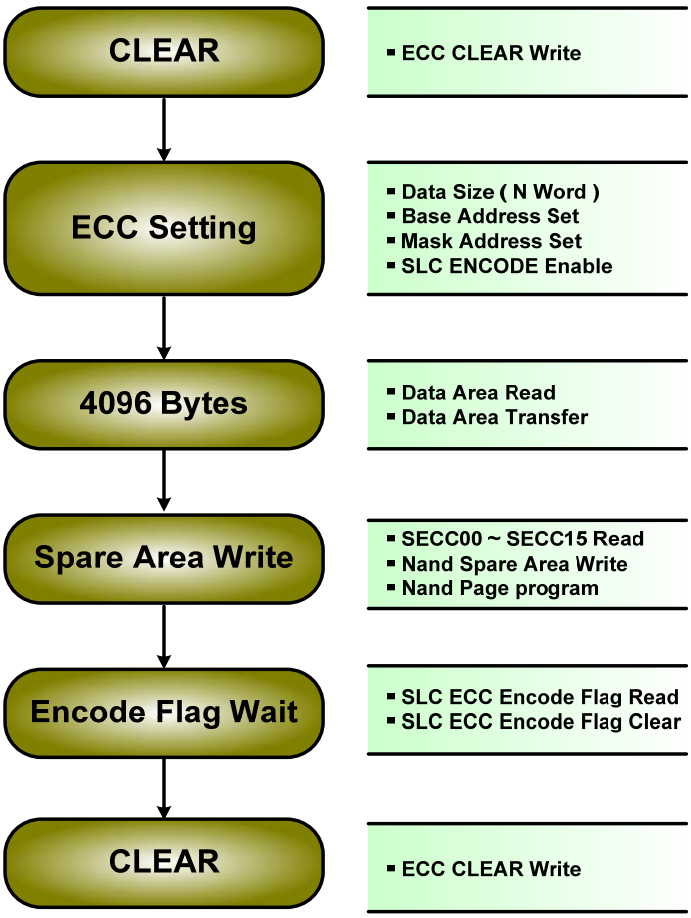


Figure 25.3 SLC ECC Encoding Sequence

25.3.2 SLC ECC Decoding Sequence

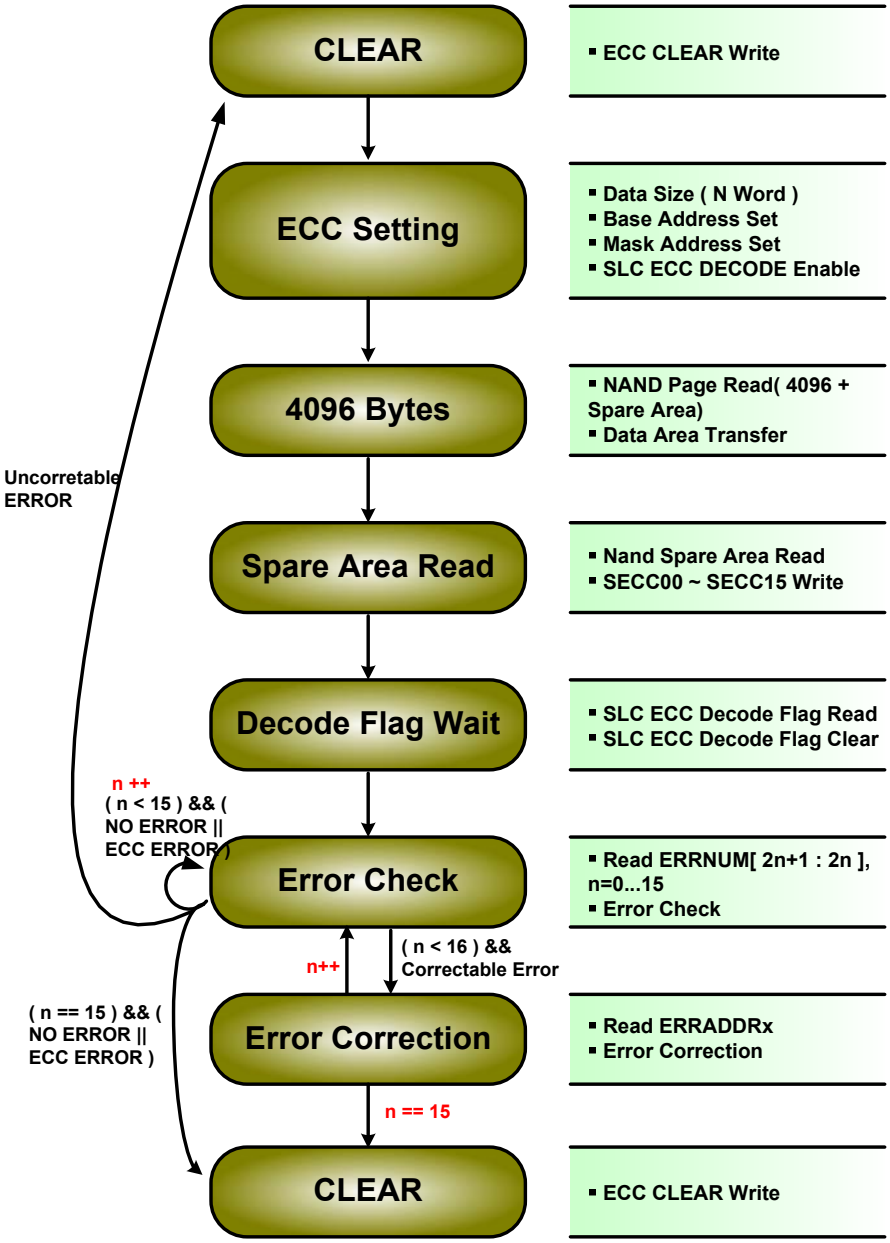


Figure 25.4 SLC ECC Decoding Sequence

25.3.3 MLC ECC4/8/12/14/16 Encoding Sequence

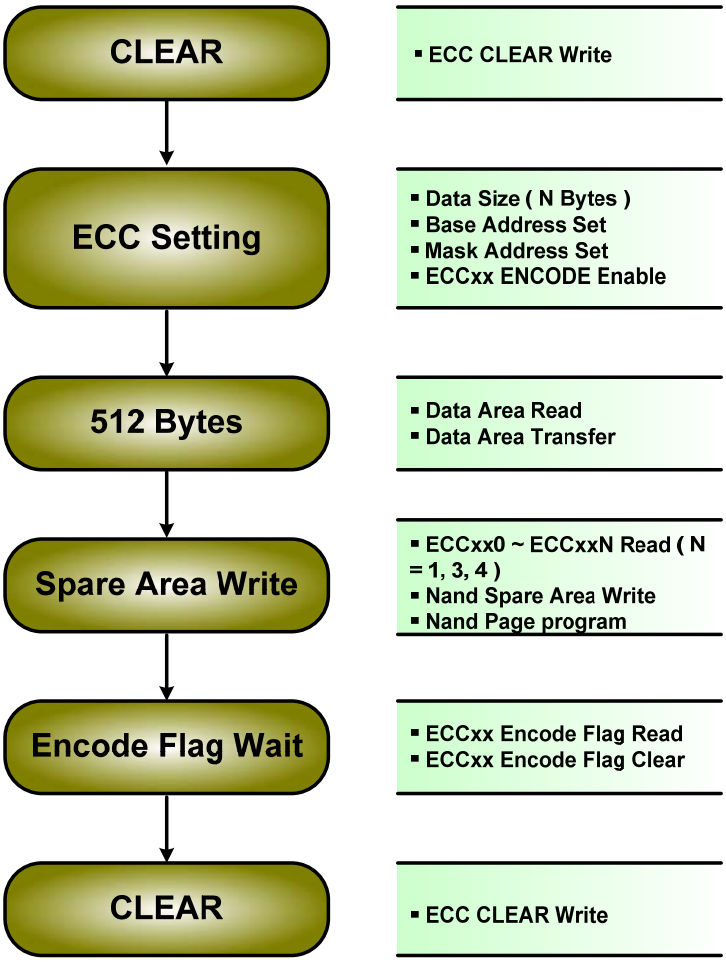


Figure 25.5 MLC ECC4/8/12/14/16 Encoding Sequence

25.3.4 MLC ECC4/8/12/14/16 Decoding Sequence

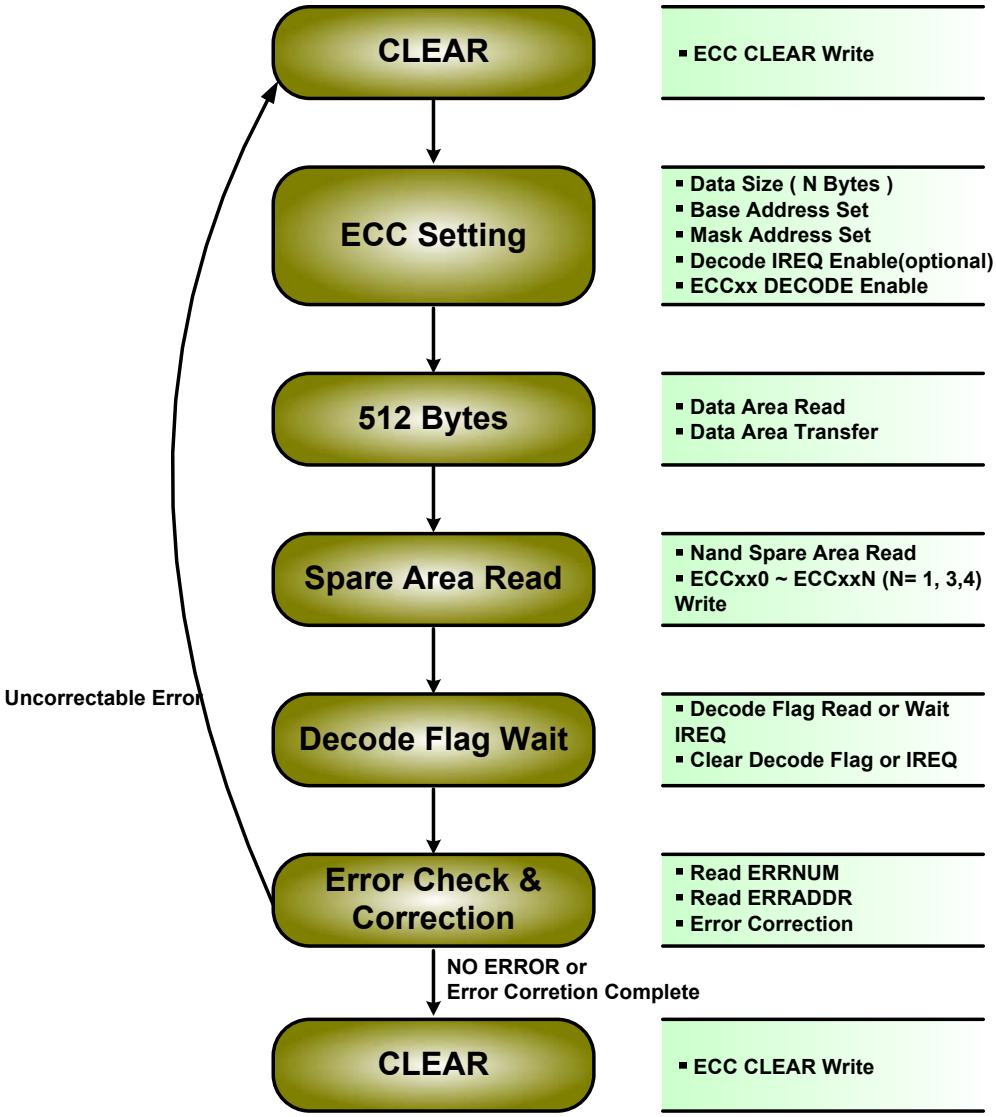


Figure 25.5 MLC ECC4/8/12/14/16 Decoding Sequence

26 MPEFEC

26.1 Overview

The MPEFEC(Multi-Protocol Encapsulation Forward Error Correction) is a error correction method in DVB-H standard. Refer to DVB-H specification for more details.

As a result of transport stream parsing in DVB-H, a MPEFEC frame is generated shown as below.

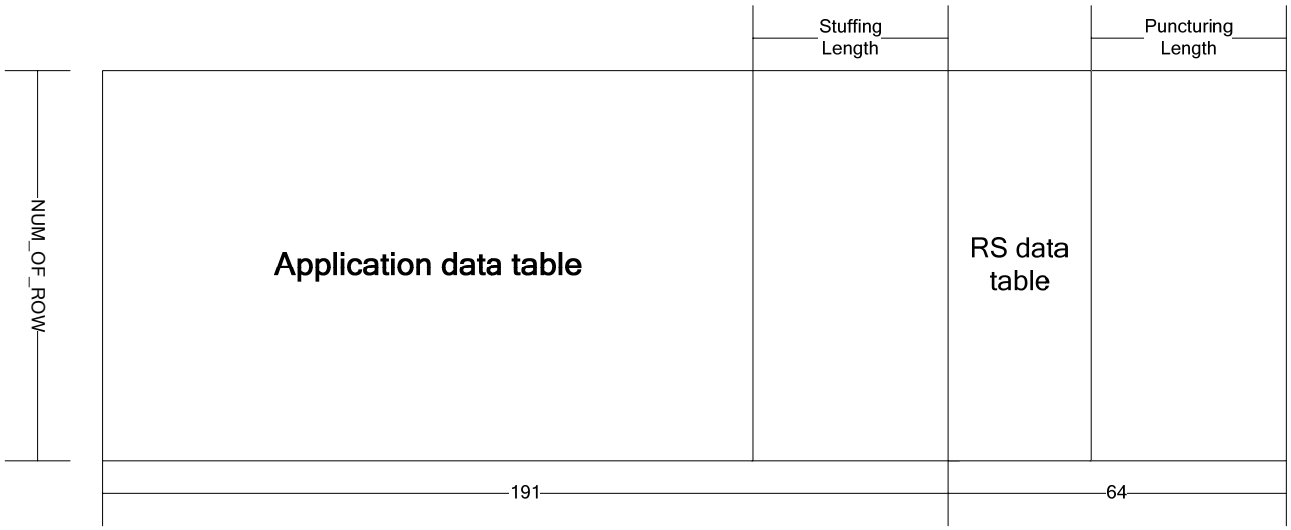


Figure 26.1 MPEFEC frame

The “NUM_OF_ROW” can be one of 4 mode, 256, 512, 768, 1024.

The MPEFEC frame is divided by two part, i.e. application data table and RS data table. The application data table is used in upper layer of DVB-H and RS data table is used for error correction.

The stuffing can not be received externally but must be filled with 0s. Also the puncturing can not be received but will be filled after error correction by MPEFEC.

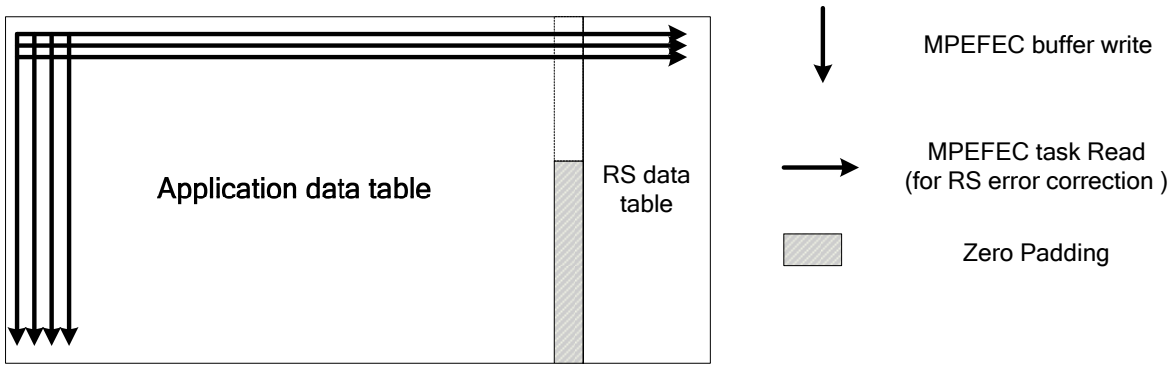


Figure 26.2 MPEFEC buffer write

The Error correction is accomplished, after writing stream into MPEFEC in the direction of column, by reading RS(255,191) code in the direction of row.

To enhance the performance, “ERASURE” can be used that MPEFEC can be notified of erroneous position in the frame. The ERASURE is shown in below.

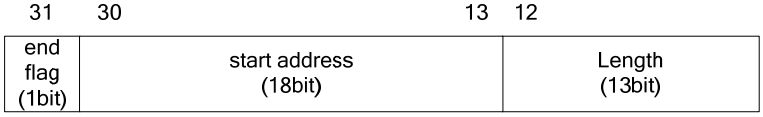


Figure 26.3 MPEFEC Erasure

Where Bit[31] indicates the end of ERASURE tags, Bit[30:13] has the start address and Bit[12:0] has the length of the ERASURE.

26.2 Block Diagram

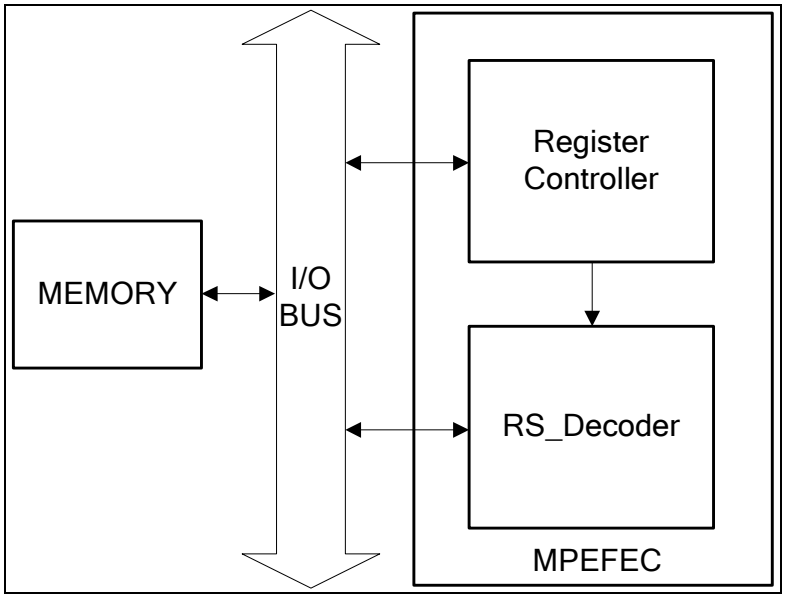


Figure 26.4 MPEFEC Top Block Diagram

26.3 Register Description

Table 26.1 MPEFEC Register Map (Base Address = 0xF0510000)

Name	Address	Type	Reset	Description
MERR	0x00	R/W	0x00000000	MPEFEC Enable/ RESET
MSR	0x04	W	0x00000000	MPEFEC Start (Auto clear)
MFRNR	0x08	R/W	0x03FF03FF	MPEFEC Frame Row Number Register
MFSAR	0x0C	R/W	0x00000000	MPEFEC Frame Source Address Register
EFSAR	0x10	R/W	0x00000000	Erasure Flag Source Address Register
MCR	0x14	R/W	0x00014000	MPEFEC Control Register
MSTR	0x18	R	0x00000000	MPEFEC Status Register
MIER	0x1C	R	0x00000000	MPEFEC IRQ Enable Register
MICR	0x20	W	0x00000000	MPEFEC IRQ Clear Register
MARR	0x24	R/W	0x03FF0000	MPEFEC Active row Register
ECR	0x28	R/W	0x00000000	Error Count Register

MPEFEC Enable/Reset Register (MERR) 0xF0510000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														ME	INIT

ME [1]	MPEFEC Enable
0	Disable
1	Enable

INIT [0]	INIT
0	MPEFEC RESET
1	Normal operation

MPEFEC Start Register (MSR) 0xF0510004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															MS

MS [0]	MPEFEC Start
0	Disable
1	MPE_FEC Start (auto clear)

MPEFEC Frame Row Number Register (MFRNR) 0xF0510008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						ARN									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						FRN									

ARN [25:16]	Active Row Number
N	Active Row number – 1 Same as Frame Row Number(FRN)

Sets the active number of rows in MPEFEC frame.

FRN [9:0]	Frame Row Number
N	Frame Row number – 1

Sets the totoal number of rows in MPEFEC frame.

MPEFEC Frame Source Address Register (MFSAR) 0xF051000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MDSA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDSA[15:0]															

MDSAR[31:0]	Memory Data Start Address
N	The memory start address where the MPEFEC frame is located

Erasure Flag Source Address Register (EFSAR) 0xF0510010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MEFSA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEFSA[15:0]															

MEFSA[31:0]	Memory Erasure Flag Start Address
N	The memory start address where the ERASURE flag is located

MPEFEC Control Register (MCR) 0xF0510014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DE	EE	R*	ET						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SCL								R*	PCL						

DE [25]	Write Direction
0	Forward (must be fixed to forward)
1	reverse

EE [24]	Erasure Enable
0	Disable
1	Enable

ET [22:16]	Erasure Threshold
N	Erasure threshold Value (0~64) If the numbers of ERASURE is more than threshold, ERASUREs are not used.

SCL [15:8]	Stuffing Column Length
N	Stuffing Column Length Value (0~191)

PCL [6:0]	Puncturing Column Length
N	Puncturing Column Length Value (0~64)

MPEFEC Status Register (MSTR) 0xF0510018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved															MS

MS [0]	MPEFEC Status
0	mpe_over (RS Fail) It notifies that MPEFEC frame has non error corrected data in it.
1	mpe_done (RS done) Indicates whether MPEFEC operation is ended or not.

MPEFEC IRQ Enable Register (MIER) 0xF051001C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved															MIE

MIE [0]	MPEFEC IRQ Enable
0	mpe_over IRQ Enable
1	mpe_done IRQ Enable

MPEFEC IRQ Clear Register (MICR) 0xF0510020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved															MIC

MI [0]	MPEFEC IRQ Clear
0	mpe_over IRQ Clear
1	mpe_done IRQ Clear

MPEFEC Active row Register (MARR) 0xF0510024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						SRN									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						ERN									

SRN [25:16]	End Row Number
N	End row number Same as Frame Row Number(FRN)

ERN [9:0]	Start Row Number
N	Start row number Always be set to 0

Error Count Register (ECR) **0xF0510028**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECNT[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ECNT[15:0]															

27 IOBUS Configuration Registers

The IOBUS Configuration block has several registers named as USBOTG, USB11H, IOBAPB, STORAGE, HCLKEN0/1, HCLKMEN, HRSTEN0/1, USBOTG0/1/2/3 and IO_A2X.

Table 27.1 IOBUS Configuration Register Map (Base Address = 0xF05F5000)

Name	Address	Description
USBOTG	0x00	Refer to USB OTG Configuration Register (OTGCR) in "15.2 Register Description for USB 2.0 OTG Controller"
USB11H	0x04	Refer to USB 1.1 Host Configuration Register (USB11H) in "14.2 Register Description for USB 1.1 Host Controller & Transceiver"
IOBAPB	0x08	IOBUS APB wait counter Register
STORAGE	0x0C	Storage Device Configuration Register
HCLKEN0	0x10	IOBUS AHB clock enable Register 0
HCLKEN1	0x14	IOBUS AHB clock enable Register 1
HCLKMEN	0x18	DMA AHB clock mask enable Register
-	0x1C	Reserved
HRSTEN0	0x20	IOBUS AHB Hreset Control register 0
HRSTEN1	0x24	IOBUS AHB Hreset Control register 1
USBOTG0	0x28	Refer to USB PHY Configuration Register0 (UPCR0) in "15.2 Register Description for USB 2.0 OTG Controller"
USBOTG1	0x2C	Refer to USB PHY Configuration Register1 (UPCR1) in "15.2 Register Description for USB 2.0 OTG Controller"
USBOTG2	0x30	Refer to USB PHY Configuration Register2 (UPCR2) in "15.2 Register Description for USB 2.0 OTG Controller"
USBOTG3	0x34	Refer to USB PHY Configuration Register3 (UPCR3) in "15.2 Register Description for USB 2.0 OTG Controller"
IO_A2X	0x38	IOBUS AHB2AXI Control Register

IOBUS APB Wait Counter Register (IOBAPB) 0xF05F5008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-														ECCS	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-							-	-				-	-		-

Field	Name	RW	Reset	Description
31-6	-		0	Reserved
5-3	TSWAIT	R/W	0x0	Delay latency added when access to TSADC block
2-0	RTCWAIT	R/W	0x0	Delay latency added when access to RTC block

Storage Controller Configuration Register (STORAGE) 0xF05F500C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-												SLUMBER	CK_RATE	ECCS	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-							-	-				-	-		-

Field	Name	RW	Reset	Description
31 ~ 20	-		0x0	Reserved
19	SLUMBER	R/W	0	SATA PHY Slumber mode on/off 0: off 1: on
18	CK_RATE	R/W	0	Reference clock rate supplied to SATA PHY 0 : 25MHz 1: 100MHz
17 ~ 16	ECCS	RW	0x0	ECC monitor bus selection register 0 : ECC 1 : AHB bus1(USB1.1Host, USBOTG, SD/MMC, IDE, DMA0, DMA1) 2 : AHB bus 2(AUDIO DMA, GPSB, DMA2, DMA3) 3 : NFC
15-0	-		0x0	Reserved

IOBUS AHB Clock Enable Register 0(HCLKEN0) 0xF05F5010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HCLKEN0[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HCLKEN0[15:0]															

Field	Name	RW	Reset	Description
31-0	HCLKEN0	R/W	0xFFFFFFFF	Enable signals of HCLK clock gating logic AHB HCLKs, supplied to each peripheral block, are controlled by clock enable signal. 0 : disable clock 1: enable clock The bit position indicates each sub-block which is controlled by. BIT 0: USB1.1 Host BIT 1: USB2.0 OTG BIT 2: IDE controller BIT 3: DMA controller BIT 4: SD/MMC Controller BIT 5: SATA Host Controller BIT 6: Memory Stick Controller BIT 7: I2C Controller BIT 8: NFC Controller BIT 9: External Host Interface 0 BIT 10: External Host Interface 1 BIT 11: UART Controller 0 BIT 12: UART Controller 1 BIT 13: UART Controller 2 BIT 14: UART Controller 3 BIT 15: UART Controller 4 BIT 16: UART Controller 5 BIT 17: GPSB Controller 0 BIT 18: GPSB Controller 1 BIT 19: GPSB Controller 2 BIT 20: GPSB Controller 3 BIT 21: GPSB Controller 4 BIT 22: GPSB Controller 5 BIT 23: DAI&CD Controller, DAI/CD interface in Audio DMA Controller BIT 24: ECC calculator BIT 25: SPDIF TX Controller, SPDIF interface in Audio DMA Controller BIT 26: RTC BIT 27: TSADC Controller BIT 28: Test Block BIT 29: Reserved BIT 30: Reserved BIT 31: AUDIO DMA Controller

IOBUS AHB Clock Enable Register 1(HCLKEN1) 0xF05F5014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													HCLKEN1		

Field	Name	RW	Reset	Description
31-3	-		0	Reserved
2-0	HCLKEN1	R/W	0x7	AHB HCLKs, supplied to each peripheral block, are controlled by clock enable signal. 0 : disable clock 1: enable clock The bit position indicates each sub-block which is controlled by. BIT 0: MPE_FEC BIT 1: TSIF BIT 2: SRAM Controller

IOBUS AHB Clock Mask Enable Register 1(HCLKMEN) 0xF05F5018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		REMCLR													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												HCLKMEN			

Field	Name	RW	Reset	Description
31-30	-			Reserved
29	REMCLR	R/W	0x1	Remote control interface Reset(active low)
28-4	-			Reserved
3	HCLKMEN	R/W	0x1	AHB HCLKs Mask enable signal 0 : disable mask 1: enable mask
2-0	-		0	Reserved

IOBUS AHB HRESETn Enable Register 0(HRSTEN0) 0xF05F5020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HRSTEN1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HRSTEN1[15:0]															

Field	Name	RW	Reset	Description
31-0	HRSTN0	R/W	0xFFFFFFFF	Enable signal of HRESETn gating logic AHB reset signals, supplied to each peripheral block, are controlled by HRESETn enable signal. 0 : goto reset state 1: bypass reset signal The bit position indicates each sub-block which is controlled by. BIT 0: USB1.1 Host BIT 1: USB2.0 OTG BIT 2: IDE controller BIT 3: DMA controller BIT 4: SD/MMC Controller BIT 5: SATA Host Controller BIT 6: Memory Stick Controller BIT 7: I2C Controller BIT 8: NFC Controller BIT 9: External Host Interface 0 BIT 10: External Host Interface 1 BIT 11: UART Controller 0 BIT 12: UART Controller 1 BIT 13: UART Controller 2 BIT 14: UART Controller 3 BIT 15: UART Controller 4 BIT 16: UART Controller 5 BIT 17: GPSB Controller 0 BIT 18: GPSB Controller 1 BIT 19: GPSB Controller 2 BIT 20: GPSB Controller 3 BIT 21: GPSB Controller 4 BIT 22: GPSB Controller 5 BIT 23: DAI/CDIF interface BIT 24: ECC calculator BIT 25: SPDIF TX Controller BIT 26: RTC BIT 27: TSADC Controller BIT 28: Test Block BIT 29: Reserved BIT 30: Reserved BIT 31: AUDIO DMA Controller

IOBUS AHB HRESETn Enable Register 1(HRSTEN1) 0xF05F5024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													HRSTEN1		

Field	Name	RW	Reset	Description
31-3	-		0	Reserved
2-0	HRSTEN1	R/W	0x7	Clock Enable signal to HCLK clock gating logic AHB reset signals, supplied to each peripheral block, are controlled by HRESETn enable signal. 0 : goto reset state 1: bypass reset signal The bit position indicates each sub-block which is controlled by. BIT 0: MPE_FEC BIT 1: TSIF BIT 2: SRAM Controller

IOBUS AHB2AXI Control Register (IO_A2X) 0xF05F5038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A2XMOD1				A2XMOD0			

Field	Name	RW	Reset	Description
31-6	-		0	Reserved
5-3	A2XMOD1	R/W	0x7	IOBUS to Memory Controller interface1 control register It affects READ operation from following bus master to memory : Audio DMA, GPSB, DMA2/3, EHI1 A2XMOD1[0] : flushs prefetch buffer when bus state is IDLE or WRITE. A2XMOD1[1] : stop reading prefetch buffer during WRITE operation is not completed. A2XMOD1[2] : not used
2-0	A2XMOD0	R/W	Ox7	IOBUS to Memory Controller interface0 control register It affects READ operation from following bus master to memory : USB1.1Host, USB OTG, SD/MMC, IDE, DMA0/1, MPEFEC, EHI0 A2XMOD0[0] : flushs prefetch buffer when bus state is IDLE or WRITE. A2XMOD0[1] : stop reading prefetch buffer during WRITE operation is not completed. A2XMOD0[2] : not used

PART6 – DDI BUS
TCC8900

High Performance and Low-Power Processor
For Digital Media Applications

Rev. 1.02

Oct 09, 2009

Telechips

Revision History

Date	Revision	Description
2008-12-11	0.00	Initial release
2009-01-21	0.01	Revise DDI_CONFIG and Camera IF register address.
2009-02-25	0.02	update LCD register map, LCDSI MODE/STATUS register description
2009-06-04	0.03	update LCD register (LCTRL) description, update example for NTSC/PAL interface
2009-08-07	1.00	add “7.3~7.8”, “7.11~7.12”in Chapter 7. NTSC/PAL Encoder Composite Output Update “8.6” in Chapter 8. HDMI. (1920x1080p@50Hz, 1920x1080i@50Hz)
2009-09-04	1.01	The description for the LSTATUS register has been changed.
2009-10-09	1.02	correct 656FCR1, ICPCR1 register in camera interface - add the field information field

TABLE OF CONTENTS

Contents

1 Introduction 1-1
1.1 Feature 1-1
2 Bus Architecture 2-3
3 Address and Register Map 3-5
4 LCD Interface 4-7
4.1 Overview 4-7
4.2 Raw Image Source Processing 4-9
4.2.1 Image Formats and Relation Between Images and Layers 4-9
4.2.2 Image Width, Height and Offset 4-11
4.2.3 Conversion Between YCbCr Color Space and RGB Color Space 4-12
4.2.4 Look-Up Table for Each Image Sources 4-12
4.2.5 Alpha-blending with Multiple Layers 4-13
4.2.6 Chroma-Keying with Multiple Layers 4-13
4.2.7 Gamma Correction Function 4-14
4.2.8 Contrast and Brightness Adjust Function 4-18
4.3 Display Interface 4-18
4.3.1 STN-LCD 4-18
4.3.2 TFT-LCD 4-20
4.3.3 NTSC/PAL Interface 4-22
4.4 Register Description 4-24
5 LCD System Interface 5-47
5.1 Overview 5-47
5.2 Operation 5-47
5.2.1 Reading/Writing operation through the on-chip CPU 5-47
5.2.2 Writing Operation Through LCD Controller 5-48
5.3 Register Descriptions 5-49
6 Memory To Memory Scaler 6-53
6.1 Overview 6-53
6.2 Operation 6-54
6.3 Registers 6-56
7 NTSC / PAL Encoder Composite Output 7-71
7.1 Overviews 7-71
7.2 Features 7-71
7.3 Selecting Operation Mode 7-71
7.4 Input Timing 7-72
7.5 Video Standard Selection 7-73
7.6 Basic Video Adjustments 7-74
7.7 Programmable Bandwidth 7-74
7.8 Analog Video Output Configuration 7-74
7.9 Registers 7-75
7.10 Example for NTSC/PAL Interface 7-85
7.11 Copy Generation Management Systems 7-87
7.12 10-Bit DAC 7-89
7.12.1 HDMI Specific Features 7-89
7.12.2 Functional Description 7-89
8 HDMI 8-91
8.1 Overview 8-91
8.1.1 HDMI Specific Features 8-91
8.2 Architecture 8-91
8.3 HDMI Controller (LINK) 8-92
8.3.1 HDCP KEY Management 8-92
8.3.2 Interface Protocol 8-92
8.3.2.1 Video Input Interface 8-92
8.3.2.2 Audio Input Interface 8-92
8.3.2.3 HPD 8-93
8.3.2.4 CEC Interface 8-93
8.3.2.5 AESKEY 8-93
8.3.2.6 Interrupt Timing 8-93
8.4 HDMI Controller Register Description 8-94
8.4.1 Control Registers 8-94
8.4.2 HDMI Core Registers 8-98
8.4.2.1 Control Registers 8-101
8.4.2.2 Video Related Registers 8-107
8.4.2.3 Audio related Packet Registers 8-115
8.4.2.4 ACP Packet Registers 8-123

8.4.2.5 ISRC1/2 packet registers 8-124
8.4.2.6 AVI InfoFrame registers 8-126
8.4.2.7 Audio InfoFrame registers 8-127
8.4.2.8 MPEG Source InfoFrame 8-128
8.4.2.9 Source Product Descriptor InfoFrame (or general packet generation) 8-129
8.4.2.10 HDCP Register Description 8-130
8.4.2.11 Ri Check Registers 8-137
8.4.2.12 Gamut Metadata Packet Registers 8-138
8.4.2.13 Video Mode Registers 8-140
8.4.2.14 AES Registers 8-142
8.4.3 SPDIF Registers 8-144
8.4.3.1 Control Registers 8-145
8.4.3.2 Channel Status Registers 8-155
8.4.3.3 SPDIFIN Info Register 8-158
8.4.4 I2S Registers 8-162
8.4.4.1 Control Registers 8-163
8.4.4.2 Channel Status Register 8-168
8.4.4.3 Mux Control Register 8-171
8.4.4.4 Interrupt Control Registers 8-172
8.4.4.5 Output Buffer Registers 8-173
8.4.5 CEC Registers 8-174
8.4.5.1 CEC Configure Registers 8-175
8.4.5.2 Tx Related Registers 8-179
8.4.5.3 Rx Related Registers 8-181
8.5 HDMI PHY 8-186
8.5.1 Clock Scheme of the HDMI TX PHY Core 8-186
8.5.1.1 Using the Internal Video PLL for Pixel Clock Generation 8-186
8.5.1.2 Using an External Video PLL for Pixel Clock Generation 8-186
8.5.2 PHY Configuration Change through I2C 8-187
8.5.3 PHY Reset Timing 8-188
8.5.4 Register Map 8-188
8.5.5 Recommended Register Setting Value 8-190
8.5.5.1 Using the Integrated Video PLL for Pixel Clock Generation 8-190
8.5.5.2 Using an External Video PLL for Pixel Clock Generation and Configuring the Integrated Video PLL as a jitter-filter PLL 8-191
8.5.5.3 Using External Video PLL for Pixel Clock Generation and By-passing Internal Video PLL 8-193
8.6 Examples - HDMI Register Setting 8-195
8.6.1 1920x1080p 8-195
8.6.2 1920x1080i 8-197
8.6.3 1280x720p 8-198
8.6.4 720x480p 8-200
8.6.5 720x576p 8-202
8.6.6 720x480i 8-202
8.6.7 720x576i 8-203
9 Camera Interface 9-205
9.1 Overview 9-205
9.2 Operation 9-206
9.2.1 External Camera Module Interface (CAMIF) 9-206
9.2.2 Effecter 9-209
9.2.3 CIF Scaler 9-209
9.2.4 Overlay 9-210
9.2.5 Store to the Memory 9-210
9.3 Camera Register Descriptions 9-212
9.4 Effecter Register Descriptions 9-232
9.5 Scaler Register Descriptions 9-237
9.6 User guide for CAMIF Setting 9-238
9.6.1 External Camera Module Interface (CAMIF) 9-238
9.6.2 Camera Hardware ON/OFF Sequence 9-239
9.6.3 Timing Tuning for Camera Sensor 9-241
9.6.4 Zooming Control 9-242
9.6.5 Using One Frame Capture Signal 9-244
9.6.6 Register Update 9-245
9.6.7 Capture 9-245
9.6.8 Timing and Status Diagram by CAMIF HS/VS Block 9-247
9.6.9 Effecter User Guide 9-248
10 Video and Image Quality Enhancer (VIQE) 10-251
10.1 Overview 10-251
10.2 De-interlacing Module 10-253
10.3 De-noising Module 10-254

10.3.1 Temporal De-noiser 10-254
10.3.2 Spatial De-noiser 10-255
10.4 RDMA (Read DMA) Module 10-255
10.5 ODMA (OUTPUT DMA) Module 10-256
10.6 Gamut Mapper Module 10-257
10.7 Histogram Generator Module 10-257
10.8 Register Description 10-259
11 LVDS 11-323
11.1 Overview 11-323
11.1.1 LVDS Specific Features 11-323
11.2 Architecture 11-323
11.2.1 LVDS 11-324
11.2.2 Block Diagram of LVDS 11-324
11.2.3 Timing Diagram of LVDS 11-324
12 DDI_CONFIG 12-325
12.1 Overview 12-325
12.1.1 DDI_CONFIG Specific Features 12-325
12.2 Block Diagram of DDI_CONFIG 12-325
12.3 Register Description 12-325
13 DDI_Cache 13-339
13.1 Overview 13-339
13.1.1 DDI_CACHE Specific Features 13-339
13.2 Block Diagram of DDI_CACHE 13-339
13.3 Register Description 13-339

Figures

Figure 2.1 DDIBUS Hardware Architecture 2-3
Figure 4.1 LCD Controller Block Diagram 4-7
Figure 4.2 Overall Image Data Flow 4-8
Figure 4.3 Virtual Display in the LCDC 4-9
Figure 4.4 Relationship between Layers and OP Bit 4-9
Figure 4.5 Supported Pixel Data Format (BPP bits of LI*n*C register) 4-10
Figure 4.6 RGB2RGB888 Conversion 4-11
Figure 4.7 Definitions of Frame Window and Display Window 4-12
Figure 4.8 LCD Color Lookup Table 4-13
Figure 4.9 LCD Gamma Correction Function 4-15
Figure 4.10 LCD Gamma Correction Function Example 1 4-16
Figure 4.11 LCD Gamma Correction Function Example 2 4-17
Figure 4.12 STN Mode Timing Diagram 4-19
Figure 4.13 TFT Mode Timing Diagram 4-20
Figure 4.14 NTSC Interlace Mode Timing Diagram 4-22
Figure 4.15 PAL Interlace Mode Timing Diagram 4-23
Figure 4.16 CCIR656 Embedded sync. Information 4-23
Figure 4.17 CCIR656 Format Diagram 4-24
Figure 4.18 Output Pixel Data Format 4-29
Figure 4.19 Bit Padding 4-41
Figure 5.1 LCD System Interface Block Diagram 5-47
Figure 5.2 Writing / Reading Operation through on-chip CPU 5-48
Figure 5.3 Example of LCDC Output Signals for LCDSI 5-48
Figure 6.1 Scaler Block Diagram 6-53
Figure 6.2 Memory to Memory Scaling Operation 6-54
Figure 6.3 Storing the Result Image 6-55
Figure 7.1 Digital Input Timing(ITU-R BT.656 8bit parallel Input) 7-72
Figure 7.2 Digital Input Timing (ITU-R BT.601 4:2:2 16bit Parallel Input) 7-72
Figure 7.3 Example of Input Timing 7-73
Figure 7.4 Luma Bandwidth 7-74
Figure 7.5 Chroma Bandwidth 7-74
Figure 7.6 Copy Generation Management Systems 7-88
Figure 7.7 Example: CRC calculator for CGMS 7-89
Figure 7.8 10-bit DAC Connection 7-89
Figure 8.1 HDMI LINK and PHY in TCC8900 8-91
Figure 8.2 Block Diagram of HDCP Key Management 8-92
Figure 8.3 Timing Diagram for HPD Plug Interrupt 8-93
Figure 8.4 Timing Diagram for HPD Unplug Interrupt 8-93
Figure 8.5 HDMI Clock Scheme Using the Integrated Video PLL for Pixel Clock Generation 8-186
Figure 8.6 Clock Scheme Using an External Video PLL for Pixel Clock Generation (jitter-filter PLL) 8-187
Figure 8.7 Clock Scheme Using an External Video PLL for Pixel Clock Generation (by-passed) 8-187

Figure 8.8 PHY Configuration through I2C with MODE_SET_DONE Register 8-188
Figure 8.9 HDMI PHY Reset Timing 8-188
Figure 9.1 CIF Block Diagram 9-206
Figure 9.2 Input Data Format From the external camera module 9-207
Figure 9.3 Input Image Size From the External Camera Module 9-207
Figure 9.4 Port Control in PCK_CAM and FIELD 9-208
Figure 9.5 Generate Field Signal 9-208
Figure 9.6 Accepting Frame Data from the External Camera Module 9-209
Figure 9.7 Output Image of the Effecter and Input Image of the CIF Scaler 9-210
Figure 9.8 The Geometric Property of Overlay Image 9-210
Figure 9.9 Image Data to be Stored to the Memory 9-211
Figure 9.10 Two Modes to Store the Image Data 9-211
Figure 9.11 CCIR-656 Format Diagram 9-215
Figure 9.12 Input Image Windowing 9-217
Figure 9.13 Overlay Image Windowing 9-227
Figure 9.14 Interconnection between TCC8900 and Camera Sensor 9-238
Figure 9.15 Frequency Relations for All the Clocks 9-239
Figure 9.16 Frequency Relations for All the Clocks 9-239
Figure 9.17 Camera Interface Hardware Turn-On Sequence 9-240
Figure 9.18 Camera Interface Hardware Turn-Off Sequence 9-241
Figure 9.19 Operating Timing Diagram (Correct) 9-242
Figure 9.20 Operating Timing Diagram (Incorrect) 9-242
Figure 9.21 Camera Zoom-In(Up-Scale) and Zoom-Out(Down-Scale) - Correct 9-243
Figure 9.22 Camera Zoom-In(Up-Scale) and Zoom-Out(Down-Scale) - Incorrect 9-244
Figure 9.23 The Timing of Status Signals 9-244
Figure 9.24 Capture Timing Diagram 9-246
Figure 9.25 Camera Interface Hardware Capture Sequence 9-247
Figure 9.26 CAMIF HS/VS Timing Diagram 9-248
Figure 9.27 CAMIF Effector HS/VS Timing Margin 9-249
Figure 10.1 VIQE Block Diagram 10-252
Figure 10.2 De-Interlacer Block Diagram 10-253
Figure 10.3 De-noiser Block Diagram 10-254
Figure 10.4 Temporal De-noiser Block Diagram 10-254
Figure 10.5 Recursive Operation 10-255
Figure 10.6 Spatial de-noising Block 10-255
Figure 10.7 RDMA Block Diagram 10-256
Figure 10.8 ODMA Block Diagram 10-256
Figure 10.9 Timing Diagram in Internal Operating 10-265
Figure 10.10 Timing Diagram in Luminance Delay Register 10-266
Figure 10.11 Image Divide to Original and Filtered Image 10-286
Figure 11.1 Connection of LVDS in TCC8900 11-323
Figure 11.2 Block Diagram of LVDS 11-324
Figure 11.3 Timing Diagram of LVDS 11-324
Figure 12.1 Block Diagram of DDI_CONFIG in TCC8900 12-325
Figure 13.1 Block diagram of DDI_CACHE in TCC8900 13-339

Tables

Table 4.1 STN LCD Palette Address 4-18
Table 4.2 STN LCD Dithering Pattern Register map 4-19
Table 4.3 Monochrome STN LCD (4bits, 1BPP) example 4-20
Table 4.4 TFT LCD (RGB565) Example 4-21
Table 4.5 LCDC Register Map (Base Address = 0xF0200000/0xF0204000) 4-24
Table 5.1 LCDSI Register map(0xF020C000) 5-49
Table 6.1 Scaler Registers (Base Address = 0xF0210000/0xF022000) 6-56
Table 7.1 STN LCD Palette Address 7-73
Table 7.2 Summary of DAC voltage and Codes 7-74
Table 8.1 PHY Register Map (I2C Address = 0x70) 8-188
Table 9.1 CIF Register Map (Base Address = 0xF0230000) 9-212
Table 9.2 Effect Register Map (Base Address = 0xF0230100) 9-232
Table 9.3 Scaler Register Map (Base Address = 0xF0230200) 9-237
Table 10.1 VIQE Register Map (Base Address = 0xF0252000) 10-259
Table 12.1 DDI_CONFIG Register Map (Base Address = 0xF0251000) 12-325
Table 13.1 DDI_CACHE Register Map (Base Address = 0xF0250000) 13-339

1 Introduction

TCC8900 DDIBUS is a dedicated bus system for displaying or capturing video and image raw data. DDIBUS provides not only traditional and essential video interfaces such as LCD interface, NTSC/PAL encoder, and Camera interface, bus also advanced features like HDMI, video/image scaler, and video quality enhancer as post processor for displaying device. Separated from main bus, DDIBUS provides connectivity between internal blocks with sufficient bandwidth.

1.1 Feature

- LCD Interface
  - Progressive or Interlaced Digital Video Output
  - Support of STN LCD
  - Support of CCIR-601/656
  - Support of TFT LCD
    - 16/18/24 bits output formats
  - 16-Level Image Overlay & Chroma-Keying OSD
  - Programmable Timing for Different Display Panels
  - Supports the Picture-In-Picture
  - Color Space Converter
  - Three 256-Level Color Look-Up Tables
  - Dual-LCD Controller
- Memory-to-memory Scaler
  - YUV4:2:2, YUV4:2:0 (8bits), RGB444, RGB454, RGB555, RGB565
  - Support of Direct Interface with LCD interface
  - Support Rolling Mode
- NTSC/PAL Encoder Composite Output
  - NTSC-M/4.43, PAL-B/D/G/H/I/M/N/Combination N
  - 10-bit DAC
- HDMI
  - HDMI 1.3, HDCP 1.1, DVI 1.0 Complaint
  - Supports Video format:
    480p @59.94Hz/60Hz, 576p@50Hz
    720p @50Hz/59.94Hz/60Hz
    1080i @50Hz/59.94Hz/60Hz
    1080p @50Hz/59.94Hz/60Hz
  - Other various formats up to 148.5 MHz Pixel Clock
  - Supports Color Format : 4:4:4 RGB/YCbCr , 4:2:2 YCbCr 12-bit mode
  - Pixel Repetition (Up to x4)
  - Supports Bit Per Color : 8bit, 10bit ,12bit (16bit is not supported)
  - Dedicated block for CEC function
  - Supports Audio Sample packets, DSD packets, HBR packets and DST packets for audio
  - Integrated HDCP Encryption Engine for Video/Audio content protection
  - Not include DDC (recommend to use separate I2C)
- Camera Interface
  - CCIR 601/656, YUV4:2:2 (8bits), RGB555, RGB565
  - Support Interlace mode
  - Image Effector
  - Hardware Scaler
  - 4-Level Image Overlay, Chroma-Keying & Scaler for Preview
  - Various Input Format : YCbCr 4:2:2/4:2:0, RGB 4:2:2, 4:2:0, Bayer RGB
  - Color Space Dispatcher
  - Up-to-5M Pixels with Scaling
- Video/Image Quality Enhancer
  - De-interlacing
    - Edge-based/Motion-adaptive
    - File-mode processing
  - Noise Reduction
    - Spatio-temporal Engines
  - Edge Enhancing
  - Noise Measurement

- Histogram Measurement
- Contrast Enhancing
- Color Gamut Mapping

- LVDS
  - 35:7 data channel compression up to 560 Mbps on each LVDS channel
  - Up to 350 Mbytes/sec bandwidth
  - 6 LVDS output channels (5 data channels, 1clock channel)
  - Supports Audio Sample packets, DSD packets, HBR packets and DST packets for audio
  - Integrated HDCP Encryption Engine for Video/Audio content protection
  - Not include DDC (recommend to use separate I2C)

2 Bus Architecture

Figure 2.1 shows internal hardware architecture of DDIBUS.

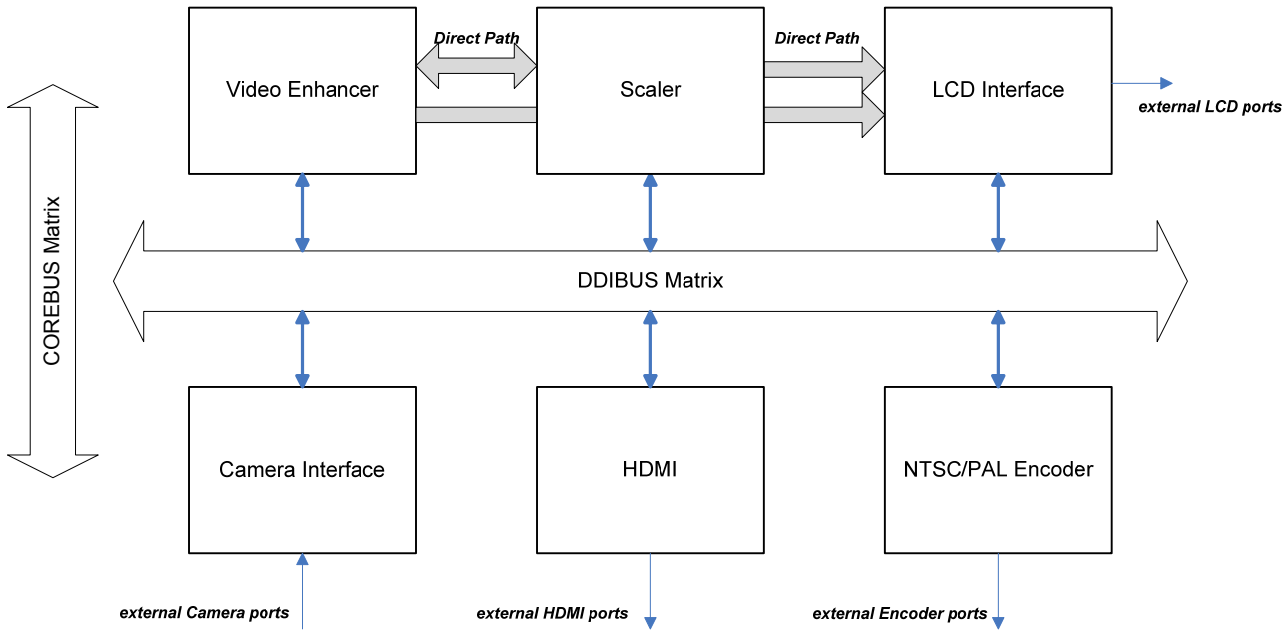


Figure 2.1 DDIBUS Hardware Architecture

3 Address and Register Map

The TCC8900 has various peripherals for specific video/image display/capture controllers or on-chip hardware components. These peripherals can be configured appropriately by its own registers that can be accessed through specially allocated address. These address maps are represented in the following table.

Refer to corresponding sections for detail information of each peripheral.

Base Address		Peripherals
0xF0200000	0xF0200000	LCD Controller 0
	0xF0200400	LCD LUT 0
	0xF0204000	LCD Controller 1
	0xF0204400	LCD LUT 1
0xF0210000		Scaler 0
0xF0220000		Scaler 1
0xF0230000		Camera Interface
0xF0240000	0xF0240000	NTSC Encoder
	0xF0240800	NTSC Encoder Controller
0xF0250000	0xF0250000	DDI Controller
0xF0251000	0xF0251000	DDIC Configuration
0xF0252000	0xF0252000	VIQE

4 LCD Interface

4.1 Overview

The LCD interface (LCDC) is used to send out image data from system memory to LCD (RGB interface type), NTSC/PAL encoder, HDMI, or LVDS by properly formatting the raw image data stored in the memory. The LCDC provides all the necessary control signals to interface directly to mono STN, color STN, TFT panels, NTSC/PAL encoders, HDMI, and LVDS. TCC8900 has 2 independent LCDCs. LCDC0 can manipulate only 1 input image. LCDC1 can manipulate 3 input images.

The features of the LCDC are as follows.

- supports Thin Film Transistor(TFT) color displays with 8-bit, 16-bit, 18-bit, 24-bit interface
- supports STN displays with 4 or 8-bit interface
- 1, 2 or 4 bits per pixel(bpp) displays for mono STN
- 8(332) /16 bpp(444dummy) color displays for color STN
- 16, 18, 24 bpp true-color non-palettized color displays for color TFT
- resolution programmable up to 4096 * 4096
- programmable timing for different display panels
- NTSC/PAL digital video encoder interface (CCIR601/656 interface)
- Supports color lookup table.
- Supports the overlay and alpha blending (2 overlay and 1 original image)
- Supports the image up scaling (x2, x3, x4, x8)
- Supports the Picture-In-Picture using the picture from the external device

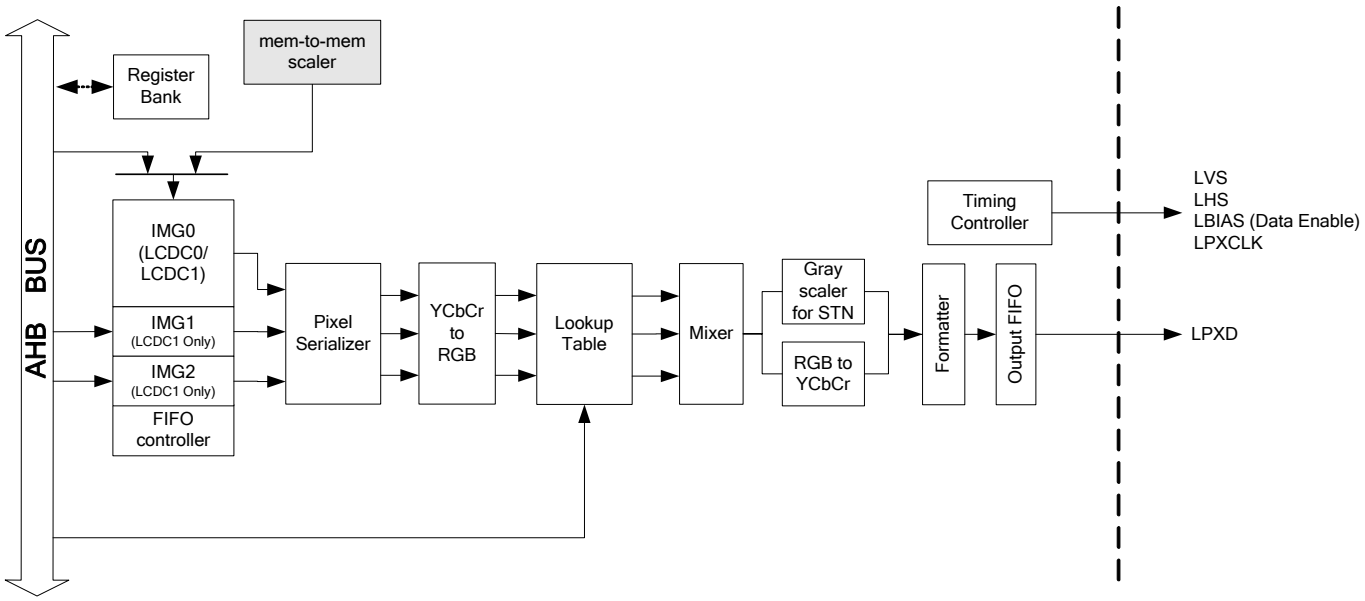


Figure 4.1 LCD Controller Block Diagram

The following key parameters can be programmed:

- horizontal front and back porch
- horizontal synchronization pulse width
- number of panel clocks per line
- vertical front and back porch
- vertical synchronization pulse width
- vertical synchronization pulse delay for STN mode
- number of lines per panel
- signal polarity
- panel clock frequency
- AC panel bias
- bits-per-pixel
- display type, STN mono/color or TFT
- STN 4 or 8-bit interface mode

- NTSC/PAL, Interlace/Non-interlace mode
- HDMI, Interlace/Non-interlace mode
- LVDS
- Overlay/alpha blending mode
- Image up scale ratio

The raw image sources stored in frame buffer are transferred to the LCDC's input FIFO, on a demand basis, using the AMBA AHB master interface.

The LCDC starts the DMA data transfer after it has been initialized and enabled. The DMA automatically performs burst word (64-bit) transfers, filling empty entries of the input FIFO. The data in the FIFO are fetched one entry at a time, and each 64-bit data is unpacked into appropriate pixel data formats (1, 2, 4, 8, or 16bpp) according to the raw image data format information. The frame buffer is in an off-chip memory area used to supply enough encoded pixel values to fill the entire screen one or more times. The pixel data buffer contains one encoded pixel values for each of the pixels present on the screen. The number of pixel data values depends on the size of the screen. The LCDC can fetch up to three raw image sources simultaneously and mixes them for making one frame image.

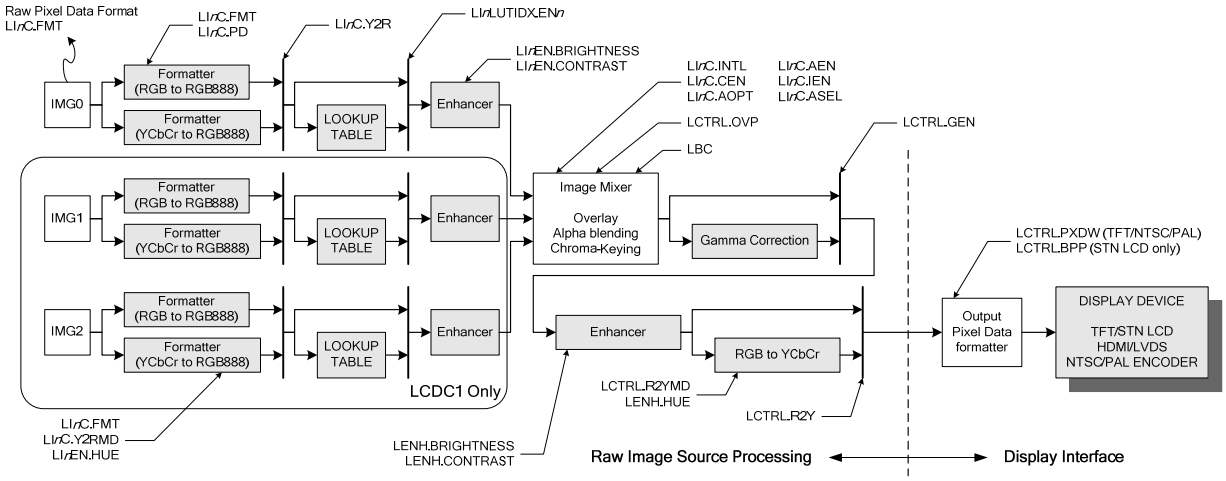


Figure 4.2 Overall Image Data Flow

Figure 4.2 shows the overall image data flow in the LCDC.

4.2 Raw Image Source Processing

4.2.1 Image Formats and Relation Between Images and Layers

For processing the raw image sources, the LCDC defines a virtual display. It is shown in Figure 4.3. Display size, which consists of “Display Width” and “Display Height”, is determined by LDS register. Actually, it becomes the active display size of the connected LCD or NTSC/PAL encoder.

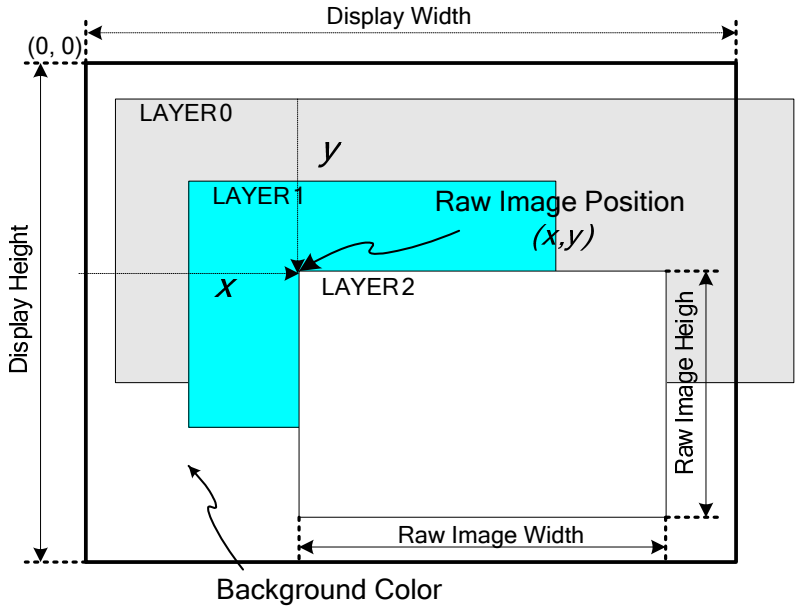


Figure 4.3 Virtual Display in the LCDC

The LCDC can have up to 3 raw image sources, which are named as Image 0, Image 1, and Image 2. IEN bits of each LI0C, LI1C, and LI2C register are used for enabling the corresponding raw image source. It can display simultaneously up to 3 layers, which are named as LAYER0, LAYER1, and LAYER2. LAYER2 is the top layer. Each layer is allocated for one of Image 0, Image 1, and Image 2. It is controlled by OP bits of LI0C register. Refer to the following table.

OP	LAYER2	LAYER1	LAYER0
0	Image 0	Image 1	Image 2
1	Image 0	Image 2	Image 1
2	Image 1	Image 0	Image 2
3	Image 2	Image 0	Image 1
4	Image 1	Image 2	Image 0
5*	Image 2	Image 1	Image 0

*) Default: OP = 5

Figure 4.4 Relationship between Layers and OP Bit

The area which is not overlapped by layers is filled with the background color. It is determined by the LBC register
The properties of a layer are determined by those of the corresponding raw image source; size, pixel data format, base address, position in the virtual display, etc.

BR=0	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
1BPP	P31	P30	P29	P28	P27	P26	P25	P24	P23	P22	P21	P20	P19	P18	P17	P16
2BPP	P15		P14		P13		P12		P11		P10		P9		P8	
4BPP	P7				P6				P5				P4			
8BPP	P3								P2							
BR=0	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1BPP	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
2BPP	P7		P6		P5		P4		P3		P2		P1		P0	
4BPP	P3				P2				P1				P0			
8BPP	P1								P0							

(a) The bitmap for 1bpp, 2bpp, 4bpp, 8bpp when BR is ‘0’

BR=1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1BPP	P24	P25	P26	P27	P28	P29	P30	P31	P16	P17	P18	P19	P20	P21	P22	P23
2BPP	P12		P13		P14		P15		P8		P9		P10		P11	
4BPP	P6				P7				P4				P5			
8BPP	P3								P2							
BR=0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1BPP	P8	P9	P10	P11	P12	P13	P14	P15	P0	P1	P2	P3	P4	P5	P6	P7
2BPP	P4		P5		P6		P7		P0		P1		P2		P3	
4BPP	P2				P3				P0				P1			
8BPP	P1								P0							

(b) The bitmap for 1bpp, 2bpp, 4bpp, 8bpp when BR is '0'

RGB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RGB332	R1			G1			B1		R0			G0			B0	
RGB444	A0				R0				G0				B0			
RGB565	R0					G0					B0					
RGB555	A0	R0					G0				B0					

(c) The bitmap for RGB332, RGB444, RGB565, and RGB555 formats.

RGB	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RGB666	IGNORED								A0						R0	
RGB888	A0								R0							
RGB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RGB666	R0				G0				B0							
RGB888	G0								B0							

(d) The bitmap for RGB666 and RGB888 formats.

Sequential	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
422SEQ	Cr								Y1							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
422SEQ	Cb								Y0							

(e) The bitmap for YCbCr 4:2:2 sequential format. The Cb and Cr affect to the Y0 and Y1.

Separated	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y	Y1								Y0							
Cb	Cb1								Cb0							
Cr	Cr1								Cr0							

(f) The bitmap for YCbCr 4:2:0 and 4:2:2 separated format. The Cb0 and Cr0 affect to the Y0 and. The Cb1 and Cr1 affect to the Y2s and Y3s.

Interleaved	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y	Y1								Y0							
C(type 0)	Cr0								Cb0							
C(type 1)	Cb0								Cr0							

(g) The bitmap for YCbCr 4:2:0 and 4:2:2 interleaved format.

Figure 4.5 Supported Pixel Data Format (BPP bits of LI*n*C register)

The pixel-data formats supported in the LCDC are shown in Figure 4.5. They are determined by the LI*n*C registers (*n=0, 1, or 2*). LI0C, LI1C, and LI2C are for Image 0, Image 1, and Image 2 respectively. 1bpp, 2bpp, and 4bpp have no color information and can be arranged in big-endian pixel order as well as little-endian pixel order. Others have color information and should be arranged in little-endian pixel order only.

Image 1 and Image 2 do not support YCbCr4:2:0 and YCbCr4:2:2, but support "sequential YCbCr4:2:2". If the pixel data format of a raw image source is YCbCr4:2:0 or YCbCr4:2:2, it needs three base addresses, which are specified by the LI*n*BA0, LI*n*BA1, and LI*n*BA2 register. Otherwise, it needs one base address, which is specified by the LI*n*BA0 register. Thus, when the pixel data format of Image 0 is YCbCr4:2:0 or YCbCr4:2:2, the base address of Y, Cb, and Cr image should be set to LI0BA0, LI0BA1, and LI0BA2 respectively. But, because Image 1 and Image 2 do not support YCbCr4:2:0 and YCbCr4:2:2, there are only LI1BA0 and LI2BA0 register.

Though a raw image source is not RGB888 format in RGB color space, it is internally converted to RGB888 as appending additional bits. When this conversion is occurred, the PD bits of LI*n*C determine whether the additional bits are filled with zero or MSB. The difference between them is shown in Figure 4.6.

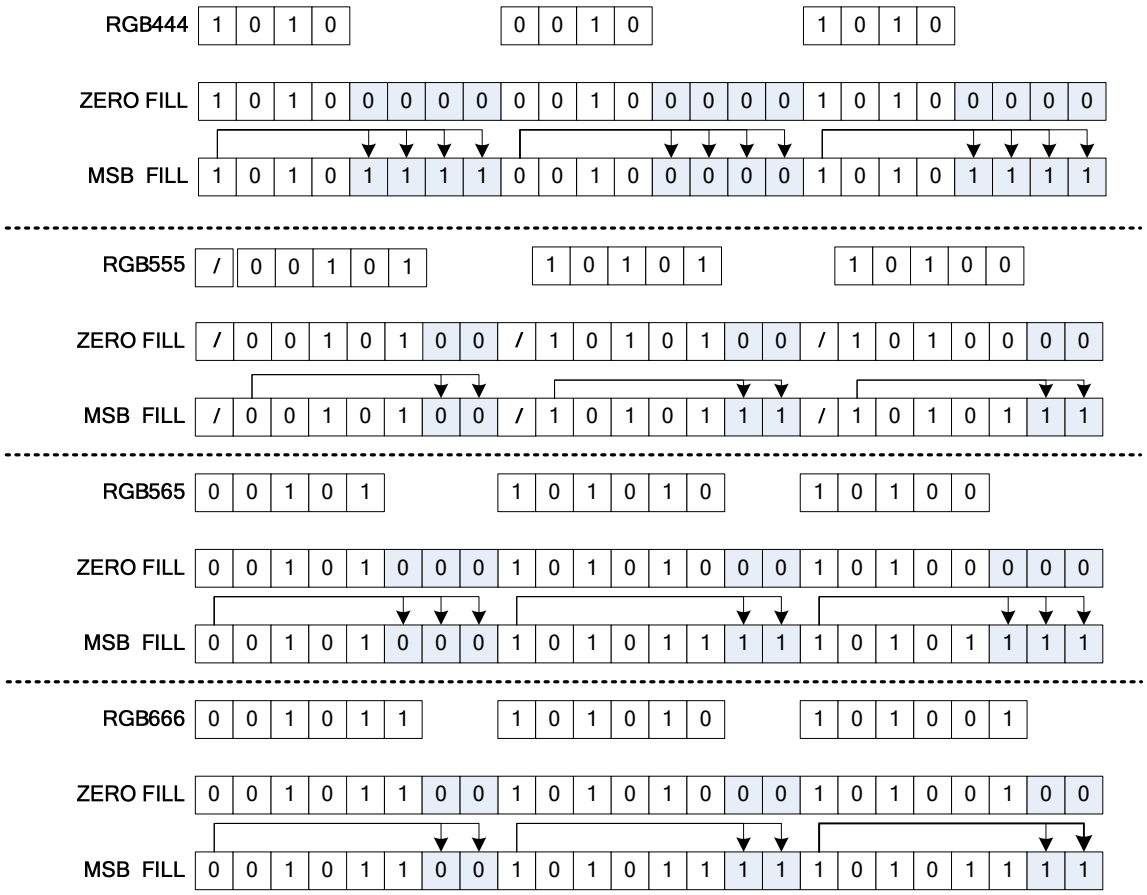


Figure 4.6 RGB2RGB888 Conversion

The origin of the virtual display is the top-left corner ((0, 0) in Figure 4.3) and “Raw Image Position” is the location of the top-left corner of a raw image within the virtual display. It is determined by the LI*n*P registers. “Raw Image Size”, which consists of “Raw Image Width” and “Raw Image Height”, is determined by the LI*n*S registers. Though the raw image sources are bigger than the size of the virtual display and/or the part of them exits the outside of the virtual display, the LCDC truncates an overlong part automatically.

And the LCDC has a simple scaler for each raw image source. It is controlled by LI*n*SC registers. But, to get the high-quality image, SCALER should be used instead of it.

Refer to Memory To Memory Scaler for more information about the image scaling.

4.2.2 Image Width, Height and Offset

The definition of frame window and image window is shown in the following figure. The frame window means that the raw image source and the image window is the window area on the frame window to be displayed. The “FP” stands for frame window pixel and “WP” stands for window pixel. The “FP(0,N)” is the pixel at the end of first line. All the pixels of the frame window should be on the linear and continuous address space.

The base address of LCD controller named as “LI*n*BA0” is the address of “WP(0,0)” and the image offset named as “LI*n*O” is the address difference between “FP(0,0)” and FP(1,0)”. ***The base address is byte-aligned and the image offset should be word-aligned”, the details are described in the explanation of the corresponding register.***

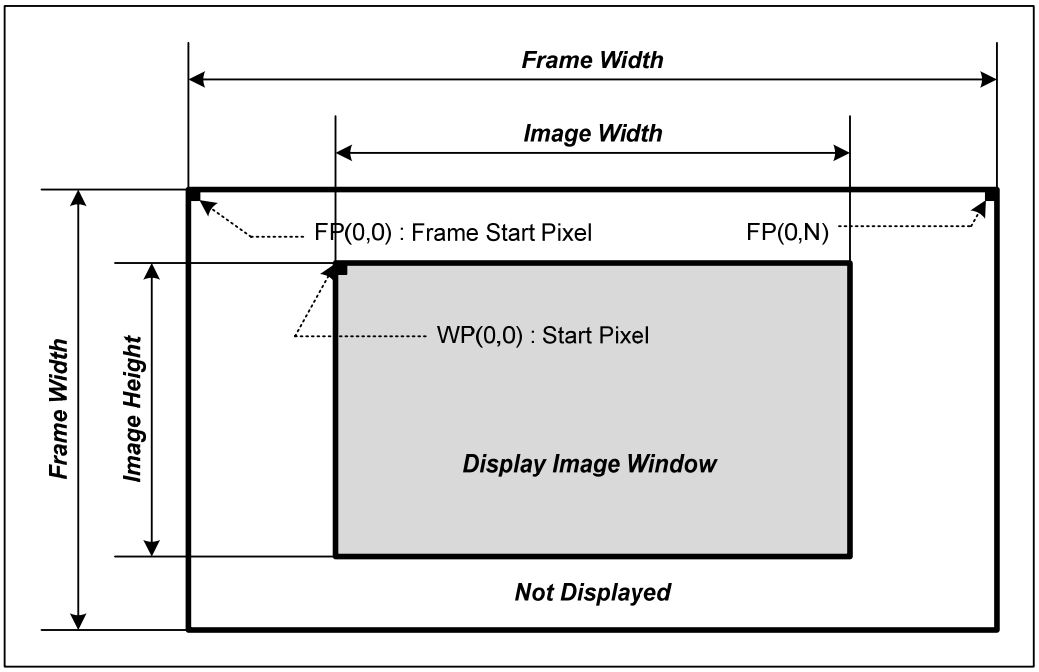


Figure 4.7 Definitions of Frame Window and Display Window

4.2.3 Conversion Between YCbCr Color Space and RGB Color Space

The internal operation in the LCD controller is designed for RGB color space. So, if the image source has the YCbCr color space, the source image should be converted to RGB color space. And if YCbCr color space is needed by display device, the internal graphic image should be converted to YCbCr color space.

The nominal ranges of “YCbCr” and “RGB” can be one of “0 ~ 255” or “16 ~ 235”. It is called “***Computer System Color***” and “***Studio Color***” for the 0 ~ 255 nominal range and 16 ~ 235 nominal range correspondingly.

The equations for color conversion can be dependent on the color types of source image and destination image.

4.2.4 Look-Up Table for Each Image Sources

The LCDC has the “24 bits x 256 entries” look-up table and it can be connected to one of Image 0, Image 1, and Image 2. Inputs of this table are RGB888 or YCbCr444 format as shown Figure 4.2. Each input pixel data addresses the table entry and the new pixel value is generated from the corresponding entry. Its entry consists of three 8-bit color values. The overall structure of Look-Up Table is shown in Figure 4.8. This table is controlled by LUT bits of the LIC2 register.

The table entries should be initialized before the LCDC uses it. Their address space is from 0xF0200400 to 0xF02007FC.

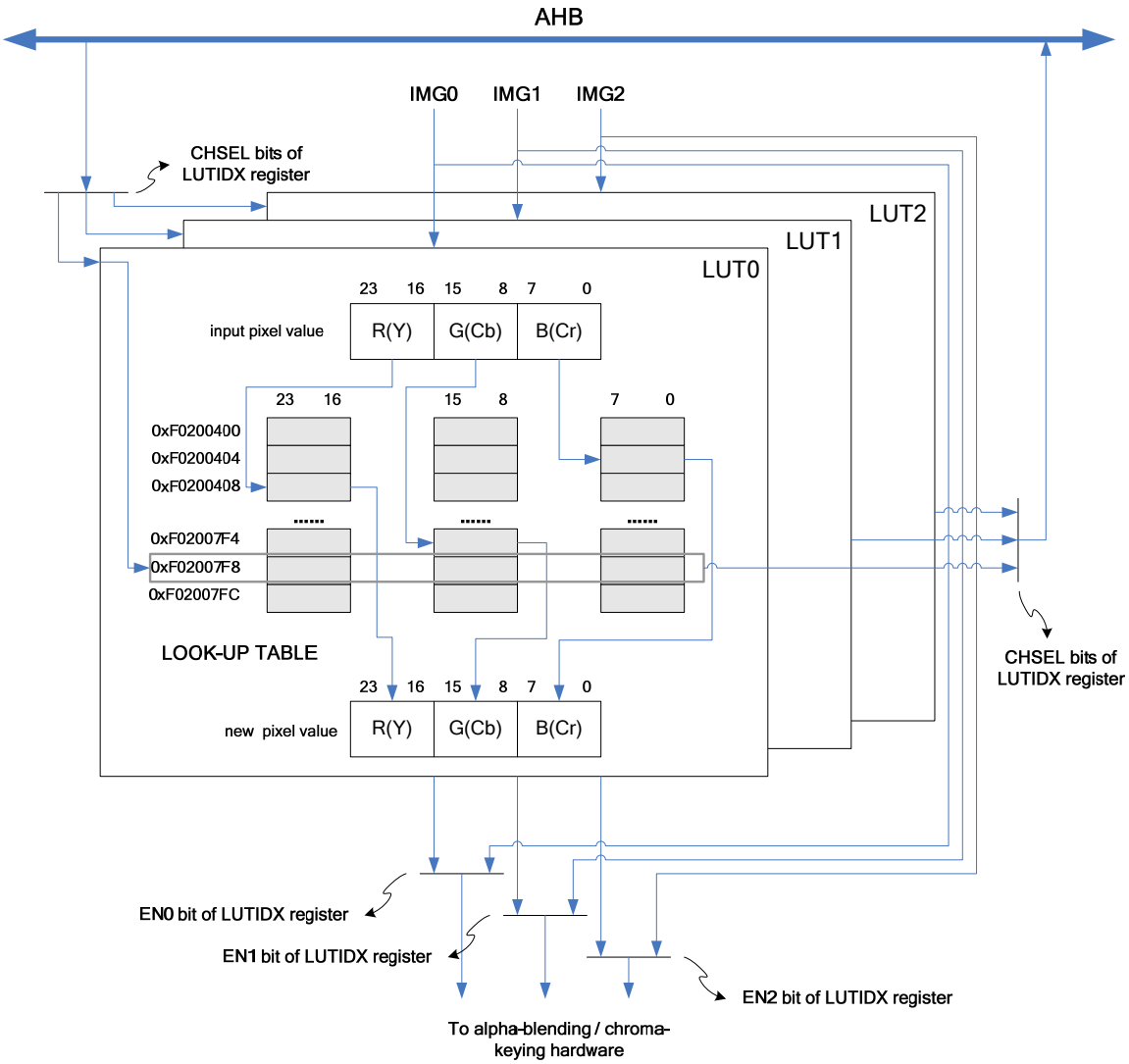


Figure 4.8 LCD Color Lookup Table

4.2.5 Alpha-blending with Multiple Layers

To process the overlapped region between layers, the LCDC supports the alpha-blending and the chroma-keying. When the alpha-blending and the chroma-keying are disabled, the top layer image is only shown in the overlapped region. Thus, LAYER1 is only shown in the overlapped region between the LAYER0 and the LAYER1.

The alpha-blending combines the overlay image (top layer) with the main image (bottom layer) according to the alpha value. The alpha value is specified by alpha bits in the pixel data of the overlay image and the alpha registers. The AEN2 bit of LCTRL register determines whether the LAYER2 enables the alpha-blending and the AEN1 bit determines whether LAYER1 enables the alpha-blending. Whether the alpha bits in the pixel data or alpha-registers are used as the alpha-value depends on the value of the SA1 bit of the LK1 register for the LAYER1 and the SA2 bit of the LK2 register for the LAYER2.

If the overlay image consists of the pixel data without the alpha value, you should use alpha-registers as the alpha-value (SA2 = 1 and/or SA1 = 1). The LAYER2 uses the A20 bits of LK2 register as the alpha-value and LAYER1 uses the A10 bits of LK2 register.

4.2.6 Chroma-Keying with Multiple Layers

Usually, a raw image source is the rectangular shape. Chroma-keying is used for clipping the arbitrary shape from a raw image source. In other words, chroma-keying disables the overlay process for selected pixels. They are identified by control registers, which are LK1, LK2. The LK1 is for the LAYER1 and the LK2 is for the LAYER2.

The overlay disabled pixel becomes the transparent pixel; the pixel of background layer is shown in the display.

For LAYER1, whether a pixel is overlaid or not depends on

$$OverlayDisabledPixel = ((Pixel(RorY) \,\&\, LKM1.MKR1) == (LK1.KR1 \,\&\, LKM1.MKR1)) \,\&$$
$$((Pixel(GorU) \,\&\, LKM1.MKG1) == (LK1.KG1 \,\&\, LKM1.MKG1)) \,\&$$
$$((Pixel(BorV) \,\&\, LKM1.MKB1) == (LK1.KB1 \,\&\, LKM1.MKB1))$$

$$OverlayEnabledPixel = ((Pixel(RorY) \,\&\, LKM1.MKR1) != (LK1.KR1 \,\&\, LKM1.MKR1)) \,|$$
$$((Pixel(GorU) \,\&\, LKM1.MKG1) != (LK1.KG1 \,\&\, LKM1.MKG1)) \,|$$
$$((Pixel(BorV) \,\&\, LKM1.MKB1) != (LK1.KB1 \,\&\, LKM1.MKB1))$$

For LAYER2, whether a pixel is overlaid or not depends on

$$OverlayDisabledPixel = ((Pixel(RorY) \,\&\, LKM2.MKR1) == (LK2.KR1 \,\&\, LKM2.MKR1)) \,\&$$
$$((Pixel(GorU) \,\&\, LKM2.MKG1) == (LK2.KG1 \,\&\, LKM2.MKG1)) \,\&$$
$$((Pixel(BorV) \,\&\, LKM2.MKB1) == (LK2.KB1 \,\&\, LKM2.MKB1))$$

$$OverlayEnabledPixel = ((Pixel(RorY) \,\&\, LKM2.MKR1) != (LK2.KR1 \,\&\, LKM2.MKR1)) \,|$$
$$((Pixel(GorU) \,\&\, LKM2.MKG1) != (LK2.KG1 \,\&\, LKM2.MKG1)) \,|$$
$$((Pixel(BorV) \,\&\, LKM2.MKB1) != (LK2.KB1 \,\&\, LKM2.MKB1))$$

4.2.7 Gamma Correction Function

Gamma Correction Function uses 16 correction values for each channel (GR0~15, GG0~15 and GB0~15). Each correction values are consisted of 4 bits and are represented two's complements format (value range: -8 ~ +7). Gamma correction function uses 17 correction points (16bands).

Nth Correction Point:
N<8 : ((256/N), (256/N)+LGR[N]*4)
N=8 : (32, 32+((LGR[7]+LGR[8])/2))
N>9 : (256/(N-1), (256/(N-1))+LGR[N-1]*4)

Gamma Correction Function is shown in Figure 4.9. Some examples are shown in Figure 4.10 and Figure 4.11.

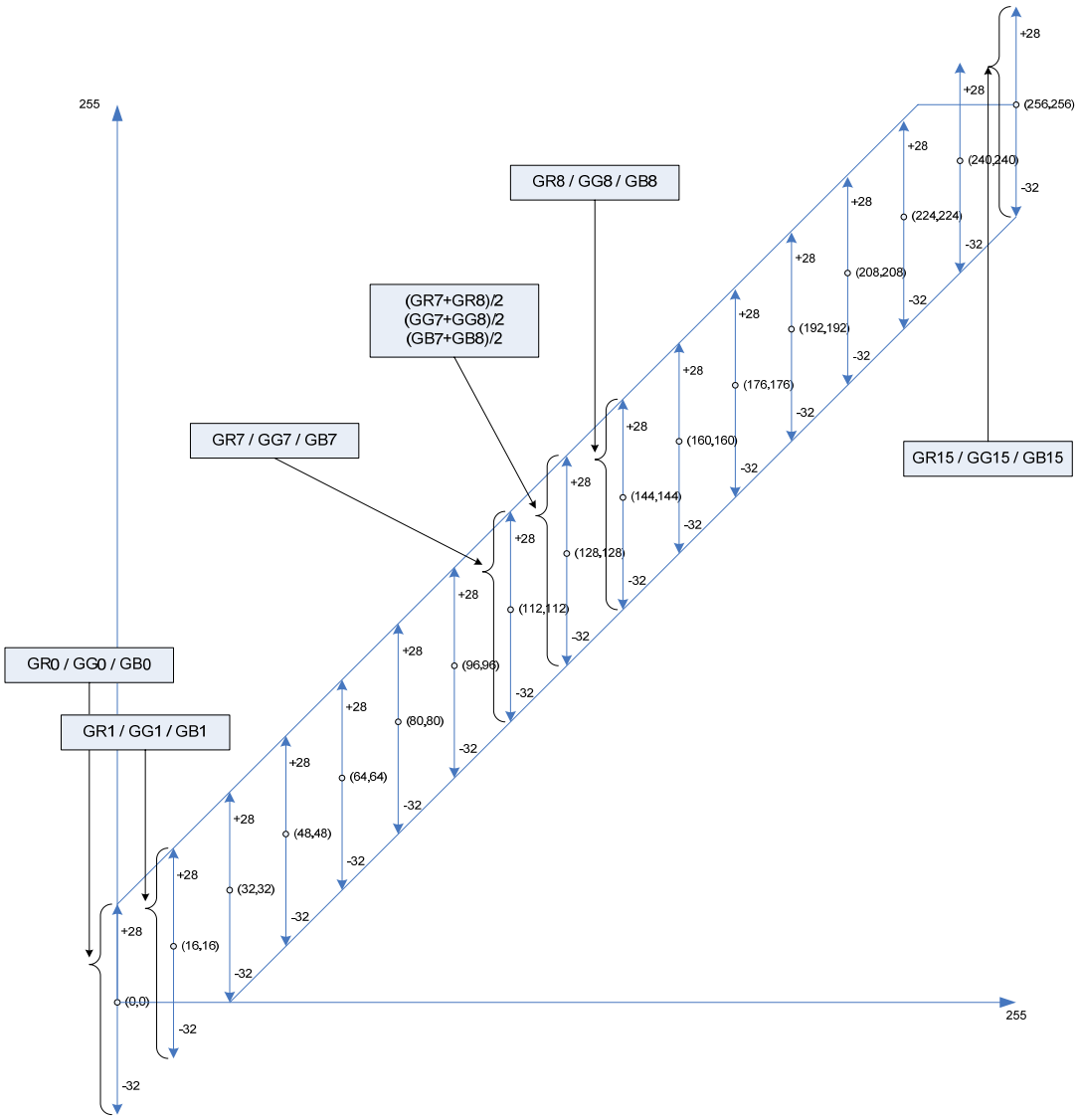


Figure 4.9 LCD Gamma Correction Function

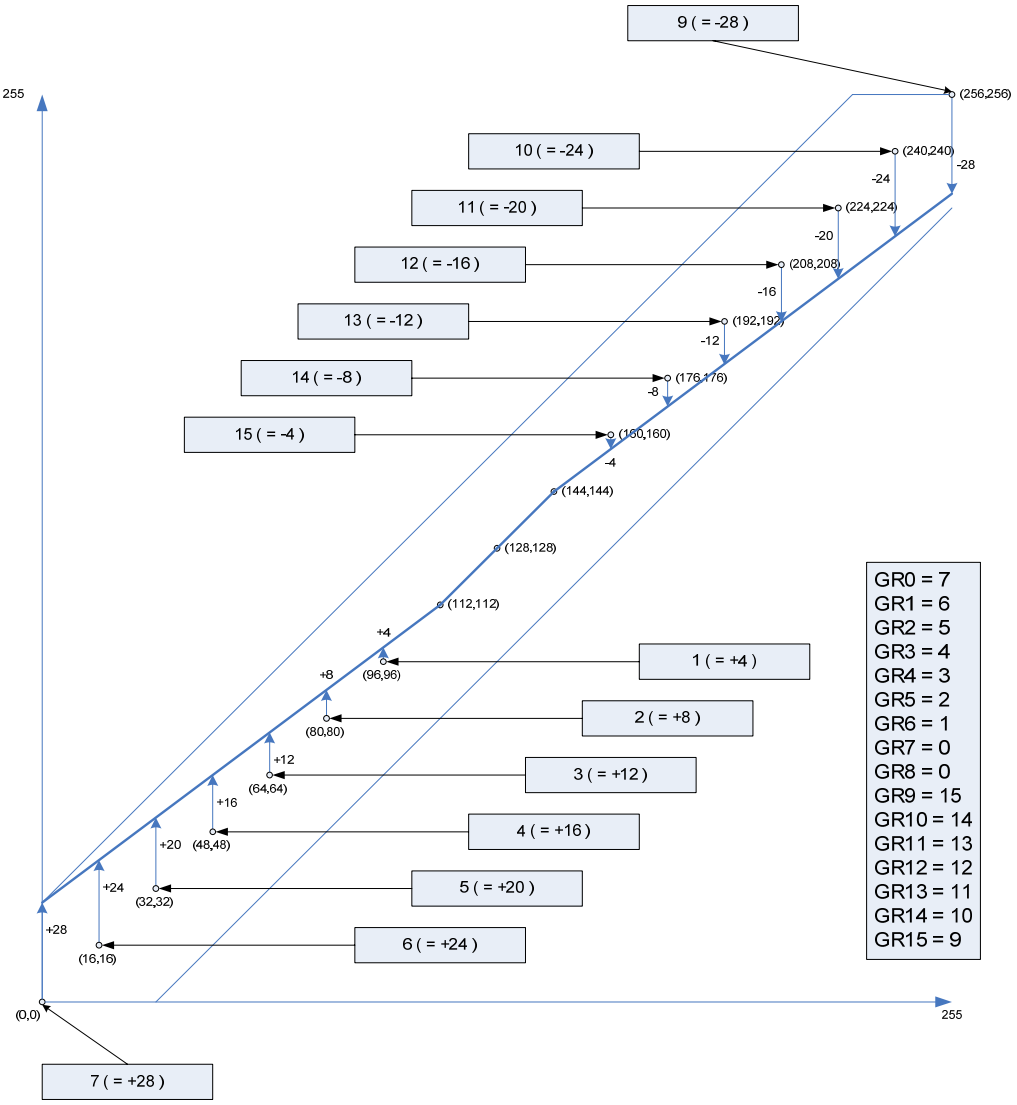


Figure 4.10 LCD Gamma Correction Function Example 1

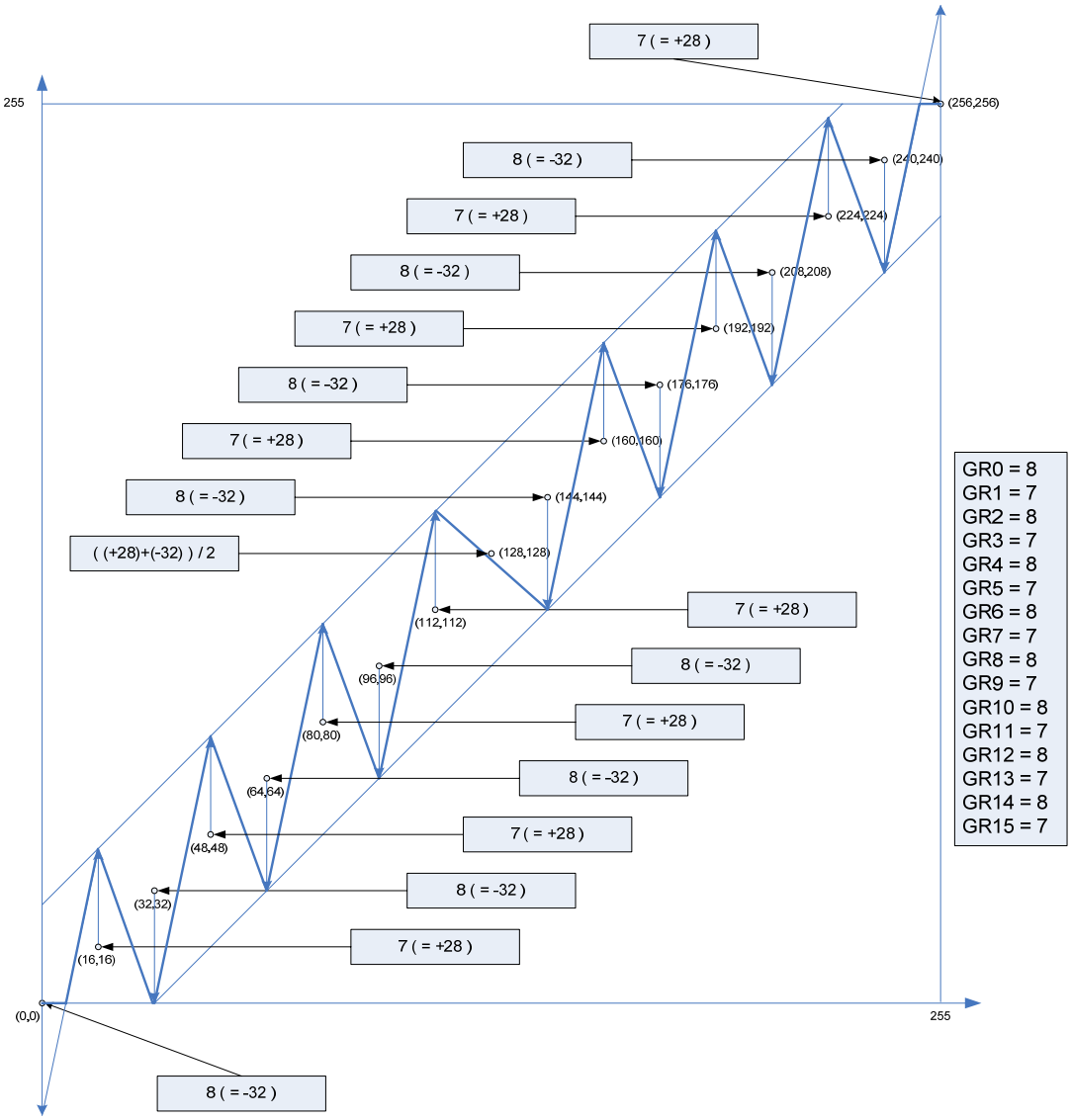


Figure 4.11 LCD Gamma Correction Function Example 2

4.2.8 Contrast and Brightness Adjust Function

Output_value = MAX(MIN((input_value – 128) x contrast + 128 + brightness, 255), 0)

4.3 Display Interface

4.3.1 STN-LCD

The LCDC generates VSYNC, HSYNC, PXCLK, ACBIAS, and PXDATA signals for STN LCD driver.

The output data format for STN LCD is determined by the interface data width and the color depth of STN LCD. The interface data width is determined by PXDW bits of LCTRL register. 4bits and 8bits are only supported for STN LCD. The color depth is BPP bits of LCTRL register and it is not related to a raw image pixel data format.

The timing diagram for STN mode is shown in Figure 4.12. VSYNC and HSYNC pulse are controlled by the configurations of the LPC field of LHTIME and FLC field of LVTIME1 and LVTIME2. Each field is related to the LCD size and display mode.

In 1bpp, 2bpp, 4bpp:

$$LPC = (HorizontalDisplaySize)/(PixelDataWidth) - 1$$

In 8pp and 16bpp (RGB):

$$LPC = (3 \times HorizontalDisplaySize)/(PixelDataWidth) - 1$$

Pixel data width is determined by PXDW of LCTRL register. In the case of STN LCD mode, it must be 4 or 8-bit width.
ACBIAS signal is used by an LCD driver in the STN LCD module to alternate the polarity of the row and column voltage used to turn the pixel on and off. It is controlled by the ACDIV field of LCLKDIV register:

$$f_{ACBIAS} = f_{HSYNC}/(2 \times ACDIV)$$

PXCLK frequency is determined by the CLKDIV field of LCLKDIV register as follows. The minimum value of CLKDIV is 3 in STN mode.

$$f_{PXCLK} = f_{LCLK}/(2 \times CLKDIV) \quad (1)$$

VSYNC frequency is related to the field of FPW, LSWC, LEWC, LPC, and FLC as well as HCLK and PXCLK.

$$f_{VSYNC} = f_{PXCLK}/[\{(LPW+1)+(LPC+1)+(LSWC+1)+(LEWC+1)\} \times \{(FLC+1)+(FPW+1)\}]$$

Therefore, if FR is the required refresh rate, f_{PXCLK_REQ}, which is the required PXCLK, is as following.

$$f_{PXCLK\_REQ} = FR \times [\{(LPW+1)+(LPC+1)+(LSWC+1)+(LEWC+1)\} \times \{(FLC+1)+(FPW+1)\}] \quad (2)$$

The LCDC contains look-up registers for STN LCD with RGB332 or 2BPP to support palletized color. They are LLUTR, LLUTG, and LLUTB. When RGB332 STN LCD is used, LLUTR register allows any 8 red levels to be selected from 16 possible red levels. LLUTG register is for green color and LLUTB register is for blue color. But, when 2BPP STN LCD is used, LLUTG register is only used. Refer to Table 4.1. When STN LCD type is not RGB332 or 2BPP, look-up registers are not used. STN LCD type is determined by BPP bits of LCTRL register.

Table 4.1 STN LCD Palette Address

Pixel Value	Red Palette	Green Palette	Blue Palette
0	LLUTR[3:0]	LLUTG[3:0]	LLUTG[3:0]
1	LLUTR[7:4]	LLUTG[7:4]	LLUTG[7:4]
2	LLUTR[11:8]	LLUTG[11:8]	LLUTG[11:8]
3	LLUTR[15:12]	LLUTG[15:12]	LLUTG[15:12]
4	LLUTR[19:16]	LLUTG[19:16]	N/A
5	LLUTR[23:20]	LLUTG[23:20]	N/A
6	LLUTR[27:24]	LLUTG[27:24]	N/A
7	LLUTR[31:28]	LLUTG[31:28]	N/A

The LCDC contains the dithering pattern registers for STN LCD: a 48-bit modulo 7 dithering pattern register (LDP7L and LDP7H), a 32-bit modulo 5 dithering pattern register (LDP5), a 16-bit modulo 4 dithering pattern register (LDP4), and a 16-bit modulo 3(LDP3) dithering pattern register. These dithering pattern registers can contain the programmable pre-dithered pattern values for each duty cycle ratio.

The LDP7H and LDP7L contain 5 pre-dithered patterns for 1/7, 3/7, 4/7, 5/7, and 6/7 duty cycle rate. Each field of LDP7H and LDP7L is 7-bit long. The LDP5 has 4 pre-dithered pattern fields for 1/5, 2/5, 3/5, and 4/5 duty cycle rate. Each field of

LDP5 is 5-bit long. The LDP4 has 3 pre-dithered pattern fields for 1/4, 1/2(=2/4), and 3/4 duty cycle rate, and each field is 4-bit long. Likewise, the LDP3 has 2 fields for 1/3 and 2/3 duty cycle rate with 3-bit length.

Note that the pre-dithered data for 1 and 0 is not defined in the dithering pattern register, because these values are implemented with VDD and VSS condition. The pre-dithered value is pixel value or palletized color value according to STN LCD type. Therefore, if BPP bits of LCTRL register represent RGB332 or 2BPP, pre-dithered value is palletized color. Otherwise, it is pixel value.

Table 4.2 STN LCD Dithering Pattern Register map

Pre-dithered value	Dithering register	Duty cycle ratio
0	all 0s	0
1	DP1_7	1/7
2	DP1_5	1/5
3	DP1_4	1/4
4	DP1_3	1/3
5	DP2_5	2/5
6	DP3_7	3/7
7	DP2_4	1/2
8	DP4_7	4/7
9	DP3_5	3/5
10	DP2_3	2/3
11	DP5_7	5/7
12	DP3_4	3/4
13	DP4_5	4/5
14	DP6_7	6/7
15	all 1s	1

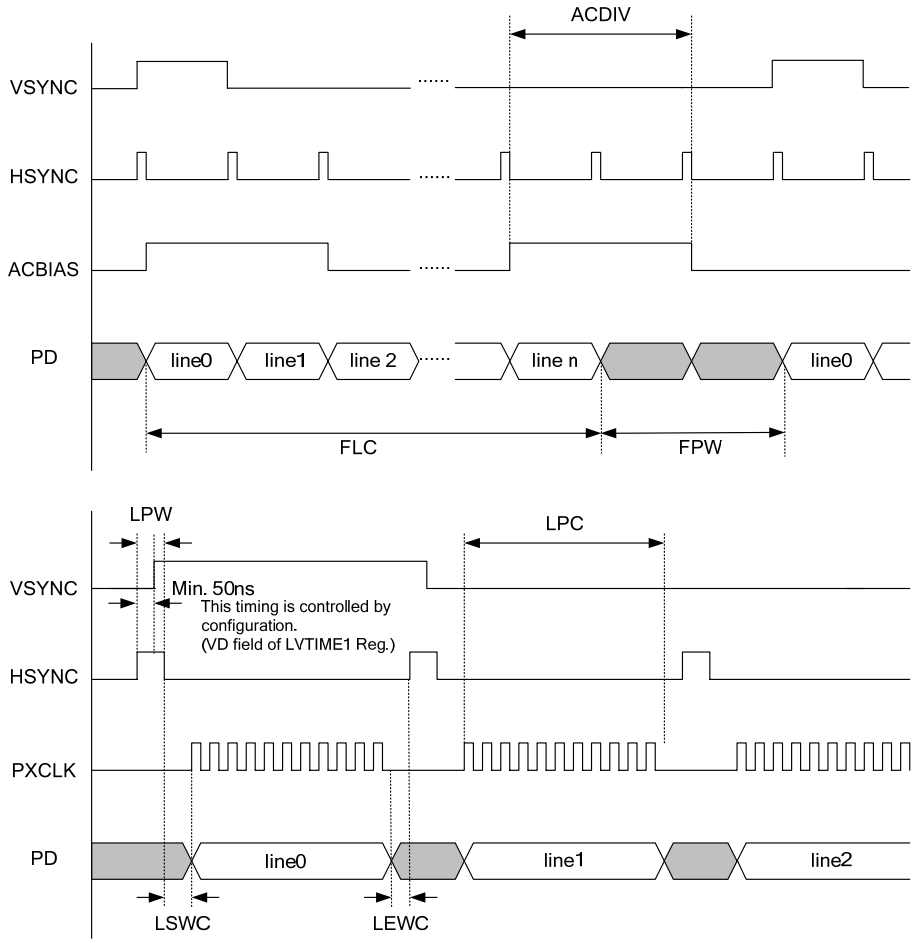


Figure 4.12 STN Mode Timing Diagram

EXAMPLE)

For a monochrome STN LCD, 4-bit interface panel, 4 pixels are captured by the panel in every panel clock cycle. Table 4.3 gives the major registers to be programmed for supporting 4-bit interface STN LCD. LCLK and Refresh rate are examples only. And LSWC, LEWC, LPW, and FPW are STN LCD panel dependent.

LCLK = 20 MHz, Refresh rate = 60 Hz
PXDW* = 0 (4bits), BPP* = 2 (4bpp) , DP* = 0 (one pixel data per one pixel cycle)
NI = 1, TV* = 0, TFT* = 0, STN* = 1
LSWC* = LEWC* = LPW* = FPW* = 1 (STN LCD dependent)

Table 4.3 Monochrome STN LCD (4bits, 1BPP) example

Width (pixel)	Height (pixel)	LPC*	FLC*	DHSIZE*	DVSIZE*	f_{PXCLK_REQ}	CLKDIV*	f_{PXCLK}***
160	160	39	159	160	160	0.393	25	0.4
160	200	39	199	160	200	0.491	20	0.5
320	200	79	199	320	200	0.973	10	1

*) control registers to be programmed. **) Refer to expression (2). ***) Refer to expression (1).

4.3.2 TFT-LCD

The LCDC supports 16, 18, 24 bpp true-color non-palletized color displays for color TFT LCD. It generates the control signals for LCD driver such as, VSYNC, HSYNC, PXCLK, PXDEN (ACBIAS) and PXDATA. The timing diagram of TFT mode is shown in Figure 4.13

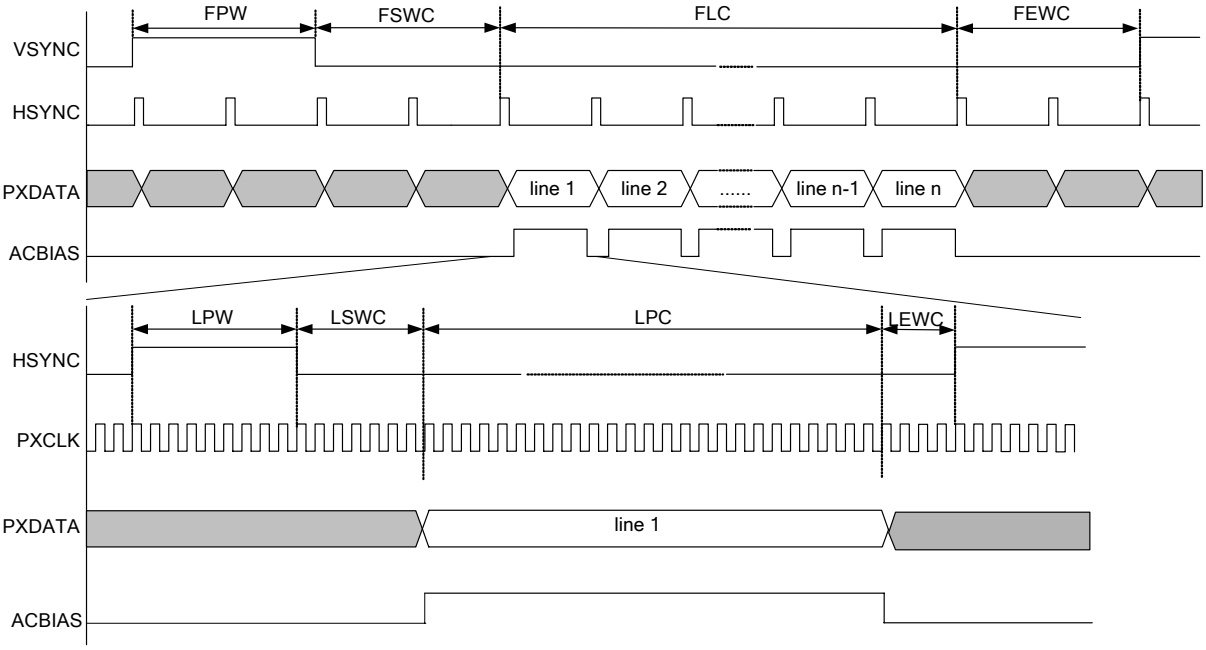


Figure 4.13 TFT Mode Timing Diagram

The VSYNC and HSYNC frequency is controlled by the LPC and FLC field.

$$LPC = (HorizontalDisplaySize) - 1$$
$$FLC = (VerticalDisplaySize) - 1$$

And PXCLK frequency is determined by the CLKDIV value.

$$f_{PXCLK} = f_{LCLK} / (2 \times CLKDIV) \quad (3)$$

The frequency of VSYNC signal is the frame rate. So the frame rate can be calculated as follows:

$$f_{HSYNC} = \frac{f_{PXCLK}}{\{(LPW+1)+(LPC+1)+(LSWC+1)+(LEWC+1)\}}$$

$$f_{VSYNC} = \frac{f_{HSYNC}}{\{(FPW+1)+(FPC+1)+(FSWC+1)+(FEWC+1)\}}$$

Therefore, if FR is the required refresh rate in TFT mode, f_{PXCLK_REQ}, which is the required PXCLK, is as following.

$$f_{PXCLK\_REQ} = FR \times \{(FPW+1)+(FPC+1)+(FSWC+1)+FEWC+1)\} \times \{(LPW+1)+(LPC+1)+(LSWC+1)+(LEWC+1)\} \quad (4)$$

Example)

For TFT LCD(RGB565), if LCLK = 80MHz and Refresh rate = 60Hz,

PXDW* = 0x4, YCbCr* = 0, BPP* = 0x4, DP* = 0, NI* = 1, TV* = 0, TFT* = 1, STN* = 0
LSWC* = LEWC* = LPW* = 3 (TFT LCD dependent)
FSWC* = FEWC* = FPW* = 1 (TFT LCD dependent)

Table 4.4 TFT LCD (RGB565) Example

Width (pixel)	Height (pixel)	LPC*	FLC*	DHSIZE*	DVSIZE*	F_{PXCLK_REQ}**	CLKDIV*	f_{PXCLK}***
176	220	175	219	176	220	2.55	15	2.67
240	320	239	319	240	320	4.93	8	5
640	480	639	479	640	480	19.01	2	20

*) control registers to be programmed. **) Refer to expression (4). ***) Refer to expression (3).

4.3.3 NTSC/PAL Interface

The LCDC can generate the control signals for 8-bit or 16-bit NTSC/PAL encoder. The supporting mode is CCIR601/656 interlace/non-interlace.

For NTSC/PAL interface, TV bit of LCTRL register must be set. Registers used in this mode are similar to those in TFT mode except for LVTIME1 and LVTIME2 registers; LVTIME1 is for odd field and LVTIME2 is for Even field. And these registers value is not based on HSYNC, but based on half of HSYNC. For example, if FPW of LVTIME1 is 3, pulse width of VSYNC on odd field is not 4 HSYNC cycles, but 2 HSYNC cycles. And if FPW of LVTIME1 is 4, it is 2.5 HSYNC cycles.

Interlace/Non-interlace mode can be configured by NI bit of LCTRL register. Figure 4.14 and Figure 4.15 show the timing diagram of NTSC and PAL interlace mode. In non-interlace mode, odd field sync signals are repeated instead.

The CCIR656 mode can be configured by 656 bit of LCTRL register. This mode uses 8 interface signals which have SYNC information as well as 8-bit pixel data. Thus, the additional sync signals are not needed. The output pixel clock must be 27 MHz. Figure 4.16 shows the embedded sync information in the CCIR656 mode.

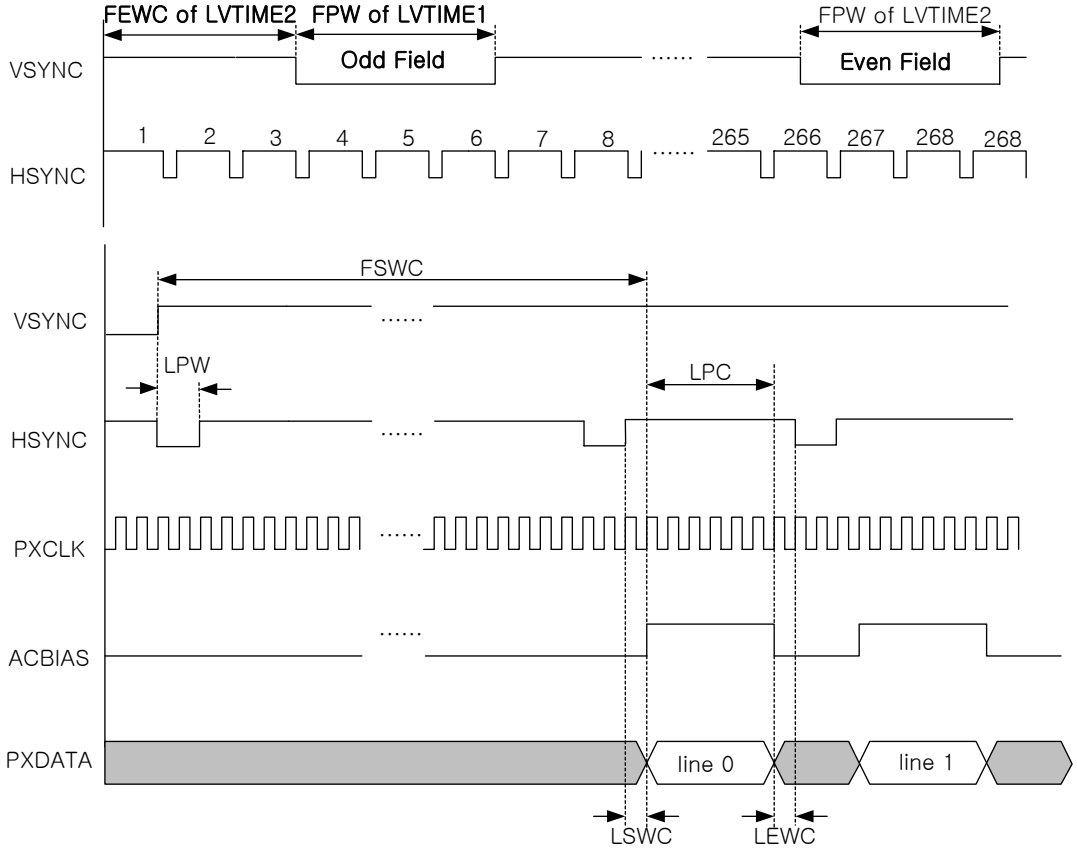


Figure 4.14 NTSC Interlace Mode Timing Diagram

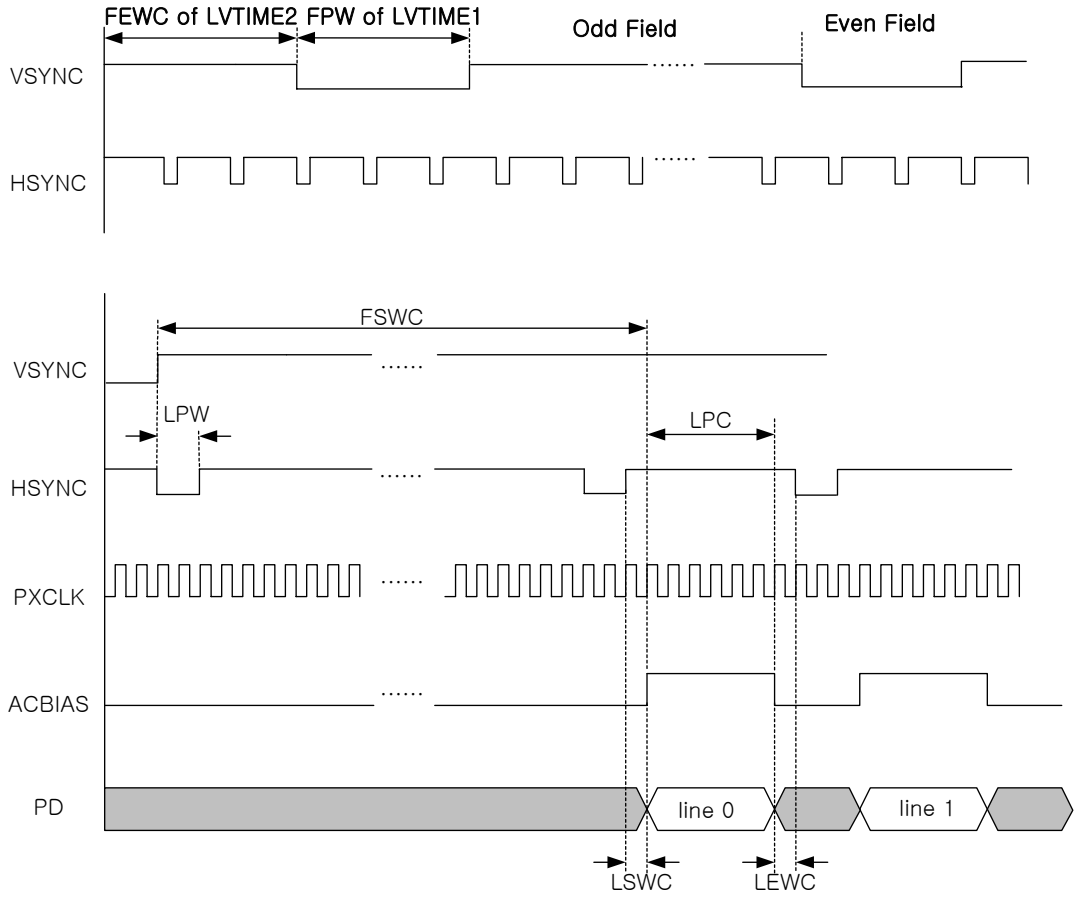
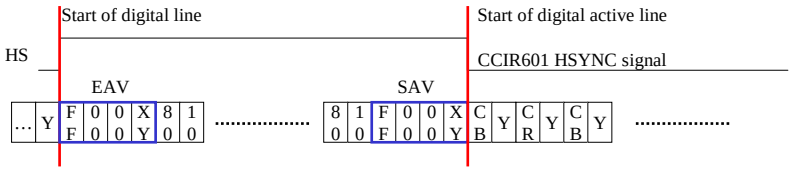


Figure 4.15 PAL Interlace Mode Timing Diagram



	8-bit data							
	D7 (MSB)	D6	D5	D4	D3	D2	D1	D0
preamble	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
status	1	F	V	H	P3	P2	P1	P0

•Status word define
- •F=‘0’ for field 1, ‘1’ for field 2
- •V=‘1’ during vertical blanking
- •H=‘0’ at SAV, ‘1’ at EAV

•Protection bits
- •P3=V xor H
- •P2=F xor H
- •P1=F xor V
- •P0=F xor V xor H

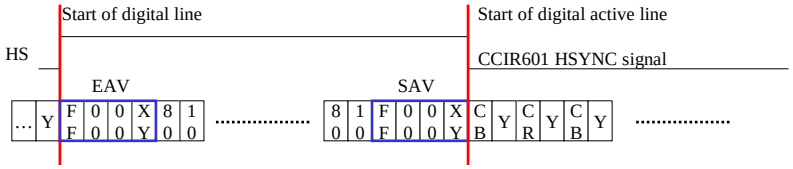
* HSYNC is active low mode

Figure 4.16 CCIR656 Embedded sync. Information

Figure 4.17 shows the standard timing diagram of CCIR.656 format.

In NTSC mode, one line consists of 1716 pixel clock cycles. The horizontal blanking area occupies 276 pixel clock cycles, and 8 pixel clock cycles within the horizontal blanking is SAV and EAV code. Active area occupies1440 pixel clock cycles. One frame is composed of 525 lines and is divided into two fields, odd and even.

In PAL mode, one line consists of 1728 pixel clock cycles. The horizontal blanking area occupies 286 pixel clock cycles, and 8 pixel clock cycles within the horizontal blanking is SAV, EAV code. Active area occupies 1440 pixel clock cycles. One frame is composed 625 lines and is divided into two fields, odd and even.



	8-bit data							
	D7 (MSB)	D6	D5	D4	D3	D2	D1	D0
preamble	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
status	1	F	V	H	P3	P2	P1	P0

•Status word define
- •F=‘0’ for field 1, ‘1’ for field 2
- •V=‘1’ during vertical blanking
- •H=‘0’ at SAV, ‘1’ at EAV

•Protection bits
- •P3=V xor H
- •P2=F xor H
- •P1=F xor V
- •P0=F xor V xor H

* HSYNC is active low mode

Figure 4.17 CCIR656 Format Diagram

4.4 Register Description

Table 4.5 LCDC Register Map (Base Address = 0xF0200000/0xF0204000)

Name	Address	Type	Reset	Description
LCTRL	0x00	R/W	0x0080004a	LCD Control Register
LBC	0x04	R/W	0x00000000	LCD Background Color Register
LCLKDIV	0x08	R/W	0x00000000	LCD Clock Divider Register
LHTIME1	0x0C	R/W	0x00000000	LCD Horizontal Timing Register 1
LHTIME2	0x10	R/W	0x00000000	LCD Horizontal Timing Register 2
LVTIME1	0x14	R/W	0x00000000	LCD Vertical Timing Register 1
LVTIME2	0x18	R/W	0x00000000	LCD Vertical Timing Register 2
LVTIME3	0x1C	R/W	0x00000000	LCD Vertical Timing Register 3
LVTIME4	0x20	R/W	0x00000000	LCD Vertical Timing Register 4
LLUTR	0x24	R/W	0x00000000	LCD Lookup Register for Red
LLUTG	0x28	R/W	0x00000000	LCD Lookup Register for Green
LLUTB	0x2C	R/W	0x00000000	LCD Lookup Register for Blue
LDP7L	0x30	R/W	0x4d2b3401	LCD Modulo 7 Dithering Pattern (low)
LDP7H	0x34	R/W	0x0000003f	LCD Modulo 7 Dithering Pattern (high)
LDP5	0x38	R/W	0x1d0b0610	LCD Modulo 5 Dithering Pattern Register
LDP4	0x3C	R/W	0x00000768	LCD Modulo 4 Dithering Pattern Register
LDP3	0x40	R/W	0x00000034	LCD 3-bit Dithering Pattern Register
LCP1	0x44	R/W	0x000000ff	LCD Clipping Register1
LCP2	0x48	R/W	0x000000ff	LCD Clipping Register2
LDS	0x4C	R/W	0x00000000	LCD Display Size Register
LSTATUS	0x50	R/CLR	0x00000000	LCD Status Register
LIM	0x54	R/W	0x0000001f	LCD Interrupt Register.
LGR0	0x58	R/W	0x00000000	LCD Gamma Correction Register 0 for Red Color
LGR1	0x5C	R/W	0x00000000	LCD Gamma Correction Register 1 for Red Color
LGG0	0x60	R/W	0x00000000	LCD Gamma Correction Register 0 for Green Color
LGG1	0x64	R/W	0x00000000	LCD Gamma Correction Register 1 for Green Color
LGB0	0x68	R/W	0x00000000	LCD Gamma Correction Register 0 for Blue Color
LGB1	0x6C	R/W	0x00000000	LCD Gamma Correction Register 1 for Blue Color
LENH	0x70	R/W	0x00000020	LCD Color Enhancement Register

Name	Address	Type	Reset	Description
LI0C	0x78	R/W	0x00000000	LCD Image 0 Control Register
LI0P	0x7C	R/W	0x00000000	LCD Image 0 Position Register
LI0S	0x80	R/W	0x00000000	LCD Image 0 Size Register
LI0BA0	0x84	R/W	0x00000000	LCD Image 0 Base Address 0 Register.
LI0CA	0x88	R/W	0x00000000	LCD Image 0 Current Address Register.
LI0BA1	0x8C	R/W	0x00000000	LCD Image 0 Base Address 1 Register
LI0BA2	0x90	R/W	0x00000000	LCD Image 0 Base Address 2 Register
LI0O	0x94	R/W	0x00000000	LCD Image 0 Offset Register
LI0SR	0x98	R/W	0x00000000	LCD Image 0 Scale ratio
LI0A	0x9C	R/W	0x00000000	LCD Image 0 Alpha Configuration Register
LI0KR	0xA0	R/W	0x00000000	LCD Image 0 Keying Register for RED or LUMA(Y)
LI0KG	0xA4	R/W	0x00000000	LCD Image 0 Keying Register for BLUE or CHROMA(Cb)
LI0KB	0xA8	R/W	0x00000000	LCD Image 0 Keying Register for GREEN or CHROMA(Cr)
LI0EN	0xAC	R/W	0x00000000	LCD Image 0 Enhancement Register
LI1C[1]	0xB0	R/W	0x00000000	LCD Image 1 Control Register
LI1P[1]	0xB4	R/W	0x00000000	LCD Image 1 Position Register
LI1S[1]	0xB8	R/W	0x00000000	LCD Image 1 Size Register
LI1BA0[1]	0xBC	R/W	0x00000000	LCD Image 1 Base Address 0 Register.
LI1CA[1]	0xC0	R/W	0x00000000	LCD Image 1 Current Address Register.
LI1BA1[1]	0xC4	R/W	0x00000000	Not Used
LI1BA2[1]	0xC8	R/W	0x00000000	Not Used
LI1O[1]	0xCC	R/W	0x00000000	LCD Image 1 Offset Register
LI1SR[1]	0xD0	R/W	0x00000000	LCD Image 1 Scale ratio-
LI1A[1]	0xD4	R/W	0x00000000	LCD Image 1 Alpha Configuration Register
LI1KR[1]	0xD8	R/W	0x00000000	LCD Image 1 Keying Register for RED or LUMA(Y)
LI1KG[1]	0xDC	R/W	0x00000000	LCD Image 1 Keying Register for BLUE or CHROMA(Cb)
LI1KB[1]	0xE0	R/W	0x00000000	LCD Image 1 Keying Register for GREEN or CHROMA(Cr)
LI1EN[1]	0xE4	R/W	0x00000000	LCD Image 1 Enhancement Register
LI2C[1]	0xE8	R/W	0x00000000	LCD Image 2 Control Register
LI2P[1]	0xEC	R/W	0x00000000	LCD Image 2 Position Register
LI2S[1]	0xF0	R/W	0x00000000	LCD Image 2 Size Register
LI2BA0[1]	0xF4	R/W	0x00000000	LCD Image 2 Base Address 0 Register.
LI2CA[1]	0xF8	R/W	0x00000000	LCD Image 2 Current Address Register.
LI2BA1[1]	0xFC	R/W	0x00000000	Not Used
LI2BA2[1]	0x100	R/W	0x00000000	Not Used
LI2O[1]	0x104	R/W	0x00000000	LCD Image 2 Offset Register
LI2SR[1]	0x108	R/W	0x00000000	LCD Image 2 Scale ratio
LI2A[1]	0x10C	R/W	0x00000000	LCD Image 2 Alpha Register
LI2KR[1]	0x110	R/W	0x00000000	LCD Image 2 Keying Register for RED or LUMA(Y)
LI2KG[1]	0x114	R/W	0x00000000	LCD Image 2 Keying Register for BLUE or CHROMA(Cb)
LI2KB[1]	0x118	R/W	0x00000000	LCD Image 2 Keying Register for GREEN or CHROMA(Cr)
LI2EN[1]	0x11C	R/W	0x00000000	LCD Image 2 Enhancement Register
LUTIDX	0x120	R/W	0x00000070	Lookup Table index Register
LCDLUT	0x400 – 0x7FC	W	-	LCD Lookup Table

[1] LCDC1 Only Register

LCTRL (LCD Control Registers) **0xF0200000/0xF0204000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EVP	EVS	R2YMD				GEN	656	CKG	BPP<2:0>			PXDW<3:0>			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ID	IV	IH	IP	CLEN	R2Y	DP	NI	TV	OPT	STN	EVSEL	OVP			LEN

EVP [31]	External VSYNC Polarity
0	Direct Input
1	Inverted Input

EVS [30]	External VSYNC Enable
0	Disabled
1	Enabled

R2YMD [29:28]	RGB to YCbCr Conversion Option
0	**** Y = 0.299*R + 0.587G + 0.114B*** **** Cb = -0.172*R – 0.339*G + 0.511*B + 128*** **** Cr = 0.511*R – 0.428*G – 0.083*B + 128*** The range for "***RGB***" is 16 ~ 235, "***Studio Color***". The result is "***Studio Color***" – Normally SDTV.
1	**** Y = 0.257*R + 0.504*G + 0.098*B + 16*** **** Cb = -0.148*R – 0.291*G + 0.439*B + 128*** **** Cr = 0.439*R – 0.368*G – 0.071*B + 128*** The range for "***RGB***" is 0 ~ 255, "***Computer System Color***" The result is "***Studio Color***" – Normally SDTV.
2	**** Y = 0.213*R + 0.715*G + 0.072*B*** **** Cb = -0.117*R – 0.394*G + 0.511*B + 128*** **** Cr = 0.511*R – 0.464*G – 0.047*B + 128*** The range for "***RGB***" is 16 ~ 235, "***Studio Color***".. The result is "***Studio Color***" – Normally HDTV.
3	**** Y = 0.183*R + 0.614*G + 0.062*B + 16*** **** Cb = -0.101*R – 0.338*G + 0.439*B + 128*** **** Cr = 0.439*R – 0.399*G – 0.040*B + 128*** The range for "***RGB***" is 0 ~ 255, "***Computer System Color***". The result is "***Studio Color***" – Normally HDTV.

Notice !!! :
The coefficients of the above table are approximated for fixed point calculation.

GEN [25]	Gamma Correction Enable Bit
0	Disabled
1	Enabled

656 [24]	CCIR 656 Mode
0	Disabled
1	Enabled

CKG [23]	Clock Gating Enable for Timing Generator
0	Pixel Clock cannot be gated. If output buffer is empty, the display image can be trembled.
1	Pixel clock can be gated when output buffer is empty. If output buffer is empty, the pixel clock holds on the previous status.

BPP [22:20]	Bit Per Pixel for STN-LCD
0	1bbp
1	2bpp
2	4bpp
3	RGB332
4	RGB444
5-7	Reserved

BPP is only for STN LCD.

PXDW [19:16]	#Bits	Description
0	4 bits	Only for STN LCD
1	8 bits	Only for STN LCD
2	8 bits	RGB Stripe Type * Odd/Even Line : R → G → B → R → G → B …
3	16bits	RGB565 Format
4	16bits	RGB555 Format
5	18bits	RGB666 Format
6	8 bits	YCbCr Format * Cb → Y → Cr → Y
7	8 bits	YCbCr Format * Cr → Y → Cb → Y
8	16 bits	YCbCr Format * {Y,Cb} → {Y,Cr}
9	16 bits	YCbCr Format * {Y,Cr} → {Y,Cb}
10	8 bits	RGB Delta Type * Odd(1st) Line : R → G → B → R → G → B … * Even(2nd) Line : G → B → R → G → B → R …
11	8 bits	RGB Delta Type * Odd(1st) Line : G → B → R → G → B → R … * Even(2nd) Line : R → G → B → R →G → B …
12	24 bits	RGB888 Format
13	8 bits	RGB Dummy Format * Odd/Even Line : R→G→B→Dummy→ …
14	16bits	RGB666 Format * R[17:12], G[11:6], B[5:0] * [15:0] : First Data * [17:16] : Second Data
15	16bits	RGB888 Format * R[23:16], G[15:8], B[7:0] * [15:0] : First Data * [23:16] : Second Data

Refer to Figure 4.18 on page 4-29 for more information about PXDW.

ID [15]	Inverted Data Enable (ACBIAS pin)
0	Active high
1	Active low

IV [14]	Inverted Vertical Sync
0	active high
1	active low

IH [13]	Inverted Horizontal Sync
0	active high
1	active low

IP [12]	Inverted Pixel Clock
0	Pixel data is driven on the rising edge of pixel clock.
1	Pixel data is driven on the falling edge of pixel clock.

CLEN[11]	Clipping Enable
0	Disable
1	Enable

R2Y[10]	RGB to YCbCr Converter Enable for OUTPUT
0	Disable
1	Output pixel data is converted to YCbCr format.

DP[9]	Double Pixel Data
0	One pixel data per 1 PXCLK cycle is output
1	One pixel data per 2 PXCLK cycle is output.

NI[8]	Non-interlace
0	Interlace mode Odd field timing control : LVTME1, LVTME2 Even field timing control : LVTIME3, LVTIME4
1	Non-interlace mode (progressive mode)

TV[7]	TV mode
0	Normal mode
1	TV Mode Values of LVTIMEn registers are based on HSYNC cycle/2.

OPT[6]	LCD DMA Operation mode
0	8 burst mode
1	16 burst mode

STN[5]	STN LCD mode
0	Normal mode
1	STN LCD mode

TV[7]	STN[5]	LCD Operation mode
0	0	TFT-LCD mode
0	1	STN-LCD mode
1	0	TV mode

EVSEL [4]	External VSYNC Select
0	External VSYNC In case of LCD Controller 0 : EVS0 In case of LCD Controller 1 : EVS1
1	Internal Generated VSYNC In case of LCD Controller 0 : VSYNC from LCD Controller 1 In case of LCD Controller 1 : VSYNC from LCD Controller 0

OVP[3:1]	Layer 2 (Top Layer)	Layer 1	Layer 0 (Bottom Layer)
0	Image 0	Image 1	Image 2
1	Image 0	Image 2	Image 1
2	Image 1	Image 0	Image 2
3	Image 2	Image 0	Image 1
4	Image 1	Image 2	Image 0
5	Image 2	Image 1	Image 0

The OVP field determines the relations between image channels and layers.

LEN[0]	LCD Controller Enable
0	Disable
1	Enable

If LEN was written ‘1’ followed by being written by ‘0’, only 1 frame would be displayed and the display controller would be stopped.

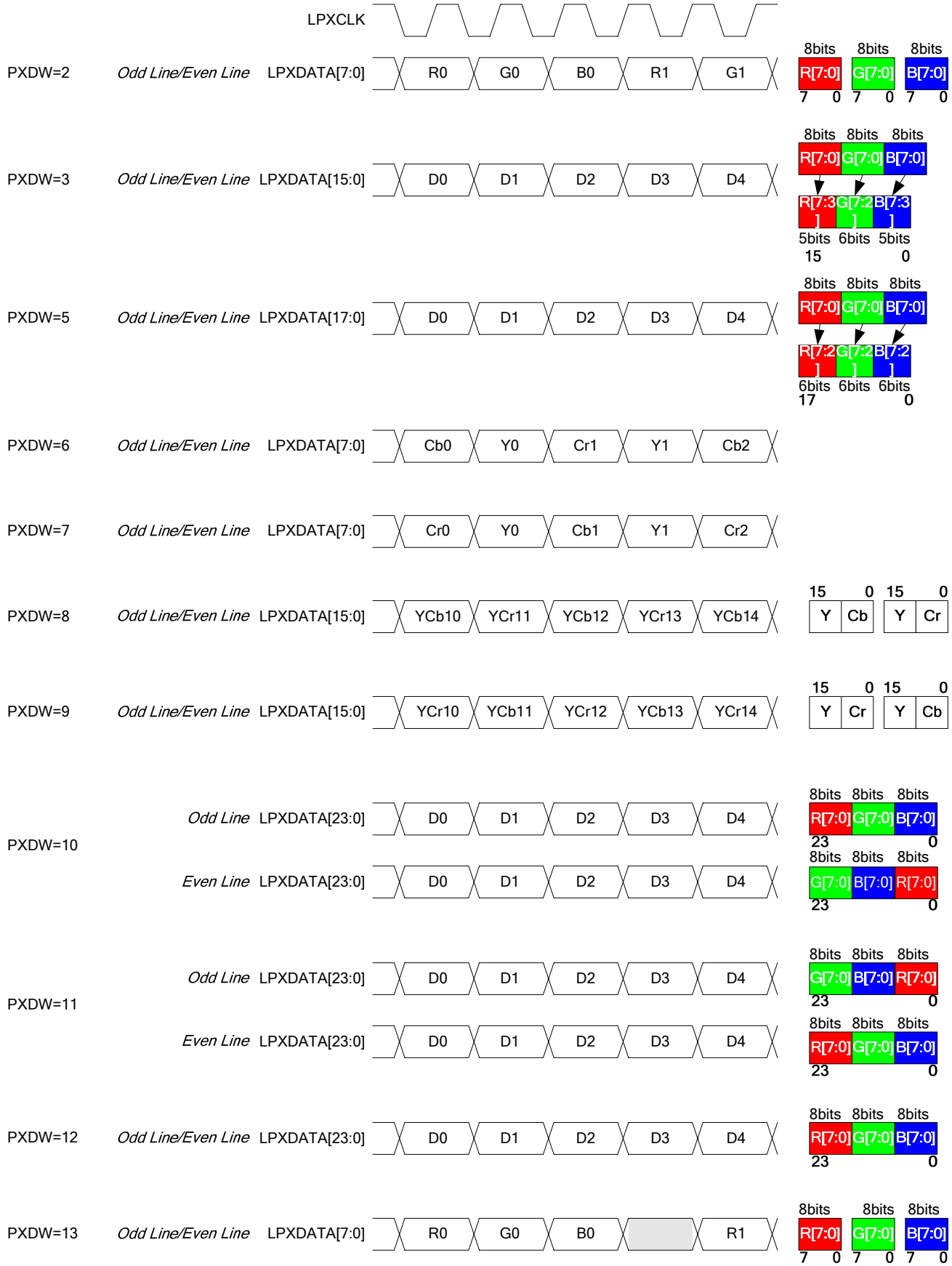


Figure 4.18 Output Pixel Data Format

LBC (LCD Background Layer Color) 0xF0200004/0xF0204004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BGV								BG2<7:0>							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BG1<7:0>								BG0<7:0>							

BGV [24]	Background Layer Data Valid Bit for Alpha-blending
0	The background data(BG0,BG1,BG2) are discarded for alpha-blending
1	The alpha-blending for background data(BG0,BG1,BG2) was applied.

FIELD	Description
BG2 [23:16]	Background color 2 (Y/B)
BG1 [15:8]	Background color 1 (Cb/G)
BG0 [7:0]	Background color 0 (Cr/R)

LCLKDIV (LCD Clock Divider Register) 0xF0200008/0xF0204008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								ACDIV<7:0>							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								PXCLKDIV<7:0>							

FIELD	Description
ACDIV [23:16]	AC bias clock divisor (STN only) The number of line clock cycle to count between each toggle of AC_BIAS pin
PXCLKDIV [7:0]	Pixel clock divider. Note that programming CLKDIV less than 3 is illegal for STN LCD. PXCLK = LCLK / (2*PXCLKDIV) (PXCLKDIV=0~255)

LHTIME1 (LCD Horizontal Timing Register 1) 0xF020000C/0xF020400C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							LPW<8:0>								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		LPC<13:0>													

FIELD	Description
LPW [24:16]	Line pulse width
LPC [13:0]	Line pulse count is the number of pixel clock cycles in each line minus 1 on the screen. TFT/NTSC(16bit)/PAL(16bit) : active horizontal pixel – 1 Color STN : (3 * Horizontal display size / pixel width) Mono STN : (Horizontal display size / pixel width) - 1

LHTIME2 (LCD Horizontal Timing Register 2) 0xF0200010/0xF0204010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							LSWC<8:0>								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0							LEWC<8:0>								

FIELD	Description
LSWC [24:16]	Line start wait clock is the number of dummy pixel clock cycles minus 1 to be inserted from the start of each horizontal line of pixels.
LEWC [8:0]	Line end wait clock is the number of dummy pixel clock cycles minus 1 to be inserted before the end of each horizontal line of pixels

VTIME1 (LCD Vertical Timing Register 1) **0xF0200014/0xF0204014**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VDB<4:0>					0	VDF<3:0>				FPW<5:0>					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0		FLC<13:0>													

FIELD	Description
VDB[31:27]	Back porch VSYNC delay Delay cycle is -10 to 10cycle delay by PXLCLK. When TV mode, VDB value is equal to VDF. Ex) if VD=5, VSYNC delay is 5 cycle delay by HSYNC.
VDF[26:23]	Front porch of VSYNC delay Delay cycle is 0 to 10 cycle delay by PXLCLK. Ex) if VD=5, VSYNC delay is 5 cycle delay by HSYNC.
FPW [21:16]	TFT/TV : Frame pulse width is the pulse width of frame clock (VSYNC). STN : N/A
FLC [13:0]	Frame line count is the number of lines in each frame on the screen.

Refer to Figure 4.12, Figure 4.13, and Figure 4.14.

LVTIME2 (LCD Vertical Timing Register 2) **0xF0200018/0xF0204018**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							FSWC<8:0>								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0							FEWC<8:0>								

FIELD	Description
FSWC [24:16]	TFT/TV: Frame start wait cycle is the number of lines to be inserted at the end of each frame. STN : FSWC is N/A. If FSWC[0] is set, VSYNC signal starts on negative falling edge of HSYNC.
FEWC [8:0]	TFT/TV: Frame end wait cycle is the number of lines to be inserted at the beginning of each frame. STN extra dummy lines between the end and beginning of frame.

LVTIME3 (LCD Vertical Timing Register 3) **0xF020001C/0xF020401C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0										FPW<5:0>					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0		FLC<13:0>													

If NI of LCTRL is 0, LVTIME3 and LVTIME4 are for even field. Otherwise, LVTIME3 and LVTIME4 must be equal to LVTIME1 and LVTIME2, respectively.

LVTIME4 (LCD Vertical Timing Register 4) **0xF0200020/0xF0204020**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							FSWC<8:0>								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0							FEWC<8:0>								

If NI of LCTRL is 0, LVTIME3 and LVTIME4 are for even field. Otherwise, LVTIME3 and LVTIME4 must be equal to LVTIME1 and LVTIME2, respectively.

LLUTR (LCD Lookup Register for RED) 0xF0200024/0xF0204024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LLUTR<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LLUTR<15:0>															

This register is used for supporting palletized color STN-LCD. It is divided into 8 nibbles. The passive color mode uses a lookup table register, which allows any 8 red levels to be selected out of the 16 possible red levels. The most significant 3-bit of 8-bit encoded pixel addresses 8 red palette locations. Note that LLUTR register is only used in STN-LCD mode.

LLUTG (LCD Lookup Register for GREEN) 0xF0200028/0xF0204028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LLUTG<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LLUTG<15:0>r															

This register is used for supporting palletized color STN-LCD. It is divided into 8 nibbles. The passive color mode uses a lookup table register, which allows any 8 green levels to be selected out of the 16 possible green levels. The most significant 3-bit of 8-bit encoded pixel addresses 8 green palette locations. Note that LLUTG register is only used in STN-LCD mode.

LLUTB (LCD Lookup Register for BLUE) 0xF020002C/0xF020402C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LLUTB<15:0>															

This register is used for supporting palletized color STN-LCD. It is divided into 4 nibbles. The passive color mode uses a lookup table register, which allows any 4 blue levels to be selected out of the 16 possible blue levels. The most significant 2-bit of 8-bit encoded pixel addresses 4 blue palette locations. Note that LLUTB register is only used in STN-LCD mode.

LDP7L (LCD Dithering Pattern Register) **0xF0200030/0xF0204030**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	DP5_7<6:0>							0	DP4_7<6:0>						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	DP3_7<6:0>							0	DP1_7<6:0>						

LDP7H (LCD Dithering Pattern Register) **0xF0200034/0xF0204034**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0									DP6_7<6:0>						

LDP5 (LCD Dithering Pattern Register) **0xF0200038/0xF0204038**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			DP4_5<5:0>					0			DP3_5<5:0>				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0			DP2_5<5:0>					0			DP1_5<5:0>				

LDP4 (LCD Dithering Pattern Register) **0xF020003C/0xF020403C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0				DP3_4<4:0>				DP2_4<4:0>				DP1_4<4:0>			

LDP3 (LCD Dithering Pattern Register) **0xF0200040/0xF0204040**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0									DP2_3<3:0>				DP1_3<3:0>		

Theses dithering pattern registers are only used by STN LCD. It is recommended that reset values are used.

Pre-dithered value	Dithering register	Duty cycle ratio
0	all 0s	0
1	DP1_7	1/7
2	DP1_5	1/5
3	DP1_4	1/4
4	DP1_3	1/3
5	DP2_5	2/5
6	DP3_7	3/7
7	DP2_4	1/2
8	DP4_7	4/7
9	DP3_5	3/5
10	DP2_3	2/3
11	DP5_7	5/7
12	DP3_4	3/4
13	DP4_5	4/5
14	DP6_7	6/7
15	all 1s	1

LCP1 (LCD Clipping Register 1) **0xF0200044/0xF0204044**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CLP1L							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								CLP1							

FIELD	Description
CLP1L[23:16]	Clipping Y/R below this value. (standard value is 0)
CLP1 [7:0]	Clipping Y/R upper this value. (standard value is 255)

LCP2 (LCD Clipping Register 2) **0xF0200048/0xF0204048**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CLP2L							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								CLP2							

FIELD	Description
CLP2L [23:16]	Clipping CHROMA/G/B below this value. (standard value is 0)
CLP2 [7:0]	Clipping CHROMA/G/B upper this value. (standard value is 255)

LCP1 and LCP2 may be used not to interpret pixel data as embedded sync signal in the CCIR-656 interface.

LDS (LCD Display Size Register) 0xF020004C/0xF020404C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			VSIZE<12:0>												
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
			HSIZE<12:0>												

FIELD	Description
VSIZE [28:16]	Horizontal size : number of active pixel in a line
HSIZE [12:0]	Vertical size : number of active lines

Note) VSIZE = 32~4096, HSIZE = 32~4096

LSTATUS (LCD Status Register) 0xF0200050/0xF0204050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VS	BUSY	EF	DEOF	I0EOF	I1EOF	I2EOF									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STATUS[15:0]															

FIELD	Description
VS[31]	Monitoring vertical sync.- Read-Only register It has the same logic level with LVSYNC port.
BUSY[30]	Busy signal During operation, is can be read by 1. If LEN is disabled, it will be read by 0 after current frame has been displayed.
EF [29]	It indicates that the type of current frame(field) - even-field (read only) 0 : Odd field or frame 1 : Even field or frame
DEOF[28]	DMA End-of-Frame flag
I0EOF[27]	Image 0 End-of-Frame flag
I1EOF[26]	Image 1 End-of-Frame flag
I2EOF[25]	Image 2 End-of-Frame flag
STATUS[15:0]	Status registers The following flag should be cleared by writing '1' to the corresponding bit.

FIELD	Description
FU[0]	LCD output fifo under-run flag.
VSR[1]	VS rising flag
VSF[2]	VS falling flag
RU[3]	Register update flag It indicates that all registers programmed are applied to current frame data.
DD[4]	Disable done If LEN is disabled, it will be set after current frame has been displayed.
DER[5]	DMA end-of-frame rising edge flag
DEF[6]	DMA end-of-frame falling edge flag
IE0R[7]	Image 0 end-of-frame rising edge flag
IE0F[8]	Image 0 end-of-frame falling edge flag
IE1R[9]	Image 1 end-of-frame rising edge flag
IE1F[10]	Image 1 end-of-frame falling edge flag
IE2R[11]	Image 2 end-of-frame rising edge flag
IE2F[12]	Image 2 end-of-frame falling edge flag

LIM (LCD Interrupt Masking Register) **0xF0200054/0xF0204054**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK[15:0]															

MASK[15:0]	Description
FU[0]	LCD output fifo under-run interrupt mask
VSR[1]	VS rising interrupt mask
VSF[2]	VS falling interrupt mask
RU[3]	Register update interrupt mask
DD[4]	Disable done interrupt mask
DER[5]	DMA end-of-frame rising edge interrupt mask
DEF[6]	DMA end-of-frame falling edge interrupt mask
IE0R[7]	Image 0 end-of-frame rising edge interrupt mask
IE0F[8]	Image 0 end-of-frame falling edge interrupt mask
IE1R[9]	Image 1 end-of-frame rising edge interrupt mask
IE1F[10]	Image 1 end-of-frame falling edge interrupt mask
IE2R[11]	Image 2 end-of-frame rising edge interrupt mask
IE2F[12]	Image 2 end-of-frame falling edge interrupt mask

LGR0 (LCD Gamma Correction Register 0 for Red Color) **0xF0200058/0xF0204058**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GR7				GR6				GR5				GR4			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GR3				GR2				GR1				GR0			

LGR1 (LCD Gamma Correction Register 1 for Red Color) **0xF020005C/0xF020405C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GR15				GR14				GR13				GR12			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GR11				GR10				GR9				GR8			

LGG0 (LCD Gamma Correction Register 0 for Green Color) **0xF0200060/0xF0204060**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GG7				GG6				GG5				GG4			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GG3				GG2				GG1				GG0			

LGG1 (LCD Gamma Correction Register 1 for Green Color) **0xF0200064/0xF0204064**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GG15				GG14				GG13				GG12			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GG11				GG10				GG9				GG8			

LGB0 (LCD Gamma Correction Register 0 for Blue Color) **0xF0200068/0xF0204068**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GB7				GB6				GB5				GB4			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GB3				GB2				GB1				GB0			

LGB1 (LCD Gamma Correction Register 1 for Blue Color) **0xF020006C/0xF020406C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GB15				GB14				GB13				GB12			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GB11				GB10				GB9				GB8			

LENH (LCD Color Enhancement Register) **0xF0200070/0xF0204070**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								HUE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BRIGHTNESS								CONTRAST							

FIELD	Description
HUE [23:16]	Hue Calibration Register – 2's complement signed value * -30 ~ 30 degree * 0x80 for -30 degree * 0x00 for 0 degree for default value * 0x7F for about 30 degree
BRIGHTNESS[15:8]	Brightness Calibration Register – 2's complement signed value * -128 ~ 128 value * 0x80 for -128 offset * 0x00 for 0 offset * 0x7F for 127 offset
CONTRAST [7:0]	Contrast Calibration Register – 2's complement signed value * -4 ~ 4 * 0x80 for -4.0 value * 0xFF for -1.0 value * 0x20 for 1.0 value * 0x7F for about 4.0 value

LI0C,LI1C,LI2C (Control Registers for Each Image) 0xF0200078/0xF0204078

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTL	AEN	CEN	IEN	SRC	AOPT		ASEL								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PD					Y2RMD		Y2R	BR			FMT				

INTL [31]	DMA Interlace Mode
0	Progressive Data Fetch
1	Interlace Data Fetch * Line Order is 1,2,3,4,5,6 … * Line Order of odd frame : 1,3,5,… * Line Order of even frame : 2,4,6,…

AEN [30]	Alpha-blending Function for Each Image
0	Disabled
1	Enabled

CEN [29]	Chroma-Keying Function for Each Image
0	Disabled
1	Enabled

IEN [28]	Image Displaying Function for Each Image
0	Disabled
1	Enabled

SRC [27]	Image Source Select
0	Image data from the memory
1	Image data from the on-the-fly path

AOPT [26:25]	Alpha-blending Option Selection Bits
0	The alpha value is not changed. ***(100% ~ 0.39% transparency)***
1	The alpha value is added by ‘1’ ***(99.61% ~ 0% transparency)***
2	The alpha value is not changed when msb of alpha is ‘0’ and added by ‘1’ when most significant bit of alpha value is ‘1’ ***(100% ~ 0% transparency)***

ASEL [24]	Image Displaying Function for Each Image
0	A10 bits and A11 are used as alpha value
1	Alpha bits in the pixel are used as alpha value.

PD [15]	Bit padding
0	The extra lower bits are padded by ‘0’.
1	The extra lower bits are padded by most significant bit.

Though raw image source is not RGB888 format, it is internally converted to RGB888 as appending additional bits. When this conversion is occurred, these padding bits are determined by PD bit.

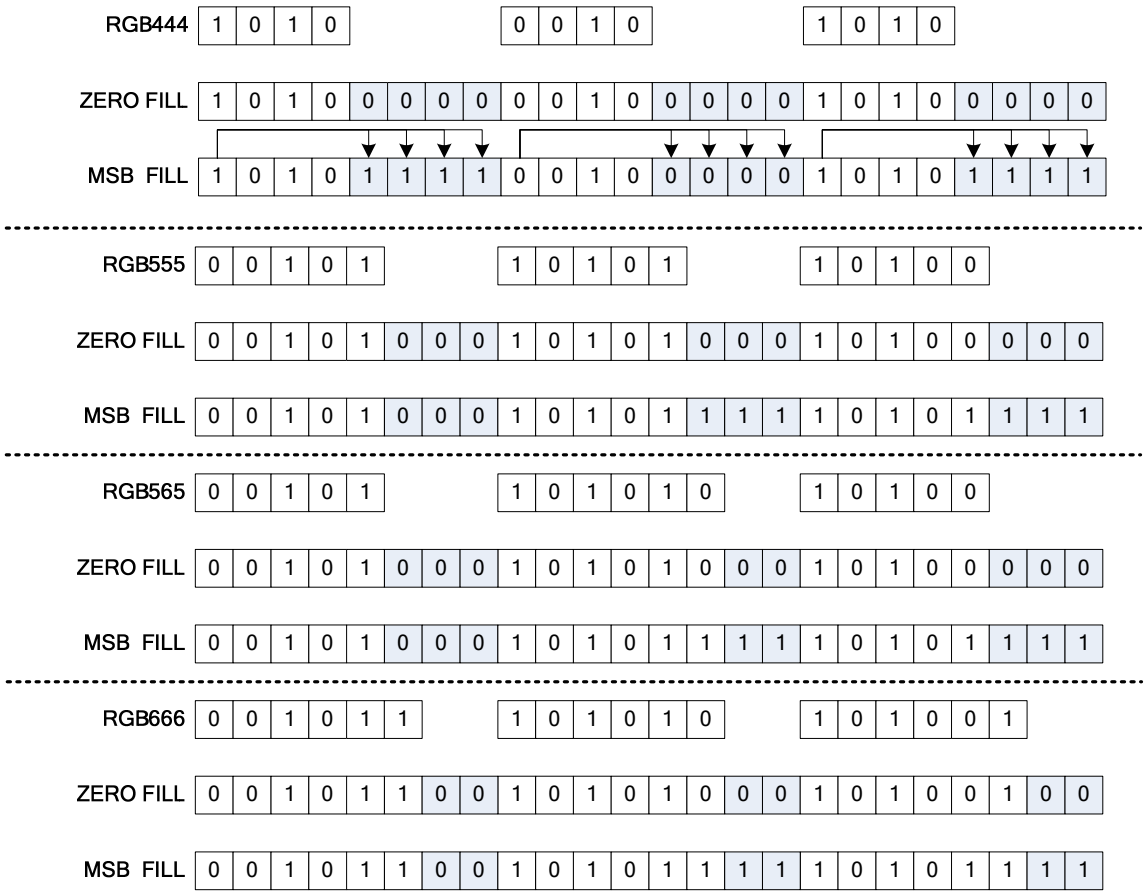


Figure 4.19 Bit Padding

Y2R [8]	YCbCr to RGB Conversion Enable Bit
0	Disable
1	Enable

Y2RMD [10:9]	YCbCr to RGB Conversion Option
0	**** R = Y + 1.371 * (Cr – 128)*** **** G = Y + 0.336 * (Cb – 128) – 0.698 * (Cr – 128)*** **** B = Y + 1.732 * (Cb – 128)*** The range for "***YCbCr***" is 16 ~ 235, "***Studio Color***". The result is "***Studio Color***" – Normally SDTV.
1	**** R = 1.164 * (Y – 16) + 1.596 * (Cr – 128)*** **** G = 1.164 * (Y – 16) – 0.391 * (Cb – 128) – 0.813 * (Cr – 128)*** **** B = 1.164 * (Y – 16) + 2.018 * (Cb – 128)*** The range for "***YCbCr***" is 16 ~ 235, "***Studio Color***". The result is "***Computer System Color***" – Normally SDTV.
2	**** R = Y + 1.540 * (Cr – 128)*** **** G = Y – 0.183 * (Cb – 128) – 0.459 * (Cr – 128)*** **** B = Y + 1.816 * (Cb – 128)*** The range for "***YCbCr***" is 16 ~ 235, "***Studio Color***". The result is "***Studio Color***" – Normally HDTV.
3	**** R = 1.164 * (Y – 16) + 1.793 * (Cr – 128)*** **** G = 1.164 * (Y – 16) – 0.213 * (Cb – 128) – 0.534 * (Cr – 128)*** **** B = 1.164 * (Y – 16) + 2.115 * (Cb – 128)*** The range for "***YCbCr***" is 16 ~ 235, "***Studio Color***". The result is "***Computer System Color***" – Normally HDTV.

Notice !!! :
The coefficients of the above table are approximated for fixed point calculation.

BR[7]	Bit Reverse
0	Little-endian pixel data
1	Big-endian pixel data

BR[7] is only used when BPP is 1, 2, or 4 bpp

FMT [4:0]	Image Format
0	1bpp indexed color
1	2bpp indexed color
2	4bpp indexed color
3	8bpp indexed color
4 ~ 7	Reserved for future use
8	RGB332 – 1 bytes aligned : R[7:5],G[4:2],B[1:0]
9	RGB444 – 2 bytes aligned : A[15:12],R[11:8],G[7:4],B[3:0]
10	RGB565 – 2 bytes aligned : R[15:11],G[10:5],B[4:0]
11	RGB555 – 2 bytes aligned : A[15],R[14:10],G[9:5],B[4:0]
12	RGB888 – 4 bytes aligned : A[31:24],R[23:16],G[15:8],B[7:0]
13	RGB666 – 4 bytes aligned : A[23:18],R[17:12],G[11:6],B[5:0]
14 ~ 23	Reserved for future use
24	YCbCr 4:2:0 separated format - ***Not Supported for Image 1 and 2***
25	YCbCr 4:2:2 separated format - ***Not Supported for Image 1 and 2***
26	YCbCr 4:2:2 sequential format
27	Reserved for future use
28	YCbCr 4:2:0 interleaved type 0 format - ***Not Supported for Image 1 and 2***
29	YCbCr 4:2:0 interleaved type 1 format - ***Not Supported for Image 1 and 2***
30	YCbCr 4:2:2 interleaved type 0 format - ***Not Supported for Image 1 and 2***
31	YCbCr 4:2:2 interleaved type 1 format - ***Not Supported for Image 1 and 2***

The "A", "R", "G" and "B" stand for alpha value, red, green and blue color correspondingly. The alpha value to be applied is determined by ASEL bit. If ASEL is zero, the alpha value equals to "A" with extra bit extended.

LI0P,LI1P,LI2P (Start Position Registers for Each Image) 0xF020007C/0xF020407C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			POSY<12:0>												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			POSX<12:0>												

FIELD	Description
POSY [28:16]	Y position to display.
POSX [12:0]	X position to display.

LI0S,LI1S,LI2S (Size Information Registers for Each Image Channel) 0xF0200080/0xF0204080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			HEIGHT<12:0>												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			WIDTH<12:0>												

FIELD	Description
HEIGHT [28:16]	Height of each Image
WIDTH [12:0]	Width of each image

FMT	Image Format	Pixel Width Constraint
0	1bpp indexed color	64 pixels
1	2bpp indexed color	16 pixels
2	4bpp indexed color	8 pixels
3	8bpp indexed color	4 pixels
4 ~ 7	Reserved for future use	
8	RGB332	4 pixels
9	RGB444	2 pixels
10	RGB565	2 pixels
11	RGB555	2 pixels
12	RGB888	1 pixels
13	RGB666	1 pixels
14 ~ 23	Reserved for future use	
24	YCbCr 4:2:0 separated format	8 pixels
25	YCbCr 4:2:2 separated format	8 pixels
26	YCbCr 4:2:2 sequential format	2 pixels
27	Reserved for future use	
28	YCbCr 4:2:0 interleaved type 0 format	4 pixels
29	YCbCr 4:2:0 interleaved type 1 format	4 pixels
30	YCbCr 4:2:2 interleaved type 0 format	4 pixels
31	YCbCr 4:2:2 interleaved type 1 format	4 pixels

Important Notice :
The value of “WIDTH” should be integer multiple of the number of pixels described in the “Pixel Width Constraint” field in the above table. The limitation can be changed by “X_SCALE” in the following LI0SC, LI1SC, LI2SC register. The downscale ratio determined by “X_SCALE” should be multiplied by the upper constraint. For example, If the downscale ratio is 3 and FMT is 2, the pixel width limitation is changed from “8 pixels” to “24 pixels”

LI0BA0,LI1BA0,LI2BA0 (Base Address for Each Images) 0xF0200084/0xF0204084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASE0<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE0<15:0>															

FIELD	Description
BASE0 [31:0]	1st base address for each image If the image format is YCbCr separated type, it is the base address of Y.

LI0CA, LI1CA, LI2CA (Current Address for Each Images) 0xF0200088/0xF0204088

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURR<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURR<15:0>															

FIELD	Description
CURR [31:0]	The working address for 1st base address.

LI0BA1,LI1BA1,LI2BA1 (2nd Base Address for Each Images) 0xF020008C/0xF020408C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASE1<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE1<15:0>															

The LI1BA1 and LI2BA1 are not supported.

FIELD	Description
BASE1 [31:0]	The 2nd base address for each image. If the image format is YCbCr separated type, it is the base address for Cb, otherwise it is not used.

LI0BA2,LI1BA2,LI2BA2 (3rd Base Address for Each Images) 0xF0200090/0xF0204090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASE2<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE2<15:0>															

The LI1BA2 and LI2BA2 are not supported.

FIELD	Description
BASE2 [31:0]	The 3rd base address for each image. If the image format is YCbCr separated type, it is the base address for Cr, otherwise it is not used.

FMT	Image Format	Base Address Constraints
0	1bpp indexed color	8 pixels aligned (1byte aligned)
1	2bpp indexed color	4 pixels aligned (1byte aligned)
2	4bpp indexed color	2 pixels aligned (1byte aligned)
3	8bpp indexed color	1 pixels aligned (1byte aligned)
4 ~ 7	Reserved for future use	
8	RGB332	1 pixels aligned (1byte aligned)
9	RGB444	1 pixels aligned (2bytes aligned)
10	RGB565	1 pixels aligned (2bytes aligned)
11	RGB555	1 pixels aligned (2bytes aligned)
12	RGB888	1 pixels aligned (4bytes aligned)
13	RGB666	1 pixels aligned (4bytes aligned)
14 ~ 23	Reserved for future use	
24	YCbCr 4:2:0 separated format	2 pixels aligned (2bytes aligned)
25	YCbCr 4:2:2 separated format	2 pixels aligned (2bytes aligned)
26	YCbCr 4:2:2 sequential format	1 pixels aligned (2bytes aligned)
27	Reserved for future use	
28	YCbCr 4:2:0 interleaved type 0 format	2 pixels aligned (2bytes aligned)
29	YCbCr 4:2:0 interleaved type 1 format	2 pixels aligned (2bytes aligned)
30	YCbCr 4:2:2 interleaved type 0 format	2 pixels aligned (2bytes aligned)
31	YCbCr 4:2:2 interleaved type 1 format	2 pixels aligned (2bytes aligned)

LI0O,LI1O,LI2O (Offset Information Registers for Each Image) 0xF0200094/0xF0204094

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFFSET1<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET0<15:0>															

FIELD	Description
OFFSET1[31:16]	The 2nd offset information for each image. It is the address offset in U or V channel of FIFO (FIFO1,2) ***Invalid for Image 1 and Image 2***
OFFSET0 [15:0]	The 1st offset information for each image. It is the address offset in Y or RGB channel of FIFO (FIFO0).

LI0SC,LI1SC,LI2SC (Scaling Information Registers for Each Image) 0xF0200098/0xF0204098

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Y_SCALE				X_SCALE			

SCALE[3]	[2]	[1]	[0]	Description	Note
0	0	0	0	Non-Scalable	
1	0	0	1	Upscale by 2	FMT 28, 29, 30 and 31 does NOT support scaling
1	0	1	0	Upscale by 3	
1	0	1	1	Upscale by 4	
1	1	1	1	Upscale by 8	

LI0A,LI1A,LI2A (Alpha Information Registers for Each Image) 0xF020009C/0xF020409C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								A1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A0							

BIT	NAME	DESCRIPTION
23:16	A1	These bits should be the same as value of “A0”.
7:0	A0	These are used as alpha value for each image when ASEL bit is ‘0’. ***Output Pixel = Main Pixel *(1- A0/256) + Overlay Pixel* (A0)***

LI0KR,LI1KR,LI2KR (Keying Registers for RED or LUMA(Y)) 0xF02000A0/0xF02040A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								KEYMASK							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								KEY							

BIT	NAME	DESCRIPTION
23-16	KEYMASK	Key Mask Value for RED(Y)
7-0	KEY	Key Value for RED(Y)

LI0KG,LI1KG,LI2KG (Keying Registers for GREEN or CHROMA(Cb)) 0xF02000A4/0xF02040A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								KEYMASK							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								KEY							

BIT	NAME	DESCRIPTION
23-16	KEYMASK	Key Mask Value for GREEN(Cb)
7-0	KEY	Key Value for GREEN(Cb)

LI0KB,LI1KB,LI2KB (Keying Registers for BLUE or CHROMA(Cr)) **0xF02000A8/0xF02040A8**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								KEYMASK							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								KEY							

BIT	NAME	DESCRIPTION
23-16	KEYMASK	Key Mask Value for BLUE(Cr)
7-0	KEY	Key Value for BLUE(Cr)

LI0EN,LI1EN,LI2EN (LCD Image Enhancement Register) **0xF02000AC/0xF02040AC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								HUE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BRIGHTNESS								CONTRAST							

FIELD	Description
HUE [23:16]	Hue Calibration Register – 2's complement signed value * -30 ~ 30 degree * 0x80 for -30 degree * 0x00 for 0 degree for default value * 0x7F for about 30 degree
BRIGHTNESS[15:8]	Brightness Calibration Register – 2's complement signed value * -128 ~ 128 value * 0x80 for -128 offset * 0x00 for 0 offset * 0x7F for 127 offset
CONTRAST [7:0]	Contrast Calibration Register – 2's complement signed value * -4 ~ 4 * 0x80 for -4.0 value * 0xFF for -1.0 value * 0x20 for 1.0 value * 0x7F for about 4.0 value

LUTIDX (Lookup Table index Register) **0xF0200120/0xF0204120**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
									EN2	EN1	EN0			CHSEL	

FIELD	Description
EN2[6]	Enable Lookup Table for Image Channel 2
EN1[5]	Enable Lookup Table for Image Channel 1
EN0[4]	Enable Lookup Table for Image Channel 0
CHSEL[1:0]	LUT Access channel select * 0 for LUT0 * 1 for LUT1 * 2 for LUT2

5 LCD System Interface

5.1 Overview

The LCD system interface (LCDSI) is used to send out the image data from the system memory to the LCD module which has 68/80-system interface. The LCDSI can be accessed by both LCDC and the on-chip CPU. The setup time, hold time, and pulse width of the interface signals are programmable.

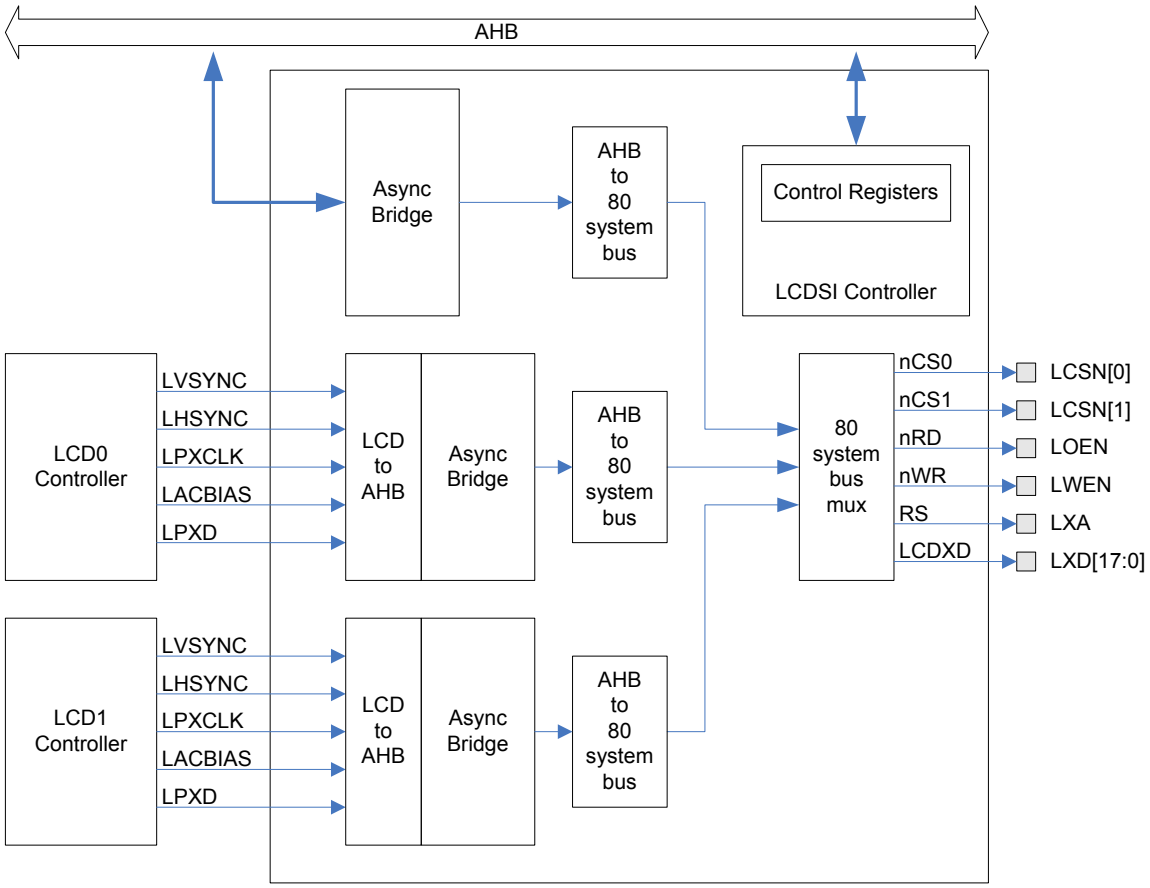


Figure 5.1 LCD System Interface Block Diagram

5.2 Operation

5.2.1 Reading/Writing operation through the on-chip CPU

The LCDSI allows the on-chip CPU to read from and write to an external LCD module, which has 68/80-system interface. If the on-chip CPU accesses LCDSI DATA0 register, then reading or writing operations are generated on the device connected to LCSN[0]. While these operations are executed, LXA is low (Figure 5.2 (a)). If the on-chip CPU accesses LCDSI DATA1 register, LXA is high (Figure 5.2 (b)). Similarly, to access the device connected to LCSN[1], the on-chip CPU must access LCDSI DATA2 or LCDSI DATA3 register. Note that LXA signal is irrelevant to LCDSI LnCTRL.RSP when the on-chip CPU access to LCDSI.

Timing and data width configuration about the LCDSI signals can be programmed via LCDSI CTRLn registers.

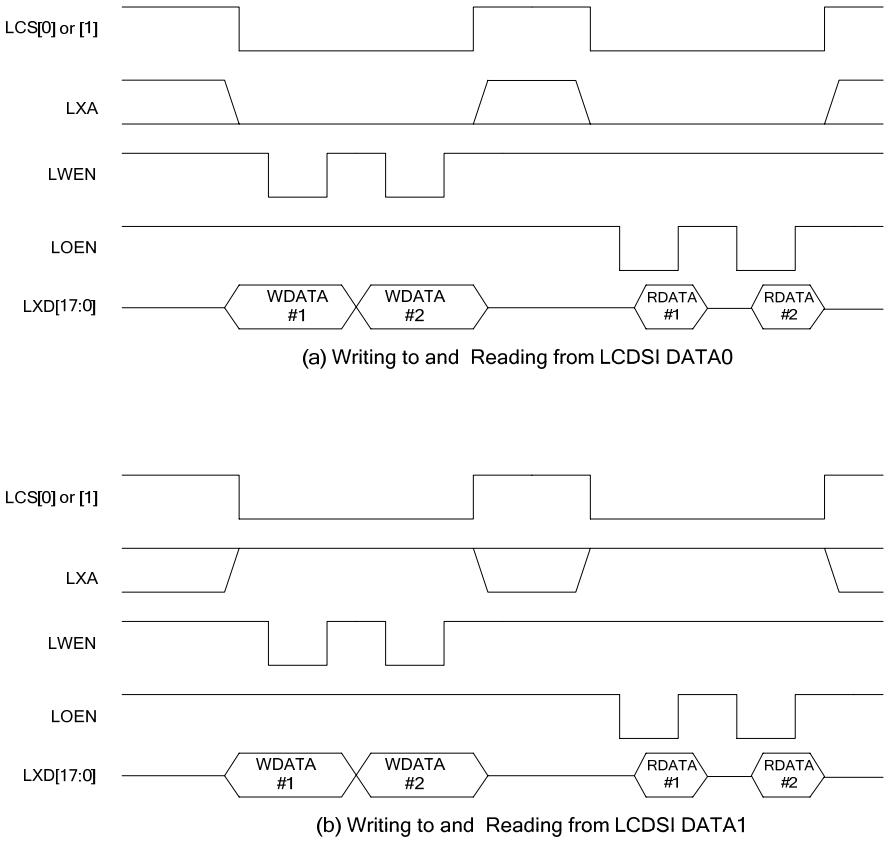


Figure 5.2 Writing / Reading Operation through on-chip CPU

5.2.2 Writing Operation Through LCD Controller

For converting LCDC control signals to 68/80-system interface signals, the LCDC must be configured to TFT mode. And LPXD must be RGB565 or RGB888 and driven at negative edge of LPXCLK. LACBIAS and LVSYNC polarity is also configured by LCDSI LnCTRL.IA and LCDSI LnCTRL.IVS and the values must be same with them of LCDC polarity registers. Refer to LCDC register set for more information.

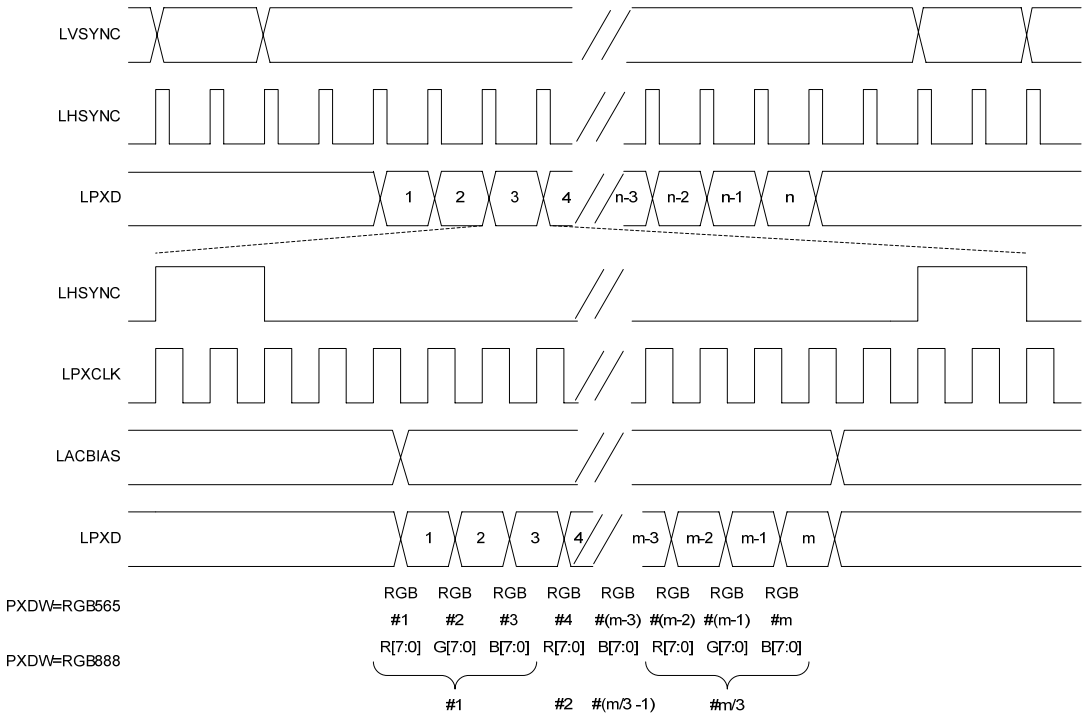


Figure 5.3 Example of LCDC Output Signals for LCDSI

LCDSI LnCTRL.CS, LnCTRL.RSP, LnCTRL.FMT, LnCTRL.IVS, and LnCTRL.IA register fields specify nCS, RS, and output pixel data format during operation and polarity of LVSYNC and LACBIAS signal. For example, if an LCD module is connected to nCS0 and requires RS signal to be low, LCDSI LnCTRL.CS and LCDSI LnCTRL.RSP must be set to 0.

And LCDSI output signals can be adjusted by programming LCDSI CTRL0[31:16] register. LCDXD is dependent on LCDC LPXD, LCDSI LnCTRL.FMT, and LCDSI CTRLn.WBW registers.

The following is the procedure that one frame data of LCDC is sent to LCD module through LCDSI.

(1) Set LCDSI LnCTRL register.
(2) Set the timing parameters for LCD module via LCDSI CTRLn registers.
(3) Set LCDC register for one frame data.
(4) Enable LCDC and Disable it sequentially.

If LCDC is not disabled at 4, LCDC output data are sent to LCD module through LCDSI continuously.

5.3 Register Descriptions

All control registers for LCDSI are listed in Table 5.1.

Table 5.1 LCDSI Register map(0xF020C000)

Name	Address	Type	Reset	Description
LCDSI CTRL0	0x00	R/W	0xA0229011	Control register for LCSN[0] when LXA=0
LCDSI CTRL1	0x04	R/W	0xA0429021	Control register for LCSN[0] when LXA=1
LCDSI CTRL2	0x08	R/W	0xA0129009	Control register for LCSN[1] when LXA=0
LCDSI CTRL3	0x0C	R/W	0xA0229011	Control register for LCSN[1] when LXA=1
LCDSI L0CTRL	0x10	R/W	0x00000020	Control register for Writing operation through LCDC0
LCDSI L1CTRL	0x14	R/W	0x00000020	Control register for Writing operation through LCDC1
LCDSI MODE	0x18	R/W	0x00000000	68/80 mode select
LCDSI STATUS	0x1C	R	-	LCDSI status
-	-	-	-	
LCDSI DATA0	0x40	R/W		If this register is read or written, reading or writing operations are generated on LCSN[0] while LXA = 0.
LCDSI DATA1	0x60	R/W		If this register is read or written, reading or writing operations are generated on LCSN[0] while LXA = 1.
LCDSI DATA2	0x80	R/W		If this register is read or written, reading or writing operations are generated on LCSN[1] while LXA = 0.
LCDSI DATA3	0xA0	R/W		If this register is read or written, reading or writing operations are generated on LCSN[1] while LXA = 1.

LCDSI CTRLn **0xF020C000, 0xF020C004, 0xF020C008, 0xF020C00C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BW[1]	W[1]STP			WPW									WHLD		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BW[0]	R[2]STP			RPW									RHLD		

BW[1:0]	Bus Width
0	LCDSI DATAn registers become 8-bit I/O devices. Therefore, when the on-chip CPU accesses them with a 32-bit data operation, LCDSI generates 4 8-bit data operation.
1	LCDSI DATAn registers become 16-bit I/O devices. Therefore, when the on-chip CPU accesses them with a 32-bit data operation, LCDSI generates 2 16-bit data operation.
2,3	LCDSI DATAn registers become 32-bit I/O devices. Therefore, when the on-chip CPU accesses them with a 32-bit data operation, LCDSI generates 2 32-bit data operation. But, upper 14bits are truncated because LXD is 18bits.

STP	Setup Time
N	N cycles are issued between the falling edge of LCSN and the falling edge of LWEN (Writing operation) or LOEN(Reading operation).

PW	Pulse Width
N	(N+1) cycles are issued between the falling edge of LWEN (or LOEN) and the rising edge of LWEN (or LOEN).

HLD	Hold Time
N	N cycles are issued between the rising edge of LWEN (or LOEN) and the rising edge of LCSN.

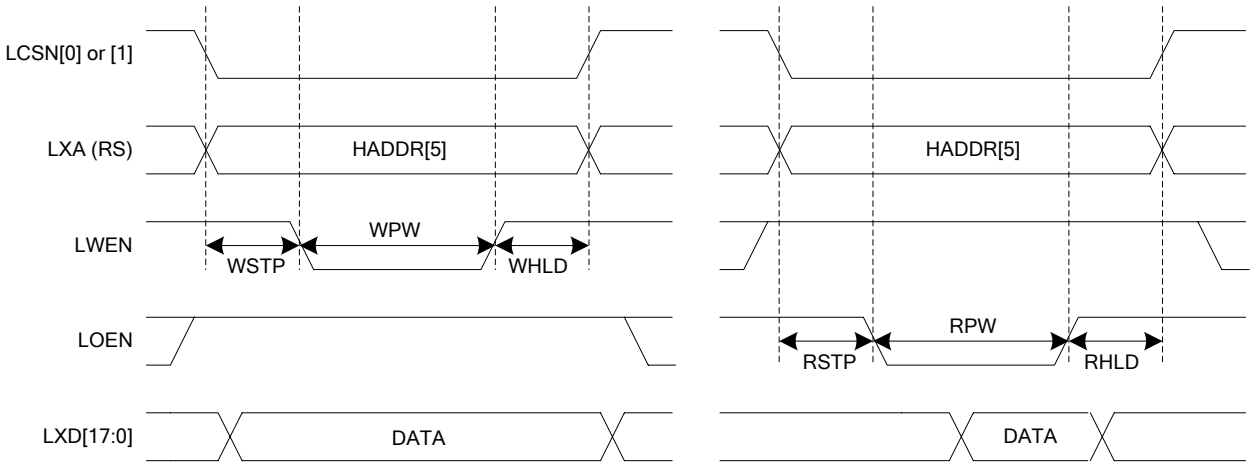


Figure 5.1 Timing Configuration of LCDSI Output Signals for the on-chip CPU Access

[1] Prefix W means writing operation.
[2] Prefix R means reading operation.

LCDSI LnCTRL **0xF020C010, 0xF020C014**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IA	IVS	0					EN	CS	RSP	FMT				0	

IA	Inverse ACBIAS
0	LACBIAS(Data Enable) signal of LCDC is active high.
1	LACBIAS(Data Enable) signal of LCDC is active low.

IVS	Inverse VSYNC
0	LVSYNC signal of LCDC is active high.
1	LVSYNC signal of LCDCL is active low.

EN	Enable
0	Disable
1	Enable

CS	Chip select
0	If IM bit is set to 1, LCSN[0] is used. Otherwise, it is not applicable.
1	If IM bit is set to 1, LCSN[1] is used. Otherwise, it is not applicable.

RSP	RS Polarity
0	If IM bit is set to 1, LXA (RS) is low while LCSN[0] or LSN[1] is asserted. Otherwise, it is not applicable.
1	If IM bit is set to 1, LXA (RS) is low while LCSN[0] or LSN[1] is asserted. Otherwise, it is not applicable.

FMT[3:0]	Pixel Data Format
0	8bit format LCDSI[7:0] <= LPD[7:0]
1~4	Reserved
5	32bit format LCDSI[31:0] <= { 4'b0, LPD[11:0], 4'b0, LPD[23:12] }
6	32bit format LCDSI[31:0] <= { LPD[15:0], 8'b0, LPD[23:16] }
7	32bit format LCDSI[31:0] <= { 8'b0, LPD[7:0], LPD[23:08] }
8	32bit format LCDSI[31:0] <= { 6'b0, LPD[17:0] }
9	32bit format LCDSI[31:0] <= { 14'b0, LPD[1:0], LPD[17:02] }
10	32bit format LCDSI[31:0] <= { 14'b0, LPD[17:0] }
11	32bit format LCDSI[31:0] <= { 7'b0, LPD[17:9], 7'b0, LPD[8:0] }
12	32bit format LCDSI[31:0] <= { 16'b0, LPD[15:0] }
13	16bit format LCDSI[15:0] <= { LPD[7:0], LPD[15:8] }
14	16bit format LCDSI[15:0] <= LPD[15:0]

FMT	LCDSI CTRL0.BW / CTRL1.BW	LCDSI Output (LPXD)
32bit format	32bit (BW = 2 or 3)	1st Data : XD[17:0] = LCDSI[17:0]
	16bit (BW = 1)	1st Data : XD[15:0] = LCDSI[15:0] 2nd Data : XD[15:0] = LCDSI[31:16]
	8bit (BW = 0)	1st Data : XD[7:0] = LCDSI[7:0] 2nd Data : XD[7:0] = LCDSI[15:8] 3rd Data : XD[7:0] = LCDSI[23:16] 4th Data : XD[7:0] = LCDSI[31:24]
16bit format	32bit (BW = 2 or 3)	1st Data : XD[17:0] = { 2'b0, LCDSI[15:0] }
	16bit (BW = 1)	1st Data : XD[15:0] = LCDSI[15:0]
	8bit (BW = 0)	1st Data : XD[7:0] = LCDSI[7:0] 2nd Data : XD[7:0] = LCDSI[15:8]
8bit format	32bit (BW = 2 or 3)	1st Data : XD[17:0] = { 10'b0, LCDSI[7:0] }
	16bit (BW = 1)	1st Data : XD[15:0] = { 8'b0, LCDSI[7:0] }
	8bit (BW = 0)	1st Data : XD[7:0] = LCDSI[7:0]

LCDSI MODE **0xF020C018**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															MODE

MODE	LCDSI MODE
0	80 mode
1	68 mode

LCDSI STATUS **0xF020C01C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														L0OVR	L1OVR

L0OVR	LCD0 Overrun
0	-
1	LCD0 register overrun

L1OVR	LCD1 Overrun
0	-
1	LCD1 register overrun

6 Memory To Memory Scaler

6.1 Overview

The block diagram of the Memory to Memory Scaler (MSC) is shown in the following figure. The hardware reads the source image and resizes it and finally writes the scaled image to the destination region. Specially, the hardware can interface to the LCD controller directly in progressive display output mode.

The maximum resolution of the MSC controller 0 and 1 is limited 4088x4088 and 2032x2032. And the output buffer sizes which can be used as the interface buffer for the on-the-fly mode are 512bytes and 4096bytes correspondingly. In the case of MSC controller 0, it can be a cause of the under-run to display with LCDC in the on-the-fly path because the output buffer is small relative to the display size."

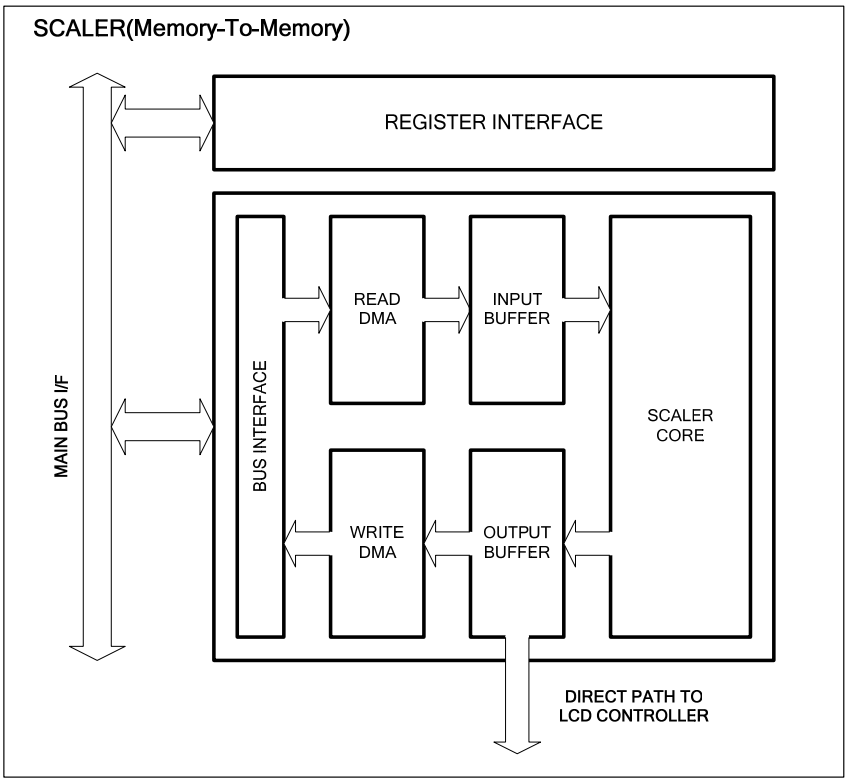


Figure 6.1 Scaler Block Diagram

6.2 Operation

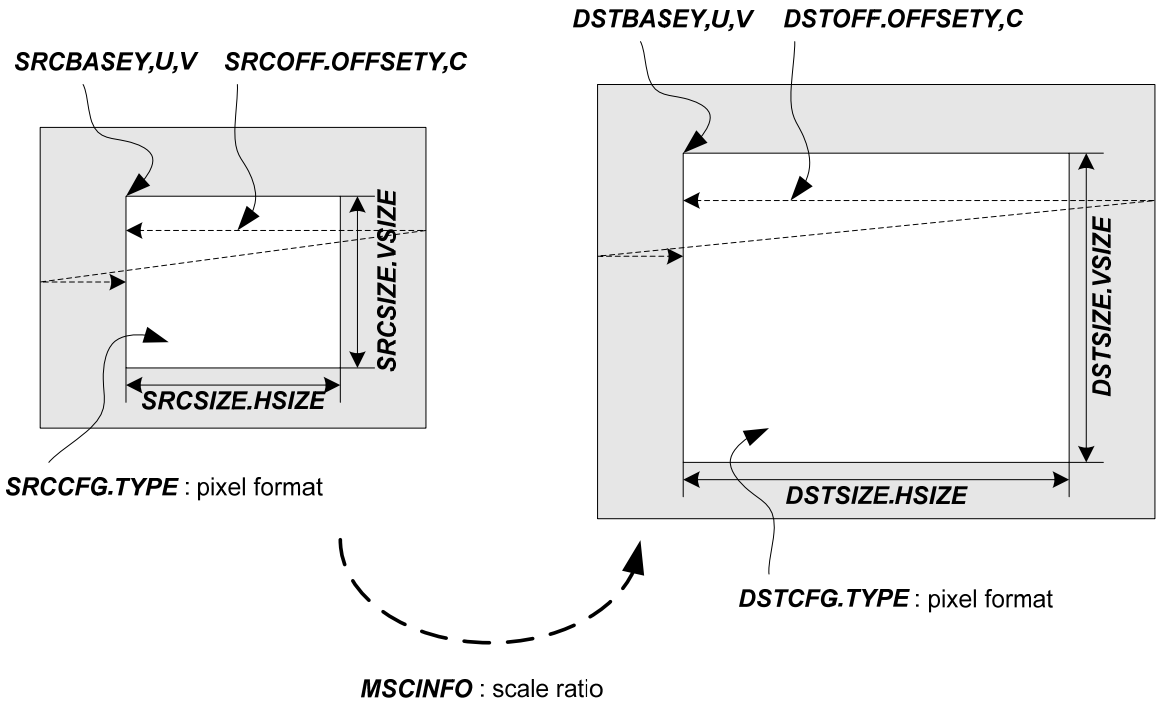


Figure 6.2 Memory to Memory Scaling Operation

The MSC reads the source image in the memory and resizes it and then stores the result image to the specified memory space or send it to the LCDC. Figure 6.2 shows that memory-to-memory scaling operation and the corresponding registers. The MSC has two modes to store the result image to the memory. One is the frame mode. The other is the rolling mode. The frame mode needs as much memory space as the whole result image size (Figure 6.3(a)). But, the rolling mode needs as much memory space as region which is specified by the rolling lines (Figure 6.3(b)).

In case of rolling mode, there are two methods the MSC informs the on-chip CPU the current status. One is to use the rolling line trigger level; the other is to use the middle line trigger level. Whenever the MSC stores one line to be scaled, the current rolling line counter (CRCNT register) increases by 1. When this counter reaches the rolling line trigger level (ROLLCNT of DSTRMCNT register), it is reset to 0 and addresses which specifies the location to store the result image return to destination base addresses (DSTBASEY, DSTBASEU, and DSTBASEV) and the MSC can issue the interrupt request. The MSC can also inform the on-chip CPU how many lines are stored from the destination base address. When the current rolling line counter reaches the middle line trigger level (MIDCNT of DSTRMCNT register), the MSC can issue the interrupt request. But, in this case, the rolling line counter is not reset to 0 and address does not return to the destination base address.

Additionally, the MSC can make scaling operation stopped temporarily. The middle line trigger level and the rolling line trigger level are used as the indicator. Whenever the current rolling line counter reaches middle line and/or the rolling line trigger level, the MSC pauses in the operation. To resume the operation, RLS bit of MSCCTR register is set to 1. To enable this function, RGSM and MGSM of MSCCTR register should be set to 1. RGSM is for the rolling line trigger level and MGSM is for the middle line trigger level.

To enable the rolling mode, REN bit of MSCCTR register should be set to 1. The rolling line trigger level is determined by ROLLCNT of DSTRMCNT register. To enable the middle line trigger level, MEN and REN bit of MSCCTR register should be set to 1 and the middle line trigger level is determined by MIDCNT of DSTRMCNT register. And to enable the corresponding interrupt, RIQREN and MIRQEN should be set to 1. They are for the rolling line trigger level interrupt and the middle line trigger level interrupt, respectively.

Finally, when the whole result image is stored to the memory completely, MSCSTR.BUSY bit is cleared to 0 and MSCSTR.RDY bit is set to 1. And these can be used as an interrupt request source.

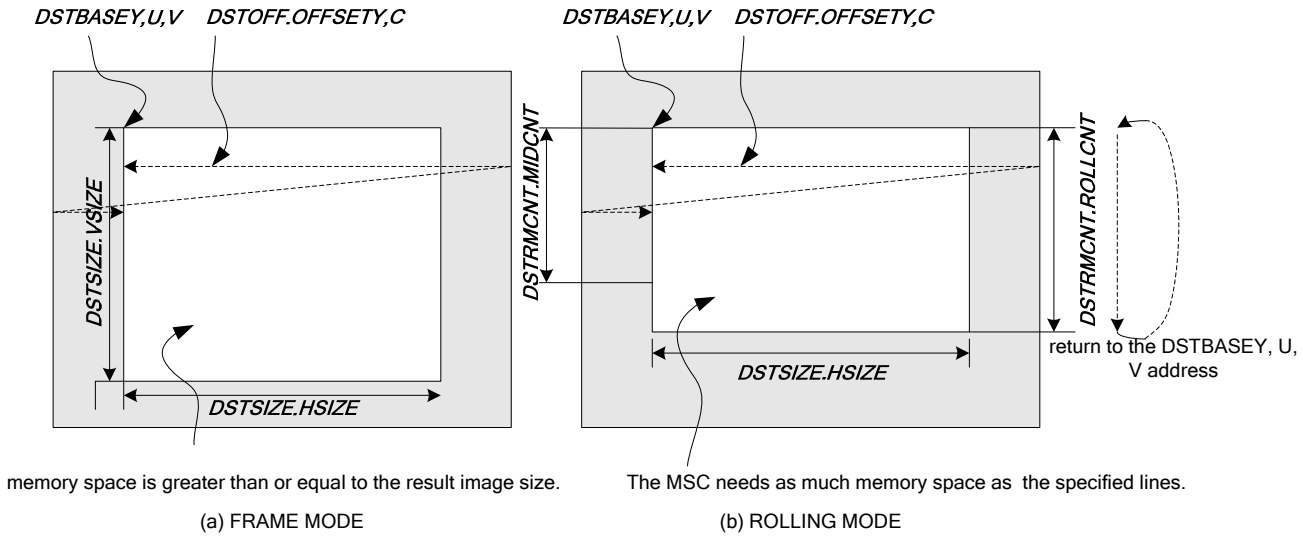


Figure 6.3 Storing the Result Image

6.3 Registers

Table 6.1 Scaler Registers (Base Address = 0xF0210000/0xF022000)

Name	Addr	Type	Reset	Description
SRCBASEY	0x000	R/W	0x00000000	Scaler source base address for Y
SRCBASEU	0x004	R/W	0x00000000	Scaler source base address for U (Cb)
SRCBASEV	0x008	R/W	0x00000000	Scaler source base address for V (Cr)
SRCSIZE	0x00c	R/W	0x00000000	Source image size register
SRCOFF	0x010	R/W	0x00000000	Source image line offset register
SRCCFG	0x014	R/W	0x00000000	Source image configuration register
DSTBASEY	0x020	R/W	0x00000000	Scaler destination base address for Y
DSTBASEU	0x024	R/W	0x00000000	Scaler destination base address for U (Cb)
DSTBASEV	0x028	R/W	0x00000000	Scaler destination base address for V (Cr)
DST_SIZE	0x02c	R/W	0x00000000	Destination image size register
DSTOFF	0x030	R/W	0x00000000	Destination image line offset register
DSTCFG	0x034	R/W	0x00000000	Destination image configuration register
MSCINF	0x040	R/W	0x00000000	Scaling information register
MSCCTR	0x044	R/W	0x00000000	Scaler control register
MSCSTR	0x048	R/W	0x00000000	Scaler status register
HSTROBE	0x4C	R/W	0x000A0002	Horizontal Strobe Timing Control Register
DSTRMCNT	0x050	R/W	0x00000000	Destination Rolling Count Register
CRCNT	0x054	R	0x00000000	Destination Rolling Status Register
CLIP0	0x58	R/W	0x00ff00ff	RGB-to-YCbCr Clipping Configuration Register 0
CLIP1	0x5C	R/W	0x000000ff	RGB-to-YCbCr Clipping Configuration Register 1
VSTROBE	0x60	R/W	0x0000000a	Vertical Strobe Timing Control Register

SRC Image Y Base Address (SRCBASEY) 0xF021/0xF0220000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCBASEY[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCBASEY[15:2]															

Field	Name	RW	Reset	Description
31 ~ 2	SRCBASEY	RW	-	Source Base Address Y for 4:2:0 or 4:2:2 Separate Mode Y for 4:2:2 Sequential Mode Y for 4:2:0 or 4:2:2 Interleaved Mode

SRC Image U Base Address (SRCBASEU) 0xF021/0xF0220004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCBASEU[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCBASEU[15:2]															

Field	Name	RW	Reset	Description
31 ~ 2	SRCBASEU	RW	-	Source Base Address for U Cb/Cr for 4:2:0 or 4:2:2 Interleaved Mode (TCC8800) SRCBASEU[31:30] and SRCBASEY[31:30] should be the same value.

SRC Image V Base Address (SRCBASEV) 0xF021/0xF0220008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCBASEV [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCBASEV[15:2]															

Field	Name	RW	Reset	Description
31 ~ 2	SRCBASEV	RW	-	Source Base Address for V SRCBASEV[31:30] and SRCBASEY[31:30] should be the same value.

SRC Image Size (SRCSIZE) 0xF021/0xF022000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				VSIZE[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				HSIZE[11:0]											

Field	Name	RW	Reset	Description
27 - 16	VSIZE	RW	0	Vertical Image Size by pixel unit
11-0	HSIZE	RW	0	Horizontal Image Size by pixel unit

SRC Image Offset (SRCOFF) 0xF021/0xF0220010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				OFFSETC[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OFFSETY[11:0]											

Field	Name	RW	Reset	Description
27 ~ 16	OFFSETC	RW	0x0	Chrominance Address Offset
11 ~ 0	OFFSETY	RW	0x0	Luminance Address Offset

SRC Image Configuration (SRCCFG) 0xF021/0xF0220014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSBF															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					WAITCNT				READY	INTM	INPATH	INTLV	TYPE		

MSBF [31]	Description
N	When RGB color used, the LSB is extended by 0: zero 1: MSB(Most Significant Bit) * The RGB color was extended to 8bits and converted to YCbCr color.

WAITCNT [10:8]	Description
N	Ready Cycle Count Register ***This is invalid for this processor – should be zero***.

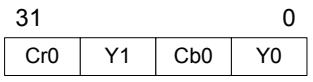
READY [6]	Description
N	Ready Control Register ***This is invalid for this processor – should be zero***.

INTM [5]	Description
N	Interleaved mode cb/cr order 0: Mode 0 1: Mode 1

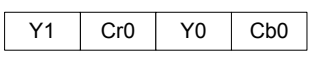
INPATH [4]	Description
N	Input Data path Configuration Register ***This is invalid for this processor – should be zero***.

INTLV [4]	Description
n	1: TYPE=2 or 3 YUV420 interleaved YUV422 interleaved 0: TYPE=2 or 3 YUV420 separate YUV422 separate

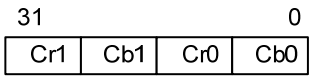
TYPE [2:0]	Description
n	7 :RGB454 6: RGB444 5: RGB555 4: RGB565 3 : YUV420 separate 2 : YUV422 separate 1 : YUV422 sequential mode 1 0 : YUV422 sequential mode 0



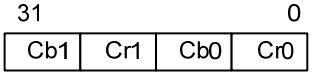
YCbCr4:2:2 Sequential Mode 0



YCbCr4:2:2 Sequential Mode 1



4:2:2/4:2: 0 Chrominance Interleaved Mode 0



4:2:2/4:2: 0 Chrominance Interleaved Mode 1

DST Image Y Base Address (DSTBASEY) 0xF021/0xF0220020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DSTBASEY[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTBASEY[15:2]															

Field	Name	RW	Reset	Description
31 ~ 2	DSTBASEY	RW	-	Destination Base Address Y for 4:2:0 or 4:2:2 Separate Mode Y for 4:2:2 Sequential Mode Y for 4:2:0 or 4:2:2 Interleaved Mode

DST Image U Base Address (DSTBASEU) 0xF021/0xF0220024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DSTBASEU[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTBASEU[15:2]															

Field	Name	RW	Reset	Description
31 ~ 2	DSTBASEU	RW	-	Destination Base Address for U Cb/Cr for 4:2:0 or 4:2:2 Interleaved Mode (TCC8800) DSTBASEU[31:30] and DSTBASEY[31:30] should be the same value.

DST Image V Base Address (DSTBASEV) 0xF021/0xF0220028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DSTBASEV[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTBASEV[15:2]															

Field	Name	RW	Reset	Description
31 ~ 2	DSTBASEV	RW	-	Destination Base Address for V DSTBASEV[31:30] and DSTBASEY[31:30] should be the same value.

DST Image Size (DSTSIZE) 0xF021/0xF022002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				VSIZE[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				HSIZE[11:0]											

Field	Name	RW	Reset	Description
27 - 16	VSIZE	RW	0	Vertical Image Size by pixel unit
11-0	HSIZE	RW	0	Horizontal Image Size by pixel unit

DST Image Offset (DSTOTFF) 0xF021/0xF0220030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				OFFSETC[11:0]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				OFFSETY[11:0]											

Field	Name	RW	Reset	Description
27 ~ 16	OFFSETC	RW	0x0	Chrominance Address Offset
11 ~ 0	OFFSETY	RW	0x0	Luminance Address Offset

DST Image Configuration (DSTCFG) 0xF021/0xF0220034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				COP	WAITCNT				RDY	INTM	PATH	INTLV	TYPE		

Field	Name	RW	Reset	Description
2 ~ 0	TYPE	RW	0x0	Destination Type Register 7 :RGB454 6: RGB444 5: RGB555 4: RGB565 3 : YUV420 separate 2 : YUV422 separate 1 : YUV422 sequential mode 1 0 : YUV422 sequential mode 1 * Byte Sequence is same as SRCCFG
3	INTLV	RW	0x0	1: TYPE=2 or 3 YUV420 interleaved YUV422 interleaved (TCC8800) 0: TYPE=2 or 3 YUV420 separate YUV422 separate
4	PATH	RW	0x0	Destination Type Register ‘0’ : Memory The scaled image is written to the destination memory. ‘1’ : LCD Channel 0 Direct Path The scaled image is written to the LCDC for IMG0.
5	INTM	RW	0x0	Interleaved mode cb/cr order 0: Mode 0 1: Mode 1
6	RDY	RW	0x0	Access Wait Control Register Valid for PATH being ‘1’ ‘0’ : Wait for “WAITCNT+1” cycles for LCD Access ‘1’ : Wait until Output FIFO is not empty
10 ~ 8	WAITCNT	RW	0x0	Wait Cycle Count for RDY being ‘1’
11	COP	RW	0x0	Chrominance Writing Mode Register Defined for 4:2:0 Separate Mode ‘0’ : Y0→U0→V0→Y1→Y2→U1 … ‘1’ : Y0→U0→Y1→V0→Y2→U1 …

MSC Information (MSCINFO) 0xF021/0xF0220040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		VRATIO[13:0]													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		HRATIO[13:0]													

Field	Name	RW	Reset	Description
29 ~ 16	VRATIO	RW	0x0	Vertical scale ratio VRATIO = 256 * SRCSIZE.VSIZE/DSTSIZE.VSIZE
13 ~ 0	HRATIO	RW	0x0	Horizontal scale ratio HRATIO = 256 * SRCSIZE.HSIZE/DSTSIZE.HSIZE

MSC Control (MSCCTR) **0xF021/0xF0220044**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TST		CKG				INPATH	REN	MEN			RLS		RGSM	MGSM
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R2YMD		Y2RMD		R2YCEN		OUIEN	RST	RIRQEN	MIRQEN	CONT	BP		BUSY	RDY	EN

Field	Name	RW	Reset	Description
0	EN	RW	0x0	Enable Register 0b : Operation Disabled 1b : Operation Enabled
1	RDY	RW	0x0	1b : Ready Interrupt Enable
2	BUSY	RW	0x0	1b : Busy Interrupt Enable
4	BP	RW	0x0	Bypass Enable Signal (Test Purpose) 0b : Not-bypassed (Normal Operation) 1b : Bypassed * This should be “ZERO”
5	CONT	RW	0x0	Continuous Mode 0 : One-Time Mode 1 : Continuous Mode **** If the output of the scaler connected to On-The-Fly and the connected device requires continuously such as LCD controller, this should be “1”.***
6	MIRQEN	RW	0x0	Enable an interrupt using the middle line counter 0b : disable 1b : enable
7	RIRQEN	RW	0x0	Enable an interrupt using the rolling line counter 0b : disable 1b : enable
8	RST	RW	0x0	Internal state machine Reset Signal 0b : Not Reset 1b : Reset
9	OUIEN	RW	0x0	Output Underrun IRQ Enable 0 : Disabled 1 : Enabled
11	R2YCEN	RW	0x0	Clipping Enable for RGB-to-YCbCr Conversion 0 : Disabled 1 : Enabled * Normally, ‘0’ is recommended.
13-12	Y2RMD	RW	0x0	YCbCr-to-RGB Conversion Mode
15-14	R2YMD	RW	0x0	RGB-to-YCbCr Conversion Mode
16	MGSM	RW	0x0	Enable Stop Mode using the middle line counter 0b : disable 1b : enable (MEN and REN should be also set to 1)
17	RGSM	RW	0x0	Enable Stop Mode using the rolling line counter 0b : disable 1b : enable (REN should be also set to 1)
19	RLS	RW	0x0	Release Stop Mode When scaling operation is stopped by “rolling line counter” or “middle line counter”, it is resumed by writing 1 to this bit.

				This bit is automatically cleared.
22	MEN	RW	0x0	Enable the middle line counter 0b : disable 1b : enable
23	REN	RW	0x0	Enable the rolling mode 0b : disable (frame mode) 1b : enable (rolling mode)
24	INPATH	RW	0x0	Read Path Configuration Register 0 : Read from Memory 1 : Read from On-the-fly Path
28	CKG	RW	0x0	Clock Gating Enable Signal 0b : Gating Enable 1b : Gating Disabled * This should be “ZERO”
30	TST	RW	0x0	Should be zero for test purpose

Y2RMD [10:9]	YCbCr to RGB Conversion Option
0	**** R = Y + 1.371 * (Cr – 128)*** **** G = Y + 0.336 * (Cb – 128) – 0.698 * (Cr – 128)*** **** B = Y + 1.732 * (Cb – 128)*** The range for “***YCbCr***” is 16 ~ 235, “***Studio Color***”. The result is “***Studio Color***” – Normally SDTV.
1	**** R = 1.164 * (Y – 16) + 1.596 * (Cr – 128)*** **** G = 1.164 * (Y – 16) – 0.391 * (Cb – 128) – 0.813 * (Cr – 128)*** **** B = 1.164 * (Y – 16) + 2.018 * (Cb – 128)*** The range for “***YCbCr***” is 16 ~ 235, “***Studio Color***”. The result is “***Computer System Color***” – Normally SDTV.
2	**** R = Y + 1.540 * (Cr – 128)*** **** G = Y – 0.183 * (Cb – 128) – 0.459 * (Cr – 128)*** **** B = Y + 1.816 * (Cb – 128)*** The range for “***YCbCr***” is 16 ~ 235, “***Studio Color***”. The result is “***Studio Color***” – Normally HDTV.
3	**** R = 1.164 * (Y – 16) + 1.793 * (Cr – 128)*** **** G = 1.164 * (Y – 16) – 0.213 * (Cb – 128) – 0.534 * (Cr – 128)*** **** B = 1.164 * (Y – 16) + 2.115 * (Cb – 128)*** The range for “***YCbCr***” is 16 ~ 235, “***Studio Color***”. The result is “***Computer System Color***” – Normally HDTV.

Notice !!! :

The coefficients of the above table are approximated for fixed point calculation.

R2YMD [29:28]	RGB to YCbCr Conversion Option
0	**** Y = 0.299*R + 0.587G + 0.114B*** **** Cb = -0.172*R – 0.339*G + 0.511*B + 128*** **** Cr = 0.511*R – 0.428*G – 0.083*B + 128*** The range for "***RGB***" is 16 ~ 235, "***Studio Color***". The result is "***Studio Color***" – Normally SDTV.
1	**** Y = 0.257*R + 0.504*G + 0.098*B + 16*** **** Cb = -0.148*R – 0.291*G + 0.439*B + 128*** **** Cr = 0.439*R – 0.368*G – 0.071*B + 128*** The range for "***RGB***" is 0 ~ 255, "***Computer System Color***" The result is "***Studio Color***" – Normally SDTV.
2	**** Y = 0.213*R + 0.715*G + 0.072*B*** **** Cb = -0.117*R – 0.394*G + 0.511*B + 128*** **** Cr = 0.511*R – 0.464*G – 0.047*B + 128*** The range for "***RGB***" is 16 ~ 235, "***Studio Color***".. The result is "***Studio Color***" – Normally HDTV.
3	**** Y = 0.183*R + 0.614*G + 0.062*B + 16*** **** Cb = -0.101*R – 0.338*G + 0.439*B + 128*** **** Cr = 0.439*R – 0.399*G – 0.040*B + 128*** The range for "***RGB***" is 0 ~ 255, "***Computer System Color***". The result is "***Studio Color***" – Normally HDTV.

Notice !!! :

The coefficients of the above table are approximated for fixed point calculation.

MSC Status (MSCSTR) **0xF021/0xF0220048**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											SCE	SCB	STS		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
							IOU	IR	IM	IBUSY	IRDY			BUSY	RDY

Field	Name	RW	Reset	Description
0	RDY	R	0x0	Ready Status Register 0b : Not-Ready 1b : Ready
1	BUSY	R	0x0	Busy Status Register 0b : Not-Busy Status 1b : Busy Status
4	IRDY	RW	0x0	Ready Interrupt Flag By writing '1', it would be cleared.
5	IBUSY	RW	0x0	Busy Interrupt Flag By writing '1', it would be cleared.
6	IR	RW	0	Rolling line interrupt Flag By writing '1', it would be cleared.
7	IM	RW	0	Middle line interrupt Flag By writing '1', it would be cleared.
8	IOU	RW	0	Output Underrun Interrupt Flag By writing '1', it would be cleared.
18 ~ 16	STS	R	0x0	Ready Status Signals (Test Purpose) [16] : Output Port Ready Status [17] : Input Port Ready Status [18] : Scaler Interface Ready Status
19	STB	R	0x0	Scaler Core Busy Flag (Test Purpose)
20	SCE	R	0x0	Scaler Error Flag (Test Purpose)

Horizontal Strobe Timing Control Register (HSTROBE) 0xF021/0xF0220048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HEN	HWAIT														
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PEN									PWAIT						

Field	Name	RW	Reset	Description
6-0	PWAIT	R/W	2	Number of Cycles from Current Pixel to Next Pixel
15	PEN	R/W	0x0	Pixel-to-Pixel Wait Cycle Control Enable 0b : Automatically Controlled by H/W 1b : Controlled by PWAIT In case of PEN = '0', the WaitCycle is 1. 256 <= HSCALE : 1 1. 128 <= HSCALE <256 : 1 2. 64 <= VSCALE < 127 : 3 3. 32 <= VSCALE < 64 : 7 4. 16 <= VSCALE < 32 : 15 5. 0 <= VSCALE < 16 : INVALID In case of PEN = '1', the WaitCycle is PWAIT.
30-16	HWAIT	RW	0xA	Number of Cycles from End-of-Line to Start-of-Next-Line
31	HEN	RW	0x0	Cycles are Determined 0 : Automatically Controlled by H/W 1 : Controlled by HWAIT In case of HEN = '0', the WaitCycle is 0x20 In case of HEN = '1', the WaitCycle is HWAIT.

Vertical Strobe Timing Control Register (VSTROBE) 0xF021/0xF0220060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VEN															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VWAIT															

Field	Name	RW	Reset	Description
15-0	VWAIT	R/W	0xA	Number of Cycles from Current Frame to Next Frame
31	VEN	R/W	0x0	Frame-to-Frame Wait Cycle Control Enable 0b : Automatically Controlled by H/W 1b : Controlled by V2VWAIT In case of VEN = '0', the WaitCycle is 1. 256 <= VSCALE : SRCHSIZE 2. 192 <= VSCALE < 256 : SRCHSIZE * 1.5 3. 128 <= VSCALE < 192 : SRCHSIZE * 2 4. 96 <= VSCALE < 128 : SRCHSIZE * 3 5. 64 <= VSCALE < 96 : SRCHSIZE * 4 6. 32 <= VSCALE < 64 : SRCHSIZE * 8 7. 16 <= VSCALE < 32 : SRCHSIZE * 16 8. 0 <= VSCALE < 16 : Not-Supported In case of VEN = '1', the WaitCycle is VWAIT.

DST Rolling/Middle Line Count (DSTRMCNT) 0xF021/0xF0220050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				ROLLCNT [11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				MIDCNT[11:0]											

Current Rolling Count (CRCNT) 0xF021/0xF0220054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				C_RCNT[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

C_ROLL_CNT	Description
	Number of lines that are stored the result image from the base address.

RGB-to-YCbCr Conversion Clipping Register (CLIP0) 0xF021/0xF0220058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLIPCBL								CLIPCBU							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLIPYL								CLIPYU							

Field	Name	RW	Reset	Description
7~0	CLIPYU	RW	0xFF	Upper bound for the Y
15~8	CLIPYL	RW	0x00	Lower bound for the Y
23~16	CLIPCBU	RW	0xFF	Upper bound for the Cb
31~24	CLIPCBL	RW	0x00	Lower bound for the Cb

RGB-to-YCbCr Conversion Clipping Register (CLIP1) 0xF021/0xF022005C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLIPCRL								CLIPCRU							

Field	Name	RW	Reset	Description
7~0	CLIPCRU	RW	0xFF	Upper bound for the Cr
15~8	CLIPCRL	RW	0x00	Lower bound for the Cr

7 NTSC / PAL Encoder Composite Output

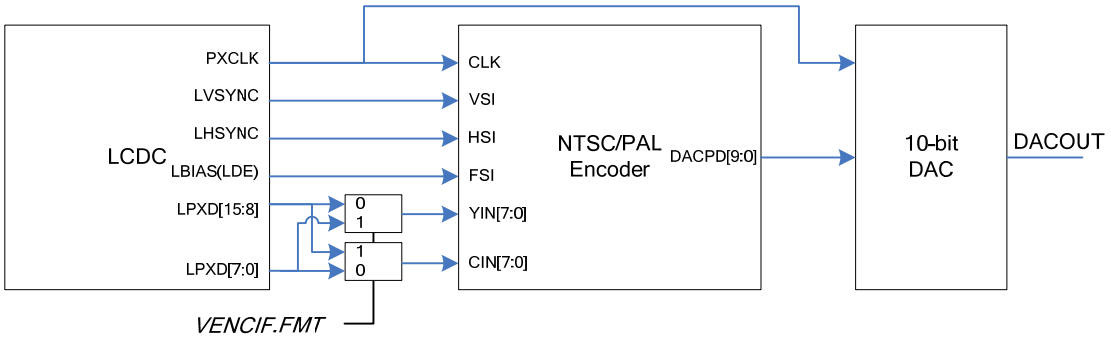
7.1 Overviews

The NTSC/PAL encoder meets all requirements of the ITU -R BT.470-3 specifications. This is designed to support all standards and variations of the NTSC and PAL encoding systems. This includes cross standards pseudo PAL and pseudo NTSC. This allows independent control of field rate, chroma subcarrier and the chroma encoding algorithm. Both the luma and chroma bandwidths can be varied to optimize for various data input conditions. Because input signals are generated by “LCD Interface”, LCD can not be displayed while NTSC/PAL composite output is enabled.

7.2 Features

- Encoding at square pixel or CCIR601 rates including flexible support for multiple clock frequencies
- Programmable Luma and Chroma bandwidth
- Programmable Saturation, Hue, Contrast and Brightness
- Supports all NTSC and PAL formats
  - NTSC-M/4.43, PAL-B/D/G/H/I/M/N/Combination N
- 8 bit or 16bit YUV input
  - ITU-R BT.601 4:2:2 16-bit Parallel Input
  - ITU-R BT.656 Parallel Input Format
- Controlled entry and exit of active video pixel and line
- Composite outputs.
- Outputs 10 bit to DAC
  - Maximum conversion rate : 27MSPS
  - Output Load Resistance : 37.5 ohm
  - Analog Output Range : 0.0 – 1.3V(Typical)

7.3 Selecting Operation Mode



To use the NTSC/PAL encoder, the LCDC should be configured as NTSC/PAL inter-face and output timing should be configured as the corresponding input timing of the NTSC.PAL encoder.
The NTSC/PAL encoder can operate at square pixel or 601 sampling rates for 50Hz or 60Hz standards. This is controlled using the PIXSEL and IFMT bits. Selecting the op-erating mode changes various internal timing locations to ensure that the final analog output signal meets appropriate standards. The operation frequencies and related in-formation are described below.

DESCRIPTION	SYMBOL	601 Sampling Clock		Square Pixel Clock	
		60Hz Field	50Hz Field	60Hz Field	50Hz Field
DAC Operation FREQ Pixel Data Rate	F_{CK27} F_{CK13}	27MHz 13.5MHz	27MHz 13.5MHz	24.5454MHz 12.2727MHz	29.5MHz 14.75MHz
Number of Clocks per Line	T_{HS}	1716	1728	1560	1888
Number of CK13(CK27) clocks for active video (Encoded pixels)	T_{AV}	720 (1440)	720 (1440)	640 (1280)	768 (1536)
Number of Lines per Field	L_{VS}	262.5	312.5	262.5	312.5
Default numbers of CK13 clocks from HS active edge to first input data	T_{FPX}	244	262	238	304

7.4 Input Timing

The 3 signals HSI, VSI and FSI are used to define the horizontal, vertical and field ref-erence points. The ISYNC[2:0] configuration register is used to select timing configura-tions. Once a timing mode is selected, each input has a polarity control (HSIP, VSIP and FSIP). These are used to define the “active” edge of the input sync. Finally, by programming the HOFFSET and VOFFSET bits, the desired pixel and line location for reset at the “active” edge is selected.

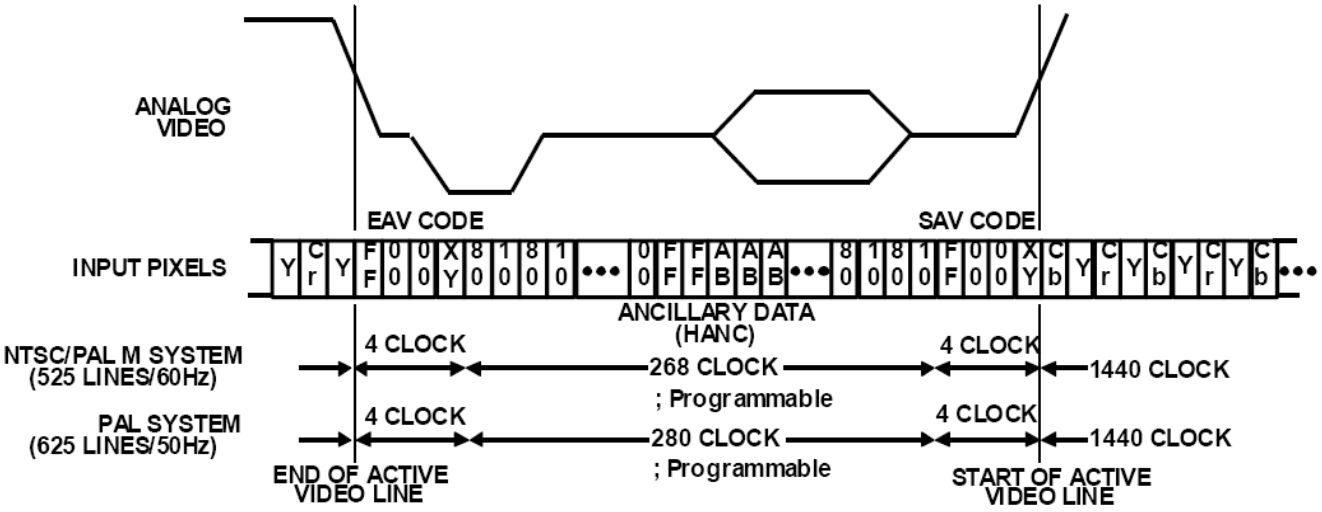


Figure 7.1 Digital Input Timing(ITU-R BT.656 8bit parallel Input)

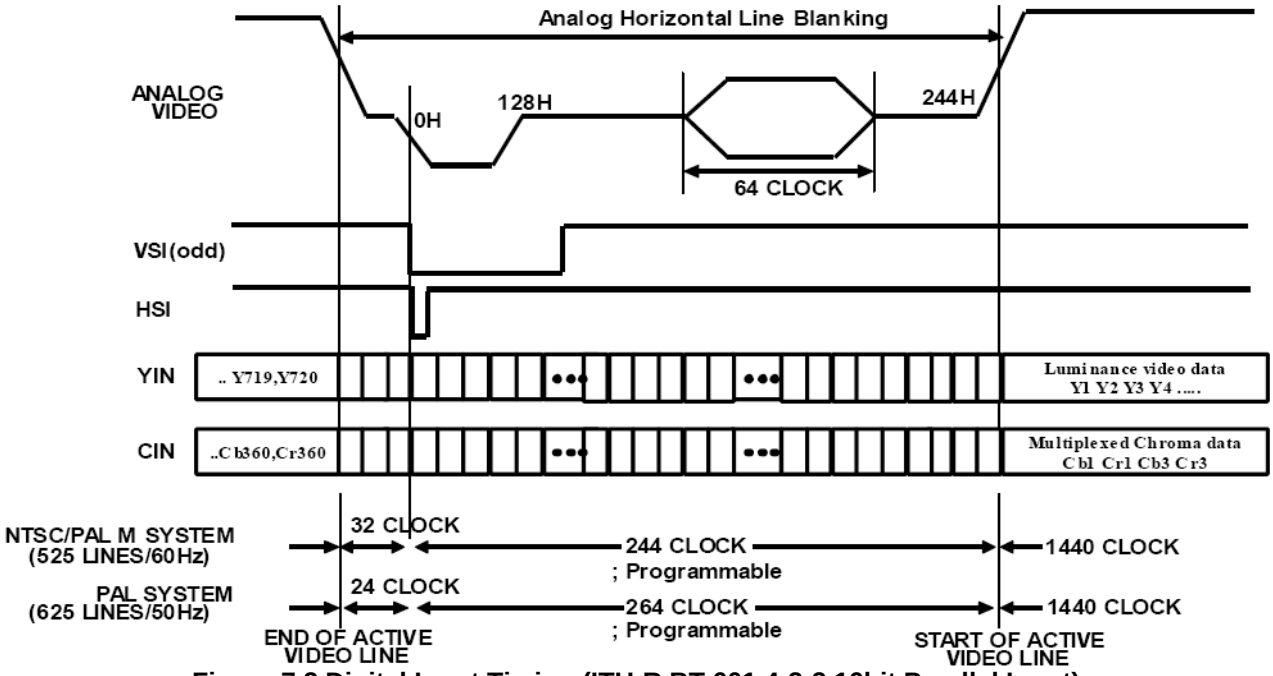


Figure 7.2 Digital Input Timing (ITU-R BT.601 4:2:2 16bit Parallel Input)

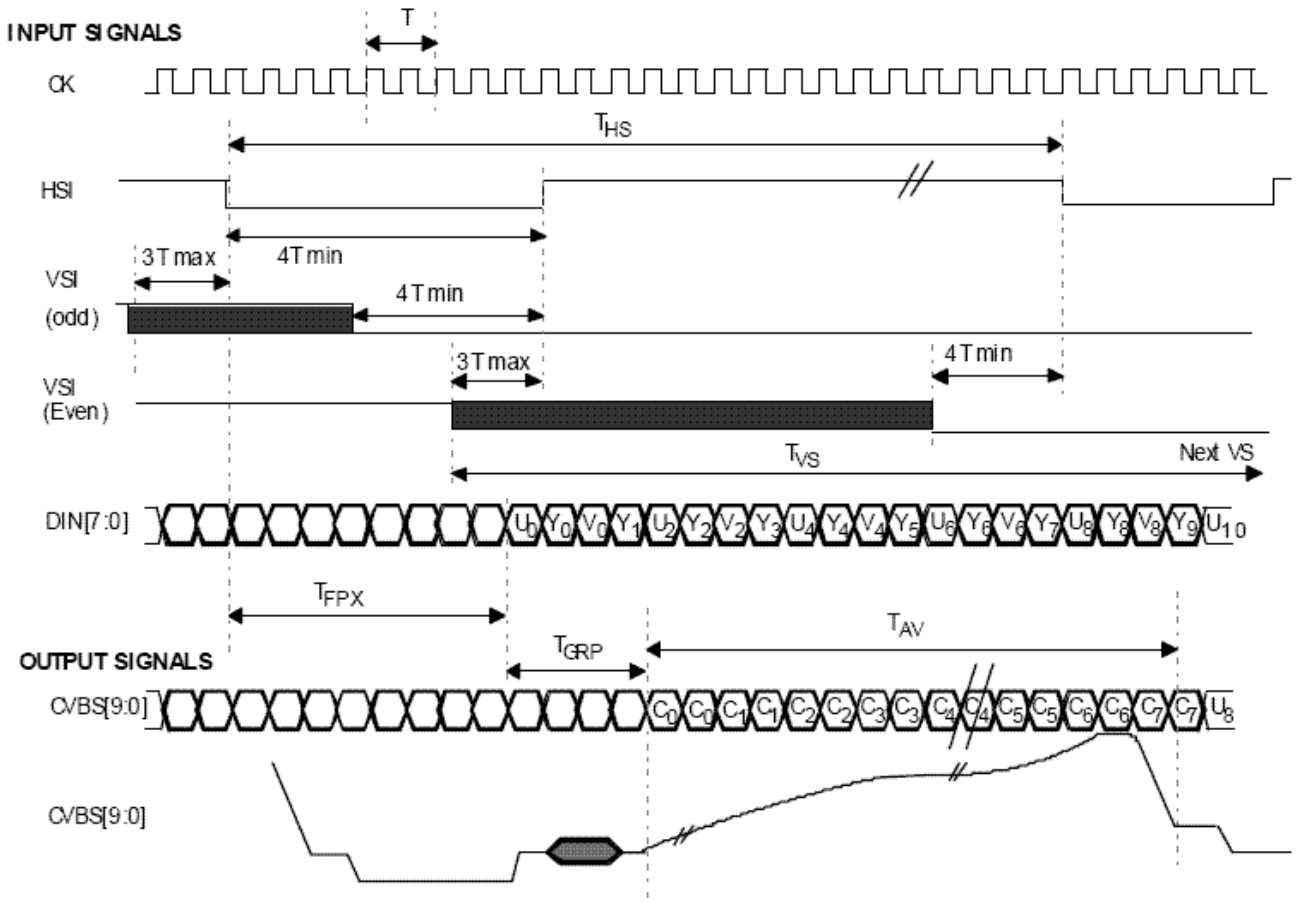


Figure 7.3 Example of Input Timing

7.5 Video Standard Selection

The NTSC/PAL encoder supports all NTSC and PAL standards throughout the world. The encoder supports independent control of the Chroma modulation frequency, selec-tion of phase alternating Line encoded chroma, and field frequency. In addition, for non SCH locked standards, the Chroma can be allowed to free run using the FDRST. If the FSCADJ register is set to any value other than '0', the SCH relationship will not be able to be maintained. For these setting it is also recommended that the FDRST bit be set to free run mode.

Table 7.1 STN LCD Palette Address

Requested output Format			Required Encoder Settings				
Format	Field Rate	FSC	IFMT	FSCSEL	PHALT	FDRST	PED
NTSC M	59.94Hz (525)	3.5795454	0	0	0	1	1/0
NTSC N	50 Hz (625)	3.5795454	1	0	0	0	1/0
PAL M	59.952 (525)	3.57561189	0	2	1	0	
PAL N	50 Hz (625)	3.58205625	1	3	1	0	
PAL B/G/H/I	50 Hz (625)	4.43361875	1	1	1	1	0
Pseudo PAL	60 Hz (525)	4.43361875	0	1	1	0	
Pseudo NTSC	50 Hz (625)	3.5795454	1	0	0	0	
NTSC 4.43	60Hz (525)	4.43361875	0	1	0	0	

7.6 Basic Video Adjustments

The standard video adjustments for Chroma and Luma are included. Chroma controls include Saturation and Hue (SAT HUE). Luma adjustments include Contrast and Brightness (CONTBRIGHT). An additional SCH control is included that changes the Chroma subcarrier phase relative to the horizontal sync. Note that this feature operates only when FDRST = 0.

7.7 Programmable Bandwidth

The data bandwidth for the Luma and chroma paths is shown in the following fre-quency plots. The YBW control allows 3 different settings to optimize the output band-width to the receiver. The same applies to chroma bandwidth using the CBW control.

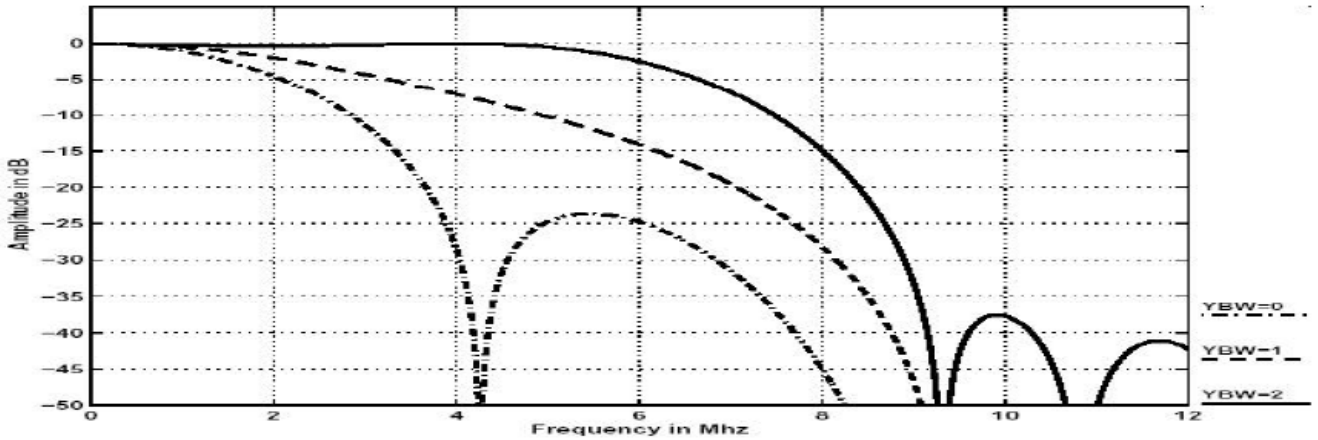


Figure 7.4 Luma Bandwidth

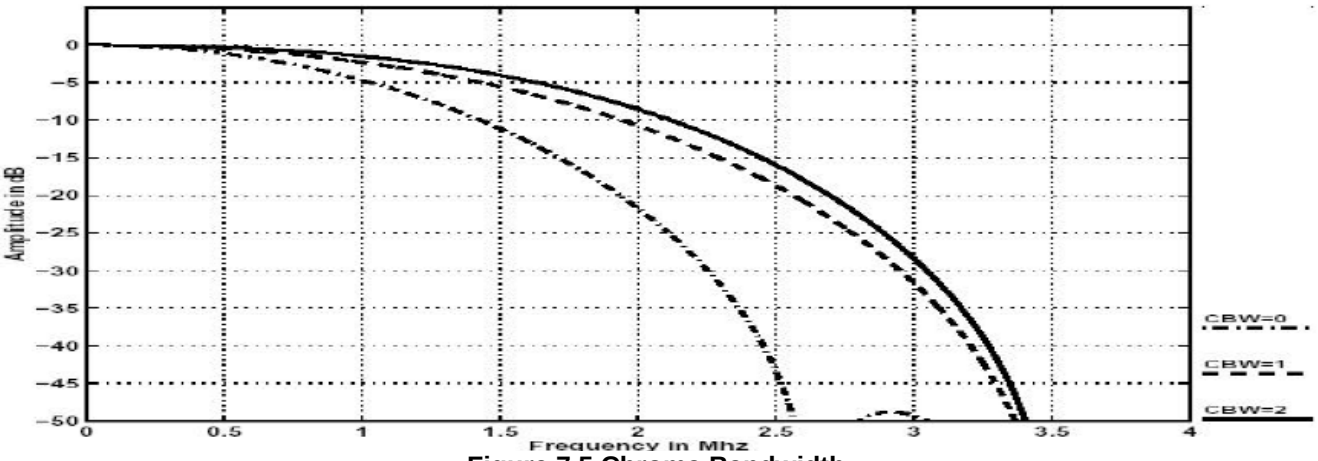


Figure 7.5 Chroma Bandwidth

7.8 Analog Video Output Configuration

The TCC8900 supports composite video. Refer to output configuration register DACSEL for further details. The DAC output levels and the associated digital codes are summarized below. DAC voltage assumes the standard 140IRE = 1v. Numbers are shown for NTSC type video with a pedestal.

Table 7.2 Summary of DAC voltage and Codes

Signal Level	CVBS / LUMA Value	IRE Value	DAC Voltage
Max output	1023	137.2	1.282v
100% White	810	100	1.015v
Black	282	7.37	353mv
Sync	12	-40	15mv
White - Blank	570	100	714mv delta
White - sync	798	140	1v delta
Color burst	228	40	285mv delta

7.9 Registers

STATA **0xF0240000**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								REVID					FIELD		

BIT	NAME	R/W	RESET	DESCRIPTION
7-5	REVID	R	0	Current Revision Identification Number
2-0	FIELD	R	-	The current video field 0-3 or 4-7 for NTSC fields 1-4 0-7 for PAL fields 1-8

ECMDA **0xF0240004**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PWDENC	FDRST	FSCCELL		PED	PIXEL	IFMT	PHALT

BIT	NAME	R/W	RESET	DESCRIPTION
7	PWDENC	R/W	0	Power Down Mode 0: Normal Operation 1 Power down mode for the entire digital logic circuits of encoder digital core.
6	FDRST	R/W	0	SCH Chroma-Luma locking control 0: Constant relation ship between color burst and horizontal sync maintained for appropriate video standards 1: Chroma is free running as compared to horizontal sync
5-4	FSCSEL	R/W	0	Modulation frequency of chroma output 0 Color subcarrier frequency = 3.57954545 MHz for NTSC 1 Color subcarrier frequency = 4.43361875 MHz for PAL - B,D,G,H,I,N 2 Color subcarrier frequency = 3.57561149 MHz for PAL - M 3 Color subcarrier frequency = 3.58205625 MHz for PAL – combination N
3	PED	R/W	0	Define input pedestal format 0 Video output has no pedestal*. 1 Video output has a pedestal.
2	PIXEL	R/W	0	Select Pixel sampling rate. 0 Input data is at 601 rates. * 1 Input data is at square pixel rates.
1	IFMT	R/W	0	Format of Output Data 0 525 Lines 1 625 Lines
0	PHALT	R/W	0	Phase Alternate control for PAL encoded chroma signal output 0 NTSC encoded color 1 PAL encoded color

ECMDB **0xF0240008**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											YBIBLK	CBW		YBW	

BIT	NAME	R/W	RESET	DESCRIPTION
4	VBIBLK	R/W	0	VBI Blanking control 0 Input data is passed through for non vbi processing lines 1 Video data is forced to black level for vertical non vbi processed lines.
3-2	CBW	R/W	0	Chroma Bandwidth control 0 Low bandwidth 1 Medium bandwidth 2 High bandwidth 3 Not used, default to low bandwidth
1-0	YBW	R/W	0	Luma Bandwidth control 0 Low bandwidth 1 Medium bandwidth 2 High bandwidth 3 Not used, default to low bandwidth

GLK **0xF024000C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											XT24	GLKEN	GLKE		GLKPL

BIT	NAME	R/W	RESET	DESCRIPTION
4	XT24	R/W	0	24MHz Crystal input when high 0 27MHz Crystal input 1 24MHz Crystal input
3	GLKEN	R/W	0	Genlock reset control 0 Genlock reset disable 1 Genlock reset enable
2-1	GLKE	R/W	0	Chroma Fsc Generation frequency control 0,1 Chroma Fsc is generated from internal constants based on current user setting. 2 Fsc is adjusted based on external RTCO input 3 Fsc tracks non standard Encoder clock (CLKI) programmed frequencies.
0	GLKPL	R/W	0	Genlock mode PAL ID Polarity control 0 PAL bit input polarity is active high 1 PAL bit input polarity is active low
0	PHALT	R/W	0	Phase Alternate control for PAL encoded chroma signal output 0 NTSC encoded color 1 PAL encoded color

SCH **0xF0240010**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SCH							

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	SCH	R/W	0	Programs the Color burst phase relation to the sync tip. '0' is the nominal value. The 8 bit control covers the entire 360 range as a 2's compliment number.

HUE **0xF0240014**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HUE							

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	HUE	R/W	0	Programs the Active video Color burst phase relative to color burst. The 8 bit control covers the entire 360 range as a 2's compliment number.

SAT **0xF0240018**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SAT							

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	SAT	R/W	0	Controls the Active video Chroma gain relative to the color burst gain. Value is 2's compliment with '0' the nominal value.

CONT **0xF024001C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CONT							

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	CONT	R/W	0	Controls Luma gain. Value is 2's compliment with '0' the nominal value.

BRIGHT **0xF0240020**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SCH							

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	BRIGHT	R/W	0	Controls Luma offset. Value is 2's compliment with '0' the nominal value.

FSC_ADJM **0xF0240024**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FSC_ADJM							

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	FSC_ADJM	R/W	0	FSC_ADJ[15:8]

FSC_ADJL **0xF0240028**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FSC_ADJ[7:0]							

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	FSC_ADJL	R/W	0	FSC_ADJ[7:0]

FSC_ADJ[7:0] allows the Pixel clock to be varied up to +/- 200ppm of its nominal value. This allows dot crawl adjustment. This 16 bit signal is multiplied by 4 and added to the internal chroma frequency constant.

ECMDC **0xF024002C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CSMDE	CSMD		RGBSYNC				

BIT	NAME	R/W	RESET	DESCRIPTION
7	CSMDE	R/W	0	Composite sync CSYN Pin tri-state control 0 Composite sync mode not enable - Pin is tri-stated 1 Composite sync mode enable - Output reflexes mode defined by CSMD
6-5	CSMD	R/W	0	Composite sync CSYN Pin output control 0 Composite Sync 1 Keyed clamp 2 Keyed pulse 3 N/A
4-3	RGBSYNC	R/W	0	Enable RGBSYNC when output is configured for analog EGB modes 0 No sync on the RGB output signals 01 Sync on the RGB output signals 10 Sync on the G output signal

DACSEL **0xF0240040**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												DACSEL			

BIT	NAME	R/W	RESET	DESCRIPTION
7-4	-	R/W	0	SHOULD BE ZERO
3-0	DACSEL	R/W	0	Data type output selection for DAC. 0: DAC digital output is disabled, Output is code 0. 1: Data output in CVBS format – Composite Video others : N/A

DACPD **0xF0240050**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															PD

BIT	NAME	R/W	RESET	DESCRIPTION
7-1	-	R/W	0	SHOULD BE ZERO
0	PD	R/W	0	0 DAC Normal Operation 1 DAC Power down

ICNTL **0xF0240080**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FSIP	VSIP	HSIP	HSVSP	VSMD	ISYNC		

BIT	NAME	R/W	RESET	DESCRIPTION
7	FSIP	R/W	0	Field polarity. It controls the polarity of any F or Field control as input or output. 0 Odd field (fields 1,3,5,7) active low. 1 Odd field active high.
6	VSIP	R/W	0	Vertical field polarity 0 Vertical sync active low. 1 Vertical sync active high.
5	HSIP	R/W	0	Horizontal sync polarity 0 Horizontal sync active low. 1 Horizontal sync active high.
4	HSVSP	R/W	0	Horizontal and vertical sync latch enable 0 Enable to latch on the falling edge of VS and HS . 1 Enable to latch on the rising edge of VS and HS.
3	VSMD	R/W	0	Vertical Sync Output format 0 Odd Field VS and FS outputs aligned to video line Start. Even Field VS and FS output aligned to video line midpoint. 1 Odd Field VS and FS outputs aligned to video line Start. Even Field VS and FS outputs aligned to video line Start.
2-0	ISYNC	R/W	0	Timing configuration for Horizontal, Vertical and Field inputs. 0 Alignment input format from pin FSI. F --- Field timing from the FSI Pin directly. V --- Vertical timing from the FSI Pin rising or falling edge. H --- Horizontal timing from the FSI Pin rising or falling edge. 1 Alignment input format from pins HSI, VSI, FSI. F --- Field timing from the FSI pin. V --- Vertical timing from the VSI pin rising or falling edge. H --- Horizontal timing from the HSI pin rising or falling edge. 2 Alignment input format from pins HSI and VSI. F --- Field timing from the latched value of HSI and VSI rising or falling edge. V --- Vertical timing from the VSI pin rising or falling edge. H --- Horizontal timing from the HSI pin rising or falling edge. 3 Alignment input format from pins VSI and FSI. F --- Field timing from the FSI pin directly. V --- Vertical timing from the VSI pin rising or falling edge. H --- Horizontal timing from the VSI pin rising or falling edge. 4 Alignment input format from pin VSI Only. Note that there is no field information in this mode so the software counter reset is required for proper field identification. F --- Field timing from software reset only. V --- Vertical timing from the VSI pin rising or falling edge. H --- Horizontal timing from the VSI pin rising or falling edge. 5 Alignment is from embedded EAV,SAV codes (line by line). F --- Field timing from the SAV/EAV ‘F’ bit. V --- Vertical timing from the SAV/EAV ‘V’ bit. H --- Horizontal timing from the SAV/EAV ‘H’ bit. 6 Alignment is from embedded EAV,SAV codes (frame by frame). F --- Field timing from the SAV/EAV ‘F’ bit. V --- Vertical timing from the SAV/EAV ‘F’ bit. --- Horizontal timing from the SAV/EAV ‘F’ bit. 7 F,V,H Alignment is Free Running (Master Mode).

HVOFFST **0xF0240084**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								INSEL				VOFFST	HOFFSET[10:8]		

BIT	NAME	R/W	RESET	DESCRIPTION
7-6	INSEL	R/W	0	Digital input format select. 0 format is 16 bit YUV 4:2:2 data Sampled at 27MHz. 1 format is 16 bit YUV 4:2:2 data Sampled at 13.5MHz . 2 Input format is 8 bit YUV 4:2:2 data 27MHz data Samples at 13.5MHz. 3 Input is from TESTI port, same format as mode 2.
3	VOFFST	R/W	0	Vertical offset bit 8. Refer to VOFFST register. VOFFSET[8:0] determines Vertical alignment point for the VSI input. Programmed value is the “line number” that pixel counter is reset to when a VSI transition in input.
2-0	HOFFSET	R/W	0	Horizontal offset bit 8 ~ 10. Refer to HOFFST register. HOFFSET[10:0] determines Horizontal alignment point for the HSI input. Programmed value is the “pixel number” that pixel counter is reset to when a HIS transition in input.

HOFFST **0xF0240088**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HOFFSET[7:0]							

Refer to the HVOFFST register description.

VOFFST **0xF024008C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								VOFFSET[7:0]							

Refer to the HVOFFST register description.

HSVSO **0xF0240090**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									VSOB	HSOB[10:8]			HSOE[10:8]		

BIT	NAME	R/W	RESET	DESCRIPTION
6	VSOB	R/W	0	VSOB[8]. Programs the Leading edge of the Vertical sync Output. This signal is not used inside the core. It is provided a reference for master mode timing.
5-3	HSOB	R/W	0	HSOB[10:8] Programs the Leading edge of the Horizontal sync output. The HSO signal is not used.
2-0	HSOE	R/W	0	HSOE[10:8] Programs the Trailing edge of the Horizontal sync output. The HSO signal is not used inside the core. It is provided as a source for master timing.

HSOE **0xF0240094**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSOE[7:0]							

Refer to HSVSO register.

HSOB **0xF0240098**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSOB [7:0]							

Refer to HSVSO register.

VSOB **0xF024009C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								VSOB [7:0]							

VSOE **0xF02400A0**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								VSOST		NOVRST	VSOE				

BIT	NAME	R/W	RESET	DESCRIPTION
6	VSOST	R/W	0	Programs the Vertical sync relative location for Odd and Even Fields.
5	NOVRST	R/W	0	No Vertical Reset, working in conjunction with EPRGOUT. 0 Normal operation (interlaced output timing). 1 No vertical reset on every field.
2-0	HSOE	R/W	0	Programs the Trailing edge of the Vertical sync Output. This signal is not used inside the core. It is provided a reference for master mode timing.

VENCON **0xF0240800**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															EN

BIT	NAME	RESET	DESCRIPTION
31-1			RESERVED
0	EN	0	It determines whether output signals of the LCDC is connected to the NTSC/PAL encoder. 0: disable 1: Enable (connection)

VENCIF **0xF0240804**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														MV	FMT

BIT	NAME	RESET	DESCRIPTION
31-2			RESERVED
1	MV	0	RESERVED
0	FMT	0	It determines data bus connection type between LCDC and NTSC/PAL encoder. 0: LCDC PXDATA[7:0] is connected to NTSC/PAL CIN[7:0] LCDC PXDATA[15:0] is connected to NTSC/PAL YIN[7:0] 1: LCDC PXDATA[7:0] is connected to NTSC/PAL YIN[7:0] LCDC PXDATA[15:8] is connected to NTSC/PAL CIN[7:0]

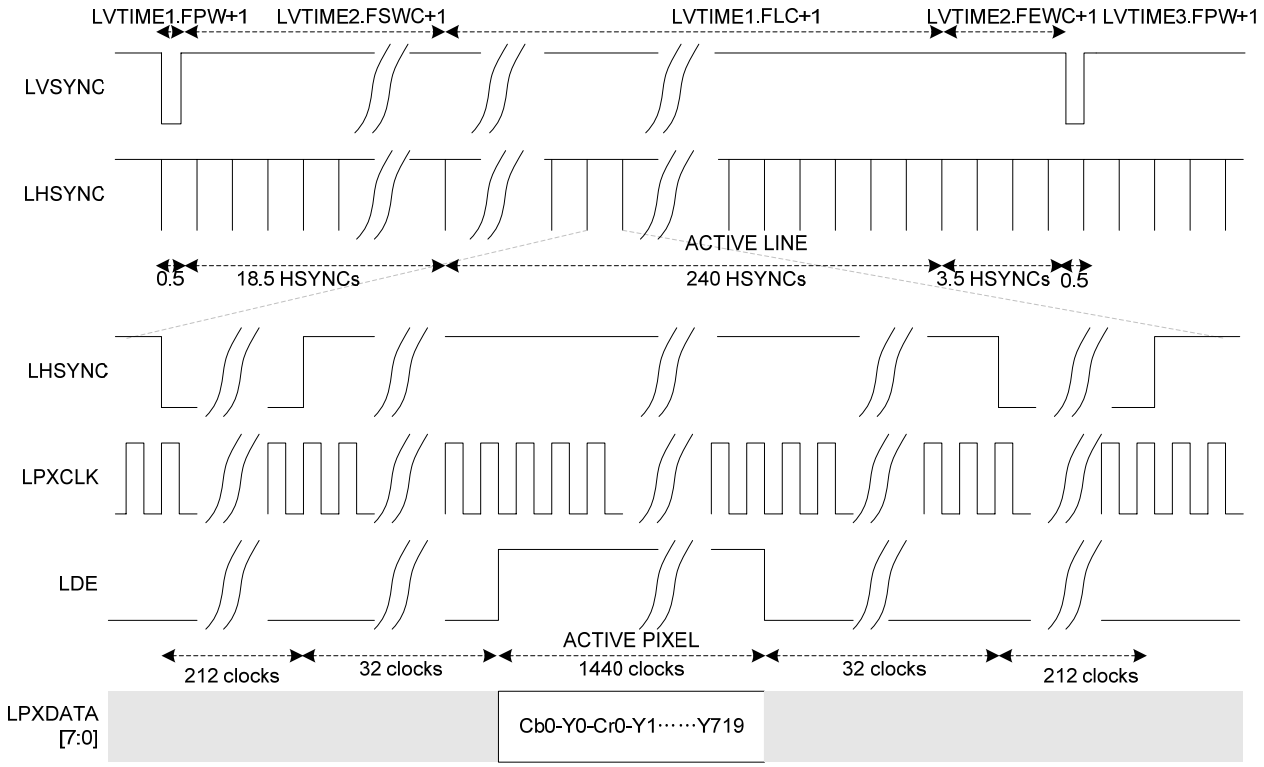
7.10 Example for NTSC/PAL Interface

We will strongly recommend you to use the values in the tables.

NTSC-M interface

The following tables show how to configure the LCDC and the NTSC/PAL encoder for NTSC. PXCLK of the LCDC should be configured as 27MHz.

LCDC register configuration for NTSC-M	
LDS.HSIZE = 720	LCTRL.TV = 1
LDS.VSIZE = 480	LCTRL.NI = 0
LHTIME1.LPC = 720 * 2 -1	LCTRL.DP = 0
LHTIME1.LPW = 212 - 1	LCTRL.PXDW = 6
LHTIME2.LSWC = 32 - 1	LCTRL.IH = 1
LHTIME2.LEWC = 32 - 1	LCTRL.IV =1
LVTIME1.FLC = 480 – 1	LCTRL IP = 1
LVTIME1.FPW = 1 - 1	
LVTIME2.FSWC = 37 -1	
LVTIME2.FEWC = 7 -1	
LVTIME3.FLC = 480 – 1	
LVTIME3.FPW = 1 – 1	
LVTIME4.FSWC = 38 – 1	
LVTIME4.FEWC = 6 - 1	



NTSC/PAL encoder register configuration for NTSC-M	
DACPD. PD = 0 (not power down)	HOFFST.HOFFSET = 0
ECMDA.PWDENC = 0	VOFFST.VOFFSET =1
ECMDA.FDRST = 1	
ECMDA.PHALT = 0	HVOFFST.HOFFSET = 0
ECMDA.IFMT = 0	HVOFFST.VOFFST = 0
ECMDA.PIXSEL = 0	HVOFFSET.INSEL = 2
ECMDA.PED = 1	
ECMDA.FSCSEL = 0	
ECMDB.VBIBLK = 0	
DACSEL.DACSEL = 1 (composite output)	
ICNTL.ISYNC = 2	
ICNTL.HSVSP = 1	
ICNTL.HSIP = 0	
ICNTL.VSIP = 1	
ICNTL.FSIP = 0	
VENCIF.FMT = 1	
VENCON.EN = 1	

PAL-B/G/H/I interface

The following tables show how to configure the LCDC and the NTSC/PAL encoder for PAL-B/G/H/I. PXCLK of the LCDC should be configured as 27MHz.

LCDC register configuration for PAL B/G/H/I	
LDS.HSIZE = 720	LCTRL.TV = 1
LDS.VSIZE = 576	LCTRL.NI = 0
LHTIME1.LPC = 720 * 2 -1	LCTRL.DP = 0
LHTIME1.LPW = 128 - 1	LCTRL.PXDW = 6
LHTIME2.LSWC = 138 - 1	LCTRL.IH = 1
LHTIME2.LEWC = 22 - 1	LCTRL.IV =1
LVTIME1.FLC = 576 – 1	LCTRL IP = 1
LVTIME1.FPW = 1 - 1	
LVTIME2.FSWC = 43 -1	
LVTIME2.FEWC = 5 -1	
LVTIME3.FLC = 576 – 1	
LVTIME3.FPW = 1 – 1	
LVTIME4.FSWC = 44 – 1	
LVTIME4.FEWC = 4 - 1	

NTSC/PAL encoder register configuration for PAL B/G/H/I	
DACPD. PD = 0 (not power down)	HOFFST.HOFFSET = 0
ECMDA.PWDENC = 0	VOFFST.VOFFSET =1
ECMDA.FDRST = 1	
ECMDA.PHALT = 1	HVOFFST.HOFFSET = 0
ECMDA.IFMT = 1	HVOFFST.VOFFST = 0
ECMDA.PIXSEL = 0	HVOFFSET.INSEL = 2
ECMDA.PED = 0	
ECMDA.FSCSEL = 1	
ECMDB.VBIBLK = 0	
DACSEL.DACSEL = 1 (composite output)	
ICNTL.ISYNC = 2	
ICNTL.HSVSP = 1	
ICNTL.HSIP = 0	
ICNTL.VSIP = 1	
ICNTL.FSIP = 0	
VENCIF.FMT = 1	
VENCON.EN = 1	

7.11 Copy Generation Management Systems

Copy Generation Management Systems (CGMS) is supported for 525/60 for Japan. Line 20 and 283 are used to transmit the CGMS data when the CGMS function is en-abled. The CGMS consists of a 2 bit start code and a 20 bit data. The 20 bits data, in-cluding 6 bits of Cyclic Redundancy Check, are loaded into the CGMS register via the host interface prior to being transmitted. The data are transmitted at a rate of 447.443 kHz per bit with the amplitude ranging from 0 IRE to 70 IRE.

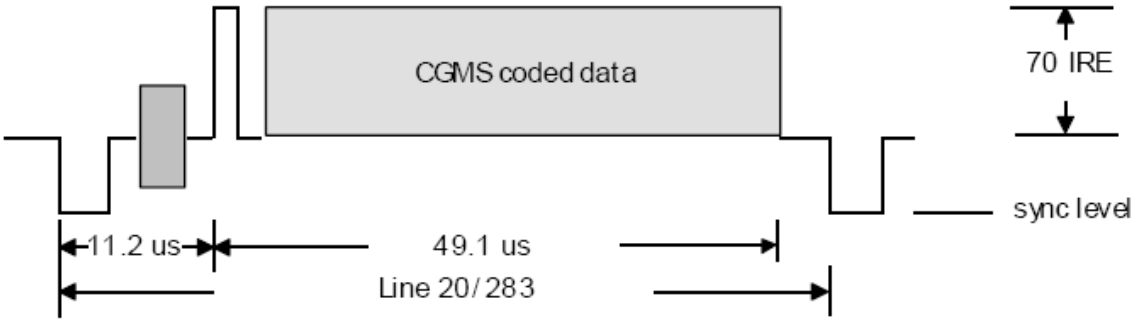


Figure 7.6 Copy Generation Management Systems

VSTAT **0xF02400C0**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														CGRDY	

BIT	NAME	RW	RESET	DESCRIPTION
Others				RESERVED
1	CGRDY	R	0	Copy Generation Management System Status (READ ONLY) 0: CGMS is able to set. 1: CGMS is set already.

VCTRL **0xF02400C4**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								0	0	0	0	0	CGOE	CGEE	0

BIT	NAME	RW	RESET	DESCRIPTION
Others				Should Be Zero
2	CGOE	R/W	0	Copy Generation Management System enable odd field. 0: CGMS is not enabled. 1: CGMS is enabled.
1	CGEE	R/W	0	Copy Generation Management System enable even field. 0: CGMS is not enabled. 1: CGMS is enabled.

CGMSA **0xF02400E0**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										CGMS6	CGMS5	CGMS4	CGMS3	CGMS2	CGMS1

CGMSB **0xF02400E4**

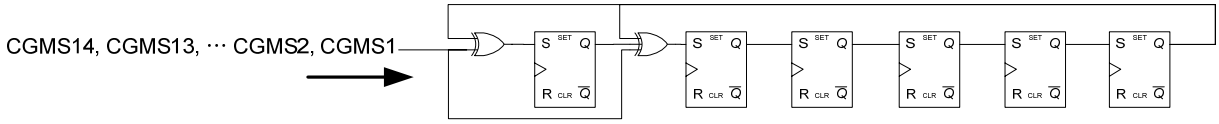
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CGMS14	CGMS13	CGMS12	CGMS11	CGMS10	CGMS9	CGMS8	CMGS7

CGMS8	CGMS7	DESCRIPTION
0	0	Copying permitted without restriction.
0	1	One copy permitted.
1	0	No more copies.(One copy has already been made.)
1	1	No copying permitted.

CGMSC **0xF02400E8**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										CGMS20	CGMS19	CGMS18	CGMS17	CGMS16	CGMS15

CGMSA, CGMSB, and CGMSC registers are 20bits CGMS data. Especially, CGMSC register is for the CRC for CGMSA and CGMSB and is manually set according to CGMSA and CGMSB value. Figure 7.7 shows example of the CRC calculator for CGMS. For example, when CGMSA=0x00 and CGMSB=0x03, CGMSC (CRC) should be set to 0x3A.



The CRC used is $X^6 + X + 1$, all preset to "1".

Figure 7.7 Example: CRC calculator for CGMS

7.12 10-Bit DAC

10-bit DAC is used for composite video signal output.

7.12.1 HDMI Specific Features

- Resolution : 10-bit
- Maximum conversion rate : 27MSPS
- BGR (Internal / External)
- Power down mode
- Output load resistance : 37.5Ω
- Analog output range : 0.0 ~ 1.3V(Typical)
- Power supply : 3.0V single

7.12.2 Functional Description

The DAC is initially power on state. To make the DAC enter the power down mode, the PD bit of DACPD register in NTSC/PAL encoder should be set to 1

The DAC uses reference current to decide the 1LSB current size by dividing the reference current by 31 times. To adjust the full current output, you must decide the "R(IREF)" resister value(connected to IREF pin) and "VREF" voltage value(connected to VREF pin). Its voltage output can be obtained by connecting RLOAD (connected to DACOUT pin).

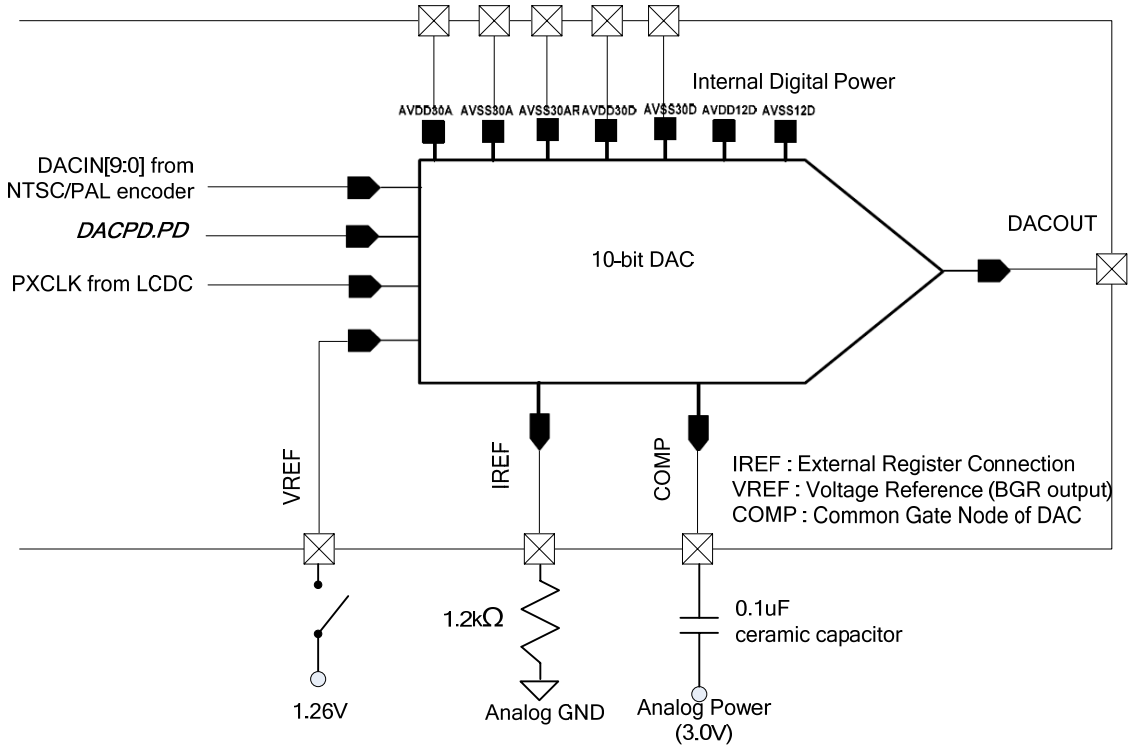


Figure 7.8 10-bit DAC Connection

The voltage output of DAC are decided by R(IREF), RLOAD, and V(VREF).

$$VO = \frac{V(VREF)}{R(IREF) \times 31} \sum_{i=0}^{9} (2^i \times DACIN[i]) * R_{LOAD}$$

For example, when the corresponding powers are applied to DAC power pins, VREF might be about 1.26V. If 1.2kΩ is connected to IREF pin and DACIN bits are all 1s and RLOAD is 37.5Ω, DACOUT is about 1.3V. in case R(IREF) is 1.5kΩ, DACOUT is about 1.04V. If you cannot change resister value connected to IREF pin, you can apply the corresponding voltage to VREF pin to adjust VO voltage.

8 HDMI

8.1 Overview

TCC8900 is comprised of an HDMI Tx controller with I2S/SPDIF input interface, CEC block and AES block for HDCP Key encryption.

8.1.1 HDMI Specific Features

- HDMI 1.3, HDCP 1.1, DVI 1.0 Complaint
- Supports Video format:
  480p @59.94Hz/60Hz, 576p@50Hz
  720p @50Hz/59.94Hz/60Hz
  1080i @50Hz/59.94Hz/60Hz
  1080p @50Hz/59.94Hz/60Hz
- Other various formats up to 148.5 MHz Pixel Clock
- Supports Color Format : 4:4:4 RGB/YCbCr , 4:2:2 YCbCr 12-bit mode
- Pixel Repetition (Up to x4)
- Supports Bit Per Color : 8bit, 10bit ,12bit (16bit is not supported)
- Dedicated block for CEC function
- Supports Audio Sample packets, DSD packets, HBR packets and DST packets for audio
- Integrated HDCP Encryption Engine for Video/Audio content protection
- Not include DDC (recommend to use separate I2C)

8.2 Architecture

TCC8900 HDMI controller has following interface.
- APB and I2C for register configuration
- Dedicated Video input interface
- Dedicated SPDIF input interface
- Dedicated Audio input interface for I2S

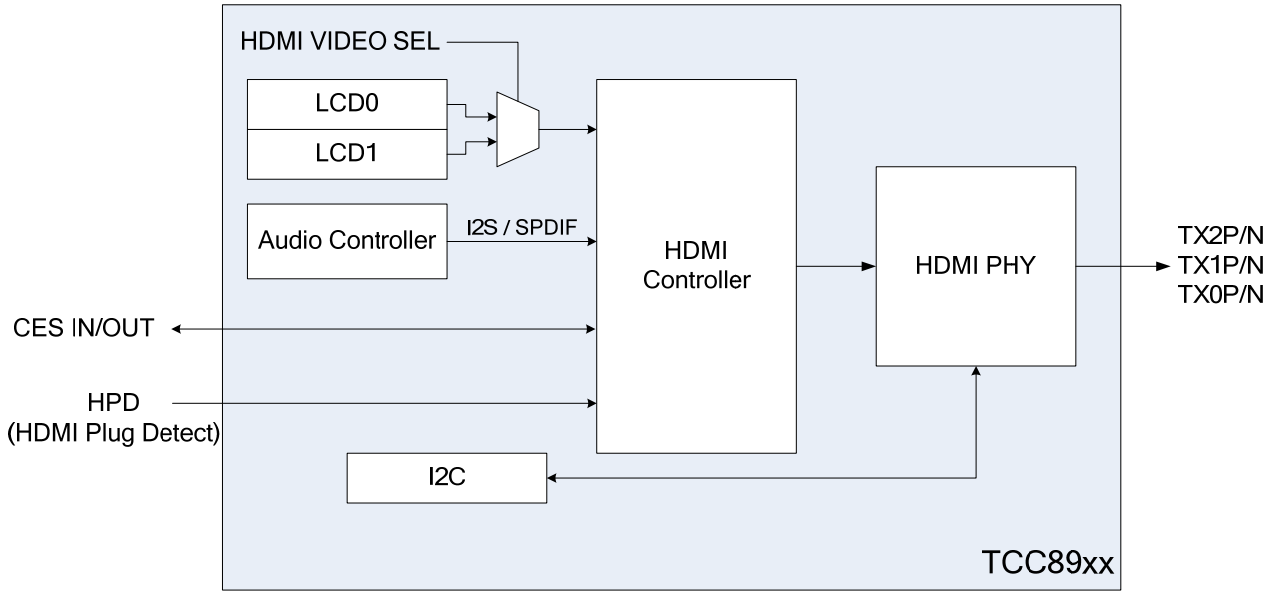


Figure 8.1 HDMI LINK and PHY in TCC8900

8.3 HDMI Controller (LINK)

8.3.1 HDCP KEY Management

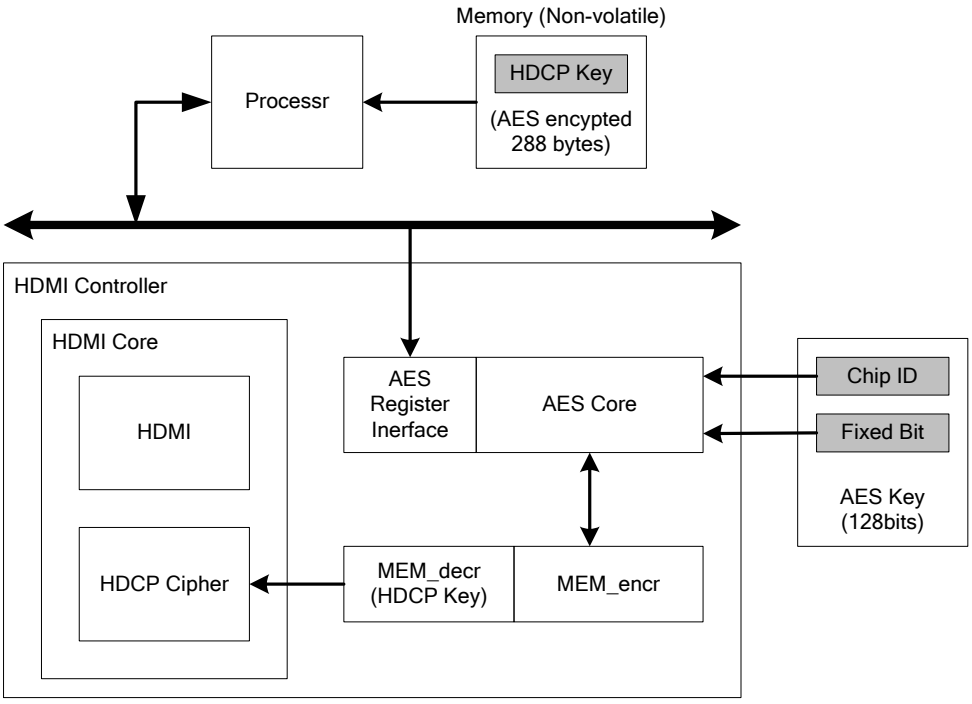


Figure 8.2 Block Diagram of HDCP Key Management

HDMI controller can decrypt AES-encrypted HDCP key. For HDMI with HDCP function, each device has unique HDCP key. In HDMI controller, HDCP key is stored outside of the HDMI controller at non-volatile memory. HDMI controller adopts AES encryption method to protect original HDCP key. Non-volatile memory does not have original raw-HDCP key, and without proper decryption the key in the non-volatile memory is useless.

In this situation, HDMI controller must have an additional AES key. This is needed to decrypt the AES encrypted HDCP key data. AES key can consist of variable chip id and fixed hardware value. However, this AES key combination can be different depending on the user configuration. Hacker may steal AES key and get an HDCP key. However, other device's HDCP key cannot be revealed because the Chip ID is different for each Chip. By this configuration HDCP key can be stored outside safely.

Core reads the AES-encrypted HDCP key once before starting HDMI and writes into mem_encr with the amount of decryption size desired. Then a command starts to decrypt encrypted data and the decrypted data are written into mem_decr. At that time the START flag goes down to notify the decryption is done.

AES core has 128-bit key that is embedded in hardwired fashion. It only supports 128-bit wide decryption. The first thing to do decryption is to load data into mem_encr. To load encryption data into encr_mem first sets AES_DATA_SIZE_H/L to decide how many bytes to decrypt and it should be 128-bit aligned because AES core is operating with 128bit data. And set AES_START to 1. If data is decrypted and full the mem_decr with decrypted data then the AES_START flag goes down to 0 to indicate the decryption and transfer are done. After checking that AES_START flag is 0, decrypted data can be used for HDCP key. HDCP cannot operate until the AES_START flag goes to 0.

8.3.2 Interface Protocol

8.3.2.1 Video Input Interface

Video input of HDMI controller is connected with TCC8900 LCD output. To use HDMI video output correctly, HDMI link setting, HDMI PHY setting and LCD setting should be consistent. LCD, HDMI link and HDMI PHY register setting values for some well-known video resolutions (for HDTV) are shown in 8.6 Examples - HDMI Register Setting.

8.3.2.2 Audio Input Interface

HDMI controller has two audio input ports (SPDIF Rx and I2S Rx). Each of audio inputs is connected with TCC8900 SPDIF Tx and TCC8900 I2S Tx.

Users can use SPDIF interface for two-channel Linear or Non-linear PCM Audio transmission and I2S interface for up-to eight channel Linear PCM or up-to six channel Super Audio CD audio transmission.

For I2S interface, users can specify the channel status block and the user bit information in registers, which does not inherently exist in I2S audio format.

8.3.2.3 HPD

HPD signal has two transactions: rising (plugged) and falling (unplugged) transition. Users can specify the stage number of the noise filter to reduce the possible glitches during transition.

8.3.2.4 CEC Interface

CEC is abbreviation of Consumer Electronics Control and is used for controlling devices connected to a one-wired “CEC bus”. The CEC output port is a bidirectional port, which should be pulled-up to 3.3V using external resistor and voltage supply. (i.e. an open-drain connection)

CEC devices send messages serially (MSB first) via CEC bus and the receiving device sends acknowledge bit by pulling the bus to low at ACK bit timing. For timing of each bit and structure of a message, refer to HDMI specification 1.3b, Supplement 1.(CEC specification)

8.3.2.5 AESKEY

HDMI controller has AES decryption function to support HDCP Key encryption in external HDCP Key configuration. Users can use the AESKEY to decrypt AES-encrypted HDCP Key. AESKEY value must be stable when AES decryption starts.

8.3.2.6 Interrupt Timing

HDMI controller generates level-triggered interrupts. Interrupts are masked in two levels: each submodule and controller. On the other hand, interrupt clear happens only in sub module for all interrupts except HPD. For HPD, every plug and unplug will generate, HPD plug interrupt and HPD unplug interrupt, respectively. When interrupt is masked on, interrupt asserts reflecting the current HPD plug status as shown in Figure 8.3 and Figure 8.4.

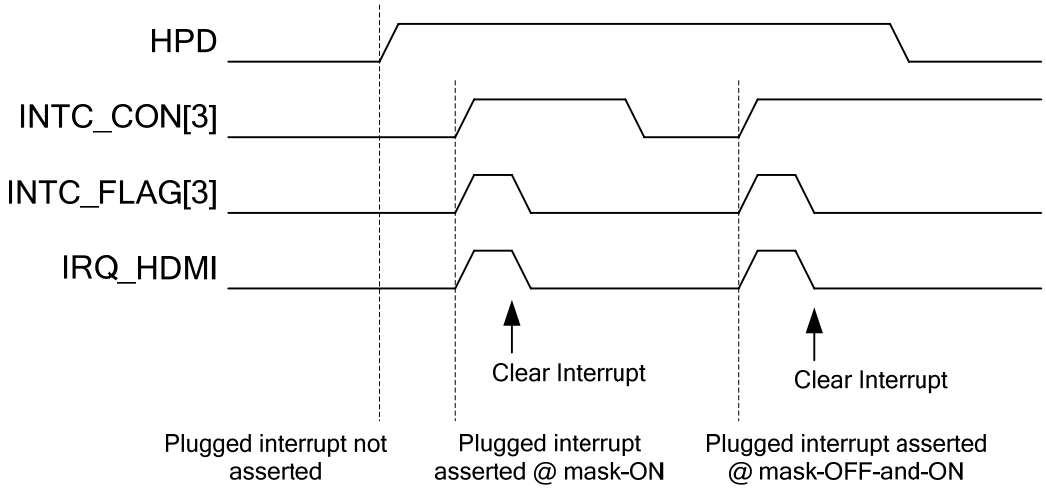


Figure 8.3 Timing Diagram for HPD Plug Interrupt

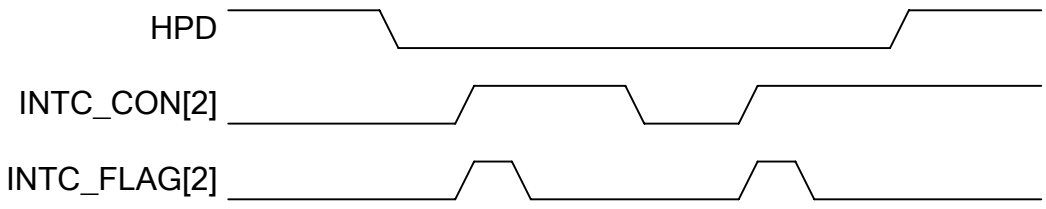


Figure 8.4 Timing Diagram for HPD Unplug Interrupt

8.4 HDMI Controller Register Description

Register Base	Offset	Description
CTRL_BASE	0xF0254000	Controller register base address
HDMI_CORE_BASE	0xF0255000	HDMI register base address
AES_BASE	0xF0256000	AES register base address
SPDIF_BASE	0xF0257000	SPDIF Receiver register base address
I2S_BASE	0xF0258000	I2S Receiver register base address
CEC_BASE	0xF0259000	CEC register base address

8.4.1 Control Registers

Name	Offset	Type	Description	Default
INTC_CON	0x0000	R/W	Interrupt Control Register	0x00
INTC_FLAG	0x0004	R/W	Interrupt Flag Register	0x00
AESKEY_VALID	0x0008	R	aeskey_valid Register	0x00
HPD	0x000C	R	HPD signal	0x00

INTC_CON **0xF0254000**

Name	Type	Description	Default
INTC_CON	R/W	Interrupt Control Register	0x00

Name	Bit	Description	Default
IntrPol	[7]	Interrupt Polarity 0 : Active high 1 : Active low	0
IntrEnGlobal	[6]	0 : All interrupts are disabled 1 : Interrupts are enabled or disabled by INTC_CON5:0]	0
IntrEnI2S	[5]	I2S interrupt enable 0 : Disabled 1 : Enabled	0
IntrEnCEC	[4]	CEC interrupt enable 0 : Disabled 1 : Enabled	0
IntrEnHPDplug	[3]	HPD plugged interrupt enable 0 : Disabled 1 : Enabled	0
IntrEnHPDunplug	[2]	HPD unplugged interrupt enable 0 : Disabled 1 : Enabled	0
IntrEnSPDIF	[1]	SPDIF interrupt enable 0 : Disabled 1 : Enabled	0
IntrEnHDCP	[0]	HDCP interrupt enable 0 : Disabled 1 : Enabled	0

INTC_FLAG **0xF0254004**

Name	Type	Description	Default
INTC_FLAG	R/W	Interrupt Flag Register	0x00

Name	Bit	Description	Default
-	[7:6]	Reserved	0b00
IntrI2S	[5]	I2S interrupt flag. (read only) 0 : not occurred 1 : interrupt occurred	0
IntrCEC	[4]	CEC interrupt flag. (read only) 0 : not occurred 1 : interrupt occurred	0
IntrHPDplug	[3]	HPD plugged interrupt flag. If it is written by 1, it is cleared. 0 : not occurred 1 : HPD plugged	0
IntrHPDunplug	[2]	HPD unplugged interrupt flag. If it is written by 1, it is cleared. 0 : not occurred 1 : HPD unplugged	0
IntrSPDIF	[1]	SPDIF interrupt flag. (read only) 0 : not occurred 1 : interrupt occurred	0
IntrHDCP	[0]	HDCP interrupt flag. (read only) 0 : not occurred 1 : interrupt occurred	0

AESKEY_VALID **0xF0254008**

Name	Type	Description	Default
AESKEY_VALID	R	i_aeskey_valid value	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
aeskey_valid	[0]	Reflects i_aeskey_valid signal value.*	0

* The use of aeskey_valid field depends on how aeskey is provided to HDMI 1.3 Tx subsystem. If program knows whether aeskey is valid via other way, this field may not need to be used. You can use this field in such a system where HDMI 1.3 Tx Subsystem is integrated with an AESKEY module that provides the AES key along with a valid signal tied up to aeskey_valid. Then S/W can read it to check AESKEY is valid.

HPD **0xF025400C**

Name	Type	Description	Default
HPD	R	HPD signal	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
HPD_Value	[0]	Value of HPD signal 0: Unplugged 1: Plugged	0

8.4.2 HDMI Core Registers

Name	Offset	Type	Description	Default
Control Registers				
HDMI_CON_0	0x0000	R/W	HDMI system control register 0	0x00
HDMI_CON_1	0x0004	R/W	HDMI system control register 1	0x00
HDMI_CON_2	0x0008	R/W	HDMI system control register 2	0x00
STATUS	0x0010	R/W	HDMI system status register	0x00
PHY_STATUS	0x0014	R	PHY status register	0x00
STATUS_EN	0x0020	R/W	HDMI system status enable register	0x00
HPD	0x0030	R/W	HPD control register	0x00
MODE_SEL	0x0040	R/W	HDMI/DVI mode selection	0x00
ENC_EN	0x0044	R/W	HDCP encryption enable register	0x00
Video Related Registers				
BLUE_SCREEN_0	0x0050	R/W	Pixel values for blue screen	0x00
BLUE_SCREEN_1	0x0054	R/W	Pixel values for blue screen 0x00	
BLUE_SCREEN_2	0x0058	R/W	Pixel values for blue screen 0x00	
HDMI_YMAX	0x0060	R/W	Maximum Y (or "R,G,B)" pixel value 0x00	
HDMI_YMIN	0x0064	R/W	Minimum Y (or "R,G,B)" pixel value 0x00	
HDMI_CMAX	0x0068	R/W	Maximum Cb/Cr pixel value	0x00
HDMI_CMIN	0x006C	R/W	Minimum Cb/Cr pixel value	0x00
H_BLANK_0	0x00A0	R/W	Horizontal blanking setting	0x00
H_BLANK_1	0x00A4	R/W	Horizontal blanking setting	0x00
V_BLANK_0	0x00B0	R/W	Vertical blanking setting	0x00
V_BLANK_1	0x00B4	R/W	Vertical blanking setting	0x00
V_BLANK_2	0x00B8	R/W	Vertical blanking setting	0x00
H_V_LINE_0	0x00C0	R/W	Horizontal line & vertical line setting	0x00
H_V_LINE_1	0x00C4	R/W	Horizontal line & vertical line setting	0x00
H_V_LINE_2	0x00C8	R/W	Horizontal line & vertical line setting	0x00
VSYNC_POL	0x00E4	R/W	Vertical sync polarity control register	0x00
INT_PRO_MODE	0x00E8	R/W	Interlace/Progressive control register	0x00
V_BLANK_F_0	0x0110	R/W	Vertical blanking setting for bottom field	0x00
V_BLANK_F_1	0x0114	R/W	Vertical blanking setting for bottom field	0x00
V_BLANK_F_2	0x0118	R/W	Vertical blanking setting for bottom field	0x00
H_SYNC_GEN_0	0x0120	R/W	Horizontal sync generation setting	0x00
H_SYNC_GEN_1	0x0124	R/W	Horizontal sync generation setting	0x00
H_SYNC_GEN_2	0x0128	R/W	Horizontal sync generation setting	0x00
V_SYNC_GEN1_0	0x0130	R/W	Vertical sync generation for top field or frame	0x01
V_SYNC_GEN1_1	0x0134	R/W	Vertical sync generation for top field or frame	0x10
V_SYNC_GEN1_2	0x0138	R/W	Vertical sync generation for top field or frame	0x00
V_SYNC_GEN2_0	0x0140	R/W	Vertical sync generation for bottom field - vertical position	0x01

V_SYNC_GEN2_1	0x0144	R/W	Vertical sync generation for bottom field - vertical position	0x10
V_SYNC_GEN2_2	0x0148	R/W	Vertical sync generation for bottom field - vertical position	0x00
V_SYNC_GEN3_0	0x0150	R/W	Vertical sync generation for bottom field - horizontal position	0x01
V_SYNC_GEN3_1	0x0154	R/W	Vertical sync generation for bottom field - horizontal position	0x10
V_SYNC_GEN3_2	0x0158	R/W	Vertical sync generation for bottom field - horizontal position	0x00
Audio Related Registers				
ASP_CON	0x0160	R/W	ASP packet control register	0x00
ASP_SP_FLAT	0x0164	R/W	ASP packet sp_flat bit control	0x00
ASP_CHCFG0	0x0170	R/W	ASP audio channel configuration	0x04
ASP_CHCFG1	0x0174	R/W	ASP audio channel configuration	0x1a
ASP_CHCFG2	0x0178	R/W	ASP audio channel configuration	0x2c
ASP_CHCFG3	0x017C	R/W	ASP audio channel configuration	0x3e
ACR_CON	0x0180	R/W	ACR packet control register	0x00
ACR_MCTS0	0x0184	R/W	Measured CTS value	0x01
ACR_MCTS1	0x0188	R/W	Measured CTS value	0x00
ACR_MCTS2	0x018C	R/W	Measured CTS value	0x00
ACR_CTS0	0x0190	R/W	CTS value for fixed CTS transmission mode.	0xe8
ACR_CTS1	0x0194	R/W	CTS value for fixed CTS transmission mode.	0x03
ACR_CTS2	0x0198	R/W	CTS value for fixed CTS transmission mode.	0x00
ACR_N0	0x01A0	R/W	N value for ACR packet	0xe8
ACR_N1	0x01A4	R/W	N value for ACR packet	0x03
ACR_N2	0x01A8	R/W	N value for ACR packet	0x00
ACR_LSB2	0x01B0	R/W	Alternate LSB for fixed CTS transmission mode	0x00
ACR_TXCNT	0x01B4	R/W	Number of ACR packet transmission per frame	0x1f
ACR_TXINTERVAL	0x01B8	R/W	Interval for ACR packet transmission	0x63
ACR_CTS_OFFSET	0x01BC	R/W	CTS offset for measured CTS mode	0x00
Packet Related Registers				
GCP_CON	0x01C0	R/W	ACR packet control register	0x00
GCP_BYTE1	0x01D0	R/W	GCP packet body	0x00
GCP_BYTE2	0x01D4	R/W	GCP packet body	0x00
GCP_BYTE3	0x01D8	R/W	GCP packet body	0x00
ACP_CON	0x01E0	R/W	ACP packet control register	0x00
ACP_TYPE	0x01F0	R/W	ACP packet header	0x00
ACP_DATA00~16	0x0200~ 0x00240	R/W	ACP packet body	0x00
ISRC_CON	0x0250	R/W	ACR packet control register	0x00
ISRC1_HEADER1	0x0264	R/W	ISCR1 packet header	0x00
ISRC1_DATA00~15	0x0270~ 0x02AC	R/W	ISRC1 packet body	0x00
ISRC2_DATA00~15	0x02B0~ 0x02EC	R/W	ISRC2 packet body	0x00
AVI_CON	0x0300	R/W	AVI packet control register	0x00
AVI_CHECK_SUM	0x0310	R/W	AVI packet checksum	0x00
AVI_BYTE01~13	0x0320~ 0x0350	R/W	AVI packet body	0x00
AUI_CON	0x0360	R/W	AUI packet control register	0x00
AUI_CHECK_SUM	0x0370	R/W	AUI packet checksum	0x00
AUI_BYTE1~5	0x0380~ 0x0390	R/W	AUI packet body	0x00
MPG_CON	0x03A0	R/W	ACR packet control register	0x00

MPG_CHECK_SUM	0x03B0	R/W	MPG packet checksum	0x00
MPG_BYTE1~5	0x03C0~ 0x03D0	R/W	MPG packet body	0x00
SPD_CON	0x0400	R/W	SPD packet control register	0x00
SPD_HEADER0	0x0410	R/W	SPD packet header	0x00
SPD_HEADER1	0x0414	R/W	SPD packet header	0x00
SPD_HEADER2	0x0418	R/W	SPD packet header	0x00
SPD_DATA00~27	0x0420~ 0x048C	R/W	SPD packet body	0x00
HDCP Related Registers				
HDCP_SHA1_00~19	0x0600~ 0x064C	R/W	SHA-1 value from repeater	0x00
HDCP_KSV_LIST_0~4	0x0650~ 0x0660	R/W	KSV list from repeater	0x00
HDCP_KSV_LIST_CON	0x0664	R/W	KSV list control	0x00
HDCP_SHA_RESULT	0x0670	R/W	SHA-1 checking result register	0x00
HDCP_CTRL1	0x0680	R/W	HDCP control register1	0x00
HDCP_CTRL2	0x0684	R/W	HDCP control register2	0x00
HDCP_CHECK_RESULT	0x0690	R/W	Ri and Pj value checking result	0x00
HDCP_BKSV_0~4	0x06A0~ 0x06B0	R/W	KSV of Rx	0x00
HDCP_AKSV_0~4	0x06C0~ 0x06D0	R/W	KSV of Tx	0x00
HDCP_An_0~7	0x06E0~ 0x06FC	R/W	An value	0x00
HDCP_BCAPS	0x0700	R/W	BCAPS from Rx	0x00
HDCP_BSTATUS_0	0x0710	R/W	BSTATUS from Rx	0x00
HDCP_BSTATUS_1	0x0714	R/W	BSTATUS from Rx	0x00
HDCP_Ri_0	0x0740	R/W	Ri value of Tx	0x00
HDCP_Ri_1	0x0744	R/W	Ri value of Tx	0x00
HDCP_I2C_INT	0x0780	R/W	I2C interrupt flag	0x00
HDCP_AN_INT	0x0790	R/W	An value ready interrupt flag	0x00
HDCP_WATCGDOG_INT	0x07A0	R/W	Wachdog interrupt flag	0x00
HDCP_Ri_INT	0x07B0	R/W	Ri value update interrupt flag	0x00
HDCP_Ri_Compare_0	0x07D0	R/W	HDCP Ri Interrupt Frame number index register 0	0x80
HDCP_Ri_Compare_1	0x07D4	R/W	HDCP Ri Interrupt Frame number index register 1	0x7f
HDCP_Frame_Count	0x07E0	R	Current value of the frame count index in the hardware	0x00
GAMUT_CON	0x0500	R/W	GAMUT packet control register	0x00
GAMUT_HEADER0	0x0504	R/W	GAMUT packet header	0x00
GAMUT_HEADER1	0x0508	R/W	GAMUT packet header	0x00
GAMUT_HEADER2	0x050C	R/W	GAMUT packet header	0x00
GAMUT_DATA00~27	0x0510~ 0x057C	R/W	GAMUT packet body	0x00
DC_CONTROL	0x05C0	R/W	Deep Color Control Register	0x00
VIDEO_PATTERN_GEN	0x05C4	R/W	Video Pattern Generation Register	0x00
HPD_GEN	0x05C8	R/W	HPD Duration value register	0x01

8.4.2.1 Control Registers

HDMI_CON_0 **0xF0255000**

Name	Type	Description	Default
HDMI_CON_0	R/W	HDMI system control register 0	0x00

Name	Bit	Description	Default
-	[7:6]	Reserved	0b00
Blue_Scr_En	[5]	Blue screen mode control. When set, the input video pixels are discarded and BLUESCREEN register values are transmitted for all video data period. 0: Disable 1: Enable	0
Encoding_Option	[4]	10-bit TMDS encoding bit order option 0: bit order reverse among the 10-bit encoding (to be set to 1 when connecting to the TMDS PHY 1.3) 1: bit order as it is	0
YCBCR422_Sel	[3]	Video Input mode control. 0: 4:4:4 mode 1: 4:2:2 12 bit YCbCr When 8-bit mode, the 12-bit inputs are rounded up to generate 8-bit outputs.	0
Asp_E	[2]	Audio sample packet generation control. This bit is only valid when SYSTEM_EN is set. 0: discard audio sample 1: When the audio sample is received, the audio sample packet is generated.	0
Power_Down	[1]	TMDS PHY power down mode. When it’ s set to 0, data could not be transferred to a receiver. 0: normal operation mode 1: power down	0
System_En	[0]	HDMI system enable. 0: No op. 1: HDMI enable	0

HDMI_CON_1 **0xF0255004**

Name	Type	Description	Default
HDMI_CON_1	R/W	HDMI system control register 1	0x00

Name	Bit	Description	Default
-	[7]	Reserved	0
Pxl_Lmt_Ctrl	[6:5]	Pixel value limitation control 0b00 : By-pass (Do not limit the pixel value) 0b01 : RGB mode All channel's video input pixels are limited according to YMAX and YMIN register values. 0b10: YCbCr mode The value of I_VIDEO_G is limited according to YMAX and YMIN. The values of I_VIDEO_B and I_VIDEO_R are limited according to CMAX and CMIN. 0b11 : Reserved	0b00
-	[4:2]	Reserved	0b000
Pxl_Rep_Ratio	[1:0]	Pixel repetition ratio 0b00 : no pixel repetition 0b01 : 2 times repetition 0b10 : 3 times repetition 0b11 : 4 times repetition * Use the resulting video mode setting for pixel repetition. e.g. For 720x480p (pixel repetition = 1) Use 1440x480p video mode	0b00

HDMI_CON_2 **0xF0255008**

Name	Type	Description	Default
HDMI_CON_2	R/W	HDMI system control register 2	0x00

Name	Bit	Description	Default
-	[7:6]	Reserved	0b00
Vid_Period_En	[5]	Video preamble control. 0 : Video Preamble is applied. (HDMI mode) 1 : Video Preamble is not applied (DVI mode)	0
-	[4:2]	Reserved	0b000
Dvi_ Band_En	[1]	In DIV mode, the leading guard band is not used. 0 : Guard band is applied (HDMI mode) 1 : Guard band is not applied (DVI mode)	0
-	[0]	Reserved	0

STATUS **0xF0255010**

Name	Type	Description	Default
STATUS	R/W	HDMI system status flag register	0x00

Name	Bit	Description	Default
Authen_Ack	[7]	When hdcp is authenticated, it occurs. This bit keeps the authentication signal constantly. It's not cleared at all. It's just one delayed signal of authen_ack signal from hdcp block. This bit is not an interrupt source. Read Only bit 0:not authenticated 1: authenticated	0
Aud_Fifo_Ovf	[6]	When audio FIFO is overflowed, this bit will be set. Once it is set, it should be cleared by host. 0: not full 1: full	0
-	[5]	Reserved	0
Update_Ri_Int	[4]	Ri Interrupt status bit If it is written by 1, it is cleared. 0: not occurred 1: interrupt occurred	0
-	[3]	Reserved	0
An_Write_Int	[2]	Indicates that {An} random value is ready. If it is written by 1, it is cleared. 0: not occurred 1: interrupt occurred	0
Watchdog_Int	[1]	Indicates that 2nd part of HDCP authentication protocol is initiated and CPU should set a watchdog timer to check 5 sec interval. If it is written by 1, it is cleared. 0: not occurred 1: interrupt occurred	0
I2c_Init_Int	[0]	Indicates that 1st part of HDCP authentication protocol can start. If it is written by 1, it is cleared. 0: not occurred 1: interrupt occurred	0

PHY_STATUS **0xF0255014**

Name	Type	Description	Default
PHY_STATUS	R/W	HDMI Phy Status	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
Phy_Ready	[0]	Indicates that the PHY is ready to receive the HDMI signals from link 0: Not Ready 1: Ready	0

STATUS_EN **0xF0255020**

Name	Type	Description	Default
PHY_STATUS	R/W	HDMI interrupt status enable register	0x00

Name	Bit	Description	Default
-	[7]	Reserved	0
Aud_Fido_Ovf_Ee	[6]	Audio buffer overflow interrupt enable When it is set to ‘1’, interrupt assertion is written on the STATUS registers. 0 : Disable 1 : Enable	0
-	[5]	Reserved	0
Update_Ri_Int_En	[4]	UPDATE_RI_INT interrupt enable. 0 : Disable 1 : Enable	0
-	[3]	Reserved	0
An_Write_Int_En	[2]	AN_WRITE_INT interrupt enable. 0 : Disable 1 : Enable	0
Watchdog_Int_En	[1]	WATCHDOG_INT interrupt enable. 0 : Disable 1 : Enable	0
I2c_Int_En	[0]	I2C_INT interrupt enable. 0 : Disable 1 : Enable	0

HPD **0xF0255030**

Name	Type	Description	Default
HPD	R/W	HPD Determines whether HPD is used from external port, or internally generated signal.	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
Sw_Hpd	[1]	When HPD_SEL bit is set, this SW_HPD signal is used for HPD (HDMI/DVI cable plugging). When set to low during HDMI transmission, status machines in HDCP core is reset. Note that other HDCP register values are not influenced. 0 : Low (unplugged) 1 : High (plugged)	0
Hpd_Sel	[0]	When clear, the I_HPD signal from the I/O port is used for HPD. When set, the SW_HPD signal is used for HPD. 0 : HPD signal 1 : SW_HPD internal HPD signal	0

MODE_SEL **0xF0255040**

Name	Type	Description	Default
MODE_SEL	R/W	HDMI/DVI mode selection for HDCP. After mode decision, one of two bits has to be set.	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
Hdmi_Mode	[1]	Select a mode. 0 : Disable 1 : Enable	0
Dvi_Mode	[0]	Select a mode. 0 : Disable 1 : Enable	0

ENC_EN **0xF0255044**

Name	Type	Description	Default
ENC_EN	R/W	HDCP encryption enable register.	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
Hdcp_Enc_En	[0]	When set, HDCP encryption is applied. Before setting this bit, the HDCP authentication process has to be accomplished. 0 : Encryption disable 1 : Enable	0

8.4.2.2 Video Related Registers

BLUESCREEN_0/1/2 **0xF0255050/54/58**

Name	Type	Description	Default
BLUESCREEN_0/1/2	R/W	Pixel values when blue screen mode is enable. (When BLUE_SCR_EN is set in HDMI_CON_0 register).	0x00

Name	Bit	Description	Default
BLUESCREEN_0	[7:0]	Channel 0 color setting. (Cb or B)	0x0
BLUESCREEN_1	[7:0]	Channel 1 color setting. (Y or G)	0x0
BLUESCREEN_2	[7:0]	Channel 2 color setting. (Cr or R)	0x0

HDMI_YMAX / HDMI_YMIN / HDMI_CMAX / HDMI_CMIN **0xF0255060/64/68/6C**

Name	Type	Description	Default
HDMI_YMAX HDMI_YMIN HDMI_CMAX HDMI_CMIN	R/W	Specify the range of pixel value	0xEB 0x10 0xF0 0x10

Name	Bit	Description	Default
HDMI_YMAX HDMI_YMIN HDMI_CMAX HDMI_CMIN	[7:0] [7:0] [7:0] [7:0]	These registers are used according to PX_LMT_CTRL bits in HDMI_CON_1 register. For RGB mode if (i_video_x > HDMI_YMAX x 16) output = HDMI_YMAX x 16 else if (i_video_x < HDMI_YMIN x 16) output = HDMI_YMIN x 16 else output = i_video_x For YCbCr mode, the Y input is dealt with the same as above. For Cb and Cr values, if (i_video_x > HDMI_CMAX x 16) output = HDMI_CMAX x 16 else if (i_video_x < HDMI_CMIN x 16) output = HDMI_CMIN x 16 else output = i_video_x Note) Value 16 in each line compensates the difference	0xEB 0x10 0xF0 0x10

H_BLANK_0/1 **0xF02550A0/A4**

Name	Type	Description	Default
H_BLANK_0 H_BLANK_1	R/W	H_BLANK[7:0] H_BLANK[15:8]	0x00

Name	Bit	Description	Default
-	[15:10]	Reserved	0b000000
H_BLANK	[9:0]	Clock Cycles of horizontal Blanking Size. Refer to the Reference CEA-861D	0x000

60Hz	720x480i	720x480p	1440x480p	1280x720p	1920x1080i	1920x1080p
H_BLANK	276(114h)	138(8Ah)	276(114h)	370(172h)	280(118h)	280(118h)
50Hz	720x576i	720x576p	1440x576p	1280x720p	1920x1080i	1920x1080p
H_BLANK	288(120h)	144(90h)	288(120h)	700(2bch)	720(2d0h)	720(2d0h)

Note) 1440x480p and 1440x576p are the pixel doubling format of 720x480p and 720x576p

V_BLANK_0/1/2 **0xF02550B0/B4/B8**

Name	Type	Description	Default
V_BLANK_0 V_BLANK_1 V_BLANK_2	R/W	V_BLANK[7:0] V_BLANK[15:8] V_BLANK[23:16]	0x00

Name	Bit	Description	Default
-	[23:22]	Reserved	0x0
V1_BLANK	[21:11]	Vertical Blanking Line Size. Front Part. Refer to the Reference CEA-861D	0x000
V2_BLANK	[10:0]	V1_BLANK+Active Lines. End Part. This value is the same as V_LINE value for progressive mode. For interlace mode, use the reference value in the table. Refer to the Reference CEA-861D	0x000

60Hz	720x480i	720x480p	1440x480p	1280x720p	1920x1080i	1920x1080p
V2_BLANK	262(d)	525(d)	525(d)	750(d)	562(d)	1125(d)
V1_BLANK	22(d)	45(d)	45(d)	30(d)	22(d)	45(d)
V_BLANK	b106(h)	16a0d(h)	16a0d(h)	f2ee(h)	b232(h)	1_6c65(h)
50Hz	720x576i	720x576p	1440x576p	1280x720p	1920x1080i	1920x1080p
V2_BLANK	312(d)	625(d)	625(d)	750(d)	562(d)	1125(d)
V1_BLANK	24(d)	49(d)	49(d)	30(d)	22(d)	45(d)
V_BLANK	c138(h)	18a71(h)	18a71(h)	F2ee(h)	B232(h)	1_6c65(h)

H_V_LINE_0/1/2 **0xF02550C0/C4/C8**

Name	Type	Description	Default
H_V_LINE_0 H_V_LINE_1 H_V_LINE_2	R/W	H_V_LINE[7:0] H_V_LINE[15:8] H_V_LINE[23:16]	0x00

Name	Bit	Description	Default
H_LINE	[23:12]	Horizontal Line Length. Refer to the Reference CEA-861D	0x000
V_LINE	[11:0]	Vertical Line Length. Refer to the Reference CEA-861D	0x000

60Hz	720x480i	720x480p	1440x480p	1280x720p	1920x1080i	1920x1080p
V_LINE	525(d)	525(d)	525(d)	750(d)	1125(d)	1125(d)
H_LINE	1716(d)	858(d)	1716(d)	1650(d)	2200(d)	2200(d)
H_V_LINE	6b420d(h)	35a20d(h)	6b420d(h)	6722ee(h)	898465(h)	898465(h)
50Hz	720x576i	720x576p	1440x576p	1280x720p	1920x1080i	1920x1080p
V_LINE	625(d)	625(d)	625(d)	750(d)	1125(d)	1125(d)
H_LINE	1728(d)	864(d)	1728(d)	1980(d)	2640(d)	2640(d)
H_V_LINE	6c0271(h)	360271(h)	6C0271(h)	7bc2ee(h)	a50465(h)	a50465(h)

VSYNC_POL **0xF02550E4**

Name	Type	Description	Default
VSYNC_POL	R/W	Vertical sync polarity control register. Refer to the Reference CEA-861D	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0x00
V_Sync_Pol_Sel	[0]	Start point detection polarity selection bit. 720p or 1080i’ s sync shapes are different from 480p,480i, and 576p’ s. They are inverted shapes. 0 : active high 1 : active low	0

60Hz	720x480i	720x480p	1440x480p	1280x720p	1920x1080i	1920x1080p
VSYNC_POL	1	1	1	0	0	0
50Hz	720x576i	720x576p	1440x576p	1280x720p	1920x1080i	1920x1080p
VSYNC_POL	1	1	1	0	0	0

INT_PRO_MODE **0xF02550E8**

Name	Type	Description	Default
INT_PRO_MODE	R/W	Interlaced or Progressive Mode Selection	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
INT_PRO_MODE	[0]	Interlaced or Progressive Mode Selection. Refer to the Reference CEA-861D 0: progressive 1: interlaced.	0

V_BLANK_F_0/1/2 **0xF0255110/114/118**

Name	Type	Description	Default
V_BLANK_F_0 V_BLANK_F_1 V_BLANK_F_2	R/W	V_BLANK_F[7:0] V_BLANK_F[15:8] V_BLANK_F[23:16]	0x00

Name	Bit	Description	Default
-	[23:22]	Reserved	0x0
V_BOT_END	[21:11]	In the interlace mode, v_blank length of even field and odd field is different. This register specifies the end position of bottom field’s active region. Refer to the Reference CEA-861D	0x000
V_BOT_ST	[10:0]	The start position of bottom field’s active region. This value is the same as V_LINE value for Interlace mode. For progressive mode, This value is not used. Refer to the Reference CEA-861D	0x000

*The above register only affects the interlace mode.

60Hz	720x480i	720x480p	1440x480p	1280x720p	1920x1080i	1920x1080p
V_BOT_ST	285(d)	Don’t care	Don’t care	Don’t care	585(d)	Don’t care
V_BOT_END	525(d)				1125(d)	
V_BLANK_F	10691d(h)				232a49(h)	
50Hz	720x576i	720x576p	1440x576p	1280x720p	1920x1080i	1920x1080p
V_BOT_ST	337(d)	Don’t care	Don’t care	Don’t care	585(d)	Don’t care
V_BOT_END	625(d)				1125(d)	
V_BLANK_F	138951(h)				232a49(h)	

H_SYNC_GEN_0/1/2 **0xF0255120/124/128**

Name	Type	Description	Default
H_SYNC_GEN_0 H_SYNC_GEN_1 H_SYNC_GEN_2	R/W	H_SYNC_GEN[7:0] H_SYNC_GEN[15:8] H_SYNC_GEN[23:16]	0x00

Name	Bit	Description	Default
-	[23:21]	Reserved	0x0
Hsync_Pol	[20]	Set this bit for inverting the generated signal to meet the modes. In 720p and 1080i modes don't need to invert the signal. Others need to be inverted. Refer to the Reference CEA-861D 0 : active high 1 : active low	0
Hsync_Edn	[19:10]	Set the end point of H sync. Refer to the Reference CEA-861D	0x000
Hsync_Start	[9:0]	Set the start point of H sync. Refer to the Reference CEA-861D	0x000

60Hz	720x480i	720x480p	1440x480p	1280x720p	1920x1080i	1920x1080p
HSYNC_START	36(d)	14(d)	30(d)	108(d)	86(d)	86(d)
HSYNC_END	160(d)	76(d)	154(d)	148(d)	130(d)	130(d)
HSYNC_POL	1	1	1	0	0	0
H_SYNC_GEN	128024(h)	11300e(h)	12681e(h)	2506c(h)	20856(h)	20856(h)
50Hz	720x576i	720x576p	1440x576p	1280x720p	1920x1080i	1920x1080p
HSYNC_START	22(d)	10(d)	22	438(d)	526(d)	526(d)
HSYNC_END	148(d)	74(d)	150	478(d)	570(d)	570(d)
HSYNC_POL	1	1	1	0	0	0
H_SYNC_GEN	125016(h)	11280a(h)	125816(h)	779b6(h)	8ea0e(h)	8ea0e(h)

V_SYNC_GEN1_0/1/2 **0xF0255130/134/138**

Name	Type	Description	Default
V_SYNC_GEN1_0	R/W	V_SYNC_GEN1[7:0]	0x01
V_SYNC_GEN1_1		V_SYNC_GEN1[15:8]	0x10
V_SYNC_GEN1_2		V_SYNC_GEN1[23:16]	0x00

Name	Bit	Description	Default
Vsync_T_St	[23:12]	Top field (or frame) V sync start line number. Refer to the Reference CEA-861D	0x001
Vsync_T_End	[11:0]	Top field (or frame) V sync end line number. Refer to the Reference CEA-861D	0x001

60Hz	720x480i	720x480p	1440x480p	1280x720p	1920x1080i	1920x1080p
VSYNC_T_END	7(d)	15(d)	15(d)	10(d)	7(d)	9(d)
VSYNC_T_ST	4(d)	9(d)	9(d)	5(d)	2(d)	4(d)
V_SYNC_GEN1	4007(h)	900f(h)	900f(h)	500a(h)	2007(h)	4009(h)
50Hz	720x576i	720x576p	1440x576p	1280x720p	1920x1080i	1920x1080p
VSYNC_T_END	5(d)	10(d)	10(d)	10(d)	7(d)	9(d)
VSYNC_T_ST	2(d)	5(d)	5(d)	5(d)	2(d)	4(d)
V_SYNC_GEN1	2005(h)	500a(h)	500a(h)	500a(h)	2007(h)	4009(h)

V_SYNC_GEN2_0/1/2 **0xF0255140/144/148**

Name	Type	Description	Default
V_SYNC_GEN2_0 V_SYNC_GEN2_1 V_SYNC_GEN2_2	R/W	V_SYNC_GEN2[7:0] V_SYNC_GEN2[15:8] V_SYNC_GEN2[23:16]	0x01 0x10 0x00

Name	Bit	Description	Default
Vsync_B_St	[23:12]	Bottom field V sync start line number. Refer to the Reference CEA-861D	0x001
Vsync_B_End	[11:0]	Bottom field V sync end line number. Refer to the Reference CEA-861D	0x001

60Hz	720x480i	720x480p	1440x480p	1280x720p	1920x1080i	1920x1080p
VSYNC_B_END	269(d)	Don't Care	Don't Care	Don't Care	569(d)	Don't Care
VSYNC_B_ST	266(d)				564(d)	
V_SYNC_GEN2	10a10d(h)				234239(h)	
50Hz	720x576i	720x576p	1440x576p	1280x720p	1920x1080i	1920x1080p
VSYNC_B_END	317(d)	Don't Care	Don't Care	Don't Care	569(d)	Don't Care
VSYNC_B_ST	314(d)				564(d)	
V_SYNC_GEN2	13a13d(h)				234239(h)	

V_SYNC_GEN3_0/1/2 **0xF0255150/154/158**

Name	Type	Description	Default
V_SYNC_GEN3_0	R/W	V_SYNC_GEN3[7:0]	0x01
V_SYNC_GEN3_1		V_SYNC_GEN3[15:8]	0x10
V_SYNC_GEN3_2		V_SYNC_GEN3[23:16]	0x00

Name	Bit	Description	Default
Vsync_H_Pos_St	[23:12]	Bottom field V sync start transition point. Refer to the Reference CEA-861D	0x001
Vsync_H_Pos_End	[11:0]	Bottom field V sync end transition point. Refer to the Reference CEA-861D	0x001

60Hz	720x480i	720x480p	1440x480p	1280x720p	1920x1080i	1920x1080p
VSYNC_H_POS_ST	896(d)	Don't Care	Don't Care	Don't Care	1188(d)	Don't Care
VSYNC_H_POS_END	896(d)				1188(d)	
V_SYNC_GEN3	380380(h)				4A44A4(h)	
50Hz	720x576i	720x576p	1440x576p	1280x720p	1920x1080i	1920x1080p
VSYNC_H_POS_ST	888(d)	Don't Care	Don't Care	Don't Care	1848(d)	Don't Care
VSYNC_H_POS_END	888(d)				1848(d)	
V_SYNC_GEN3	378378(h)				738738(h)	

8.4.2.3 Audio related Packet Registers

ASP_CON **0xF0255160**

Name	Type	Description	Default
ASP_CON	R/W	Audio Sample Packet generation control register	0x00

Name	Bit	Description	Default
DST_Double	[7]	DST double	0
Aud_Type	[6:5]	Packet type instead of audio type 00 : Audio Sample Packet 01 : One-bit audio packet 10 : HBR packet 11 : DST packet	0b00
Aud_Mode	[4]	Two channel or multi-channel mode selection This bit will be also used for layout bit in ASP header. 0 : 2 channel mode 1 : Multi channel mode. Set this bit to transmit HBR packets.	0
SP_Pre	[3:0]	Control sub-packet usage for multi channel mode only. When two-channel mode, this register value is not used.	0b0000

ASP_SP_FLAT **0xF0255164**

Name	Type	Description	Default
ASP_SP_FLAT	R/W	sp_flat or sample_invalid value for ASP header	0x00

Name	Bit	Description	Default
-	[7:4]	Reserved	0b0000
SP_Flat	[3:0]	The sp_flat/sample_invalid value in the ASP header. Refer to the HDMI specification v1.3 (5.3.4 and 5.3.9)	0x0

ASP_CHCFG0/1/2/3 **0xF0255170/174/178/17C**

Name	Type	Description	Default
ASP_CHCFG0	R/W	ASP_CHCFG[7:0]	0x04
ASP_CHCFG1		ASP_CHCFG[15:8]	0x1A
ASP_CHCFG2		ASP_CHCFG[23:16]	0x2C
ASP_CHCFG3		ASP_CHCFG[31:24]	0x3E

Name	Bit	Description	Default
-	[31:30]	Reserved	0b00
Spk3R_Sel	[29:27]	Audio channel Selection for subpacket 3 right channel data in multi channel mode. 000 : i_pcm0L is used for sub packet 0 Left channel 001 : i_pcm0R is used for sub packet 0 Left channel 010 : i_pcm1L is used for sub packet 0 Left channel 011 : i_pcm1R is used for sub packet 0 Left channel 100 : i_pcm2L is used for sub packet 0 Left channel 101 : i_pcm2R is used for sub packet 0 Left channel 110 : i_pcm3L is used for sub packet 0 Left channel 111 : i_pcm3R is used for sub packet 0 Left channel	0b111
Spk3L_Sel	[26:24]	Audio channel Selection for subpacket 3 left channel data in multi channel mode. The meaning is the same as SPK3R_SEL.	0b110
-	[23:22]	Reserved	0b00
Spk2R_Sel	[21:19]	Audio channel Selection for subpacket 2 right channel data in multi channel mode. The meaning is the same as SPK3R_SEL.	0b101
Spk2L_Sel	[18:16]	Audio channel Selection for subpacket 2 left channel data in multi channel mode. The meaning is the same as SPK3R_SEL.	0b100
-	[15:14]	Reserved	0b00
SPK1R_SEL	[13:11]	Audio channel Selection for subpacket 1 right channel data in multi channel mode. The meaning is the same as SPK3R_SEL.	0b011
Spk1L_Sel	[10:8]	Audio channel Selection for subpacket 1 left channel data in multi channel mode. The meaning is the same as SPK3R_SEL.	0b010
-	[7:6]	Reserved	0b00
Spk0R_Sel	[5:3]	Audio channel Selection for subpacket 0 right channel data in multi channel mode. The meaning is the same as SPK3R_SEL.	0b001
Spk0L_Sel	[2:0]	Audio channel Selection for subpacket 0 right channel data in multi channel mode. The meaning is the same as SPK3R_SEL.	0b000

ACR_CON **0xF0255180**

Name	Type	Description	Default
ACR_CON	R/W	ACR packet transmission control register	0x00

Name	Bit	Description	Default
-	[7:5]	Reserved	0b000
Alt_Cts_Rate	[4:3]	In some audio format, the CTS value can be changed alternately. CTS value 1 = ACR_CTS[19:0] CTS value 2 = {ARC_CTS[19:8], ACR_LSB2} These two values can be transmitted alternately at the ratio of this register setting. 00 : always CTS value 1 01 : 1:1 (CTS value 1 : CTS value2) 10 : 2:1 (CTS value 1 : CTS value2) 11 : 3:1 (CTS value 1 : CTS value2) Measured CTS mode, this value is not used.	0b00
ACR_Tx_Mode	[2:0]	000: Do not Tx (Transfer) the ACR packet. 001: Tx once – Transmit ACR packet once when anytime available after this value is set. After transmitting, these bits are reset to all zero. 010 : Tx ACR_TXCNT times during every VBI period 011 : Tx by counting i_clk_vid for a given CTS value in the ACR_CTS0~2 registers. 100 : Measured CTS mode. Make ACR packet with CTS value by counting TMDS clock for Fs x 128 / N duration. In this case, the 7 LSBs of N value (ACR_N register) should be all zero.	0b000

ACR_MCTS0/1/2 **0xF0255184/188/18C**

Name	Type	Description	Default
ACR_MCTS0 ACR_MCTS1 ACR_MCTS2	R/W	ACR_MCTS[7:0] ACR_MCTS[15:8] ACR_MCTS[23:16] ACR packet transmission control register	0x01 0x00 0x00

Name	Bit	Description	Default
-	[23:20]	Reserved	0x0
ACR_MCTS	[19:0]	This value is the TMDS clock cycles for N[19:7] number of audio sample inputs. It is only valid when measured CTS mode is set on ACR_CON register.	0x00001

ACR_CTS0/1/2 **0xF0255190/194/198**

Name	Type	Description	Default
ACR_CTS0 ACR_CTS1 ACR_CTS2	R/W	ACR_CTS[7:0] ACR_CTS[15:8] ACR_CTS[23:16]	0xE8 0x03 0x00

Name	Bit	Description	Default
-	[23:20]	Reserved	0x0
ACR_CTS	[19:0]	CTS value for transmission mode other than ‘measured CTS’ mode.	0x0003E8

ACR_N0/1/2 **0xF02551A0/1A4/1A8**

Name	Type	Description	Default
ACR_N0 ACR_N1 ACR_N2	R/W	ACR_N[7:0] ACR_N[15:8] ACR_N[23:16]	0xE8 0x03 0x00

Name	Bit	Description	Default
-	[23:20]	Reserved	0
ACR_N	[19:0]	The N value in ACR packet	0x003E8

ACR_LSB2 **0xF02551B0**

Name	Type	Description	Default
ACR_LSB2	R/W	Alternate CTS least significant byte	0x00

Name	Bit	Description	Default
ACR_LSB2	[7:0]	Alternate CTS least significant byte See ALT_CTS_RATE in ACR_CON register.	0x00

ACR_TXCNT **0xF02551B4**

Name	Type	Description	Default
ACR_TXCNT	R/W	ACR packet transmission count register	0x1F

Name	Bit	Description	Default
-	[7:5]	Reserved	0
ACR_TXCNT	[4:0]	When ACR_TX_MODE is '10', the ACR packet will be transmitted ACR_TXCNT + 1' times per every VBI period. ALT_CTS_RATE is also applied. This register is only valid when ACR_TX_MODE is '10'.	0x1F

ACR_TXINTERVAL **0xF02551B8**

Name	Type	Description	Default
ACR_TXINTERVAL	R/W	ACR packet TX interval register	0x63

Name	Bit	Description	Default
ACR_TX_INTERVAL	[7:0]	When ACR_TX_MODE is '10', the ACR packet will be transmitted ACR_TXCNT times during VBI. This register specifies the number of cycles between each ACR packets. This register is used to avoid continuous transmission more than 18 packets within single DI band. This register is only valid when ACR_TX_MODE is '10'.	0x63

ACR_CTS_OFFSET **0xF02551BC**

Name	Type	Description	Default
ACR_CTS_OFFSET	R/W	The offset for measured CTS value.	0x00

Name	Bit	Description	Default
ACR_CTS_OFFSET	[7:0]	When ‘measured CTS mode’ is used, the CTS value will be measured by counting the TMDS clock for a given duration. This value is added to measured CTS value. It is 8 bit signed integer, so subtraction is possible.	0x00

GCP_CON **0xF02551C0**

Name	Type	Description	Default
GCP_CON	R/W	GCP packet transmission control register	0x00

Name	Bit	Description	Default
-	[7:3]	Reserved	0b00000
ENABLE_1st_VSYNC	[3]	For Interlace mode, Enable this bit to transfer the GCP packet on the 1st VSYNC in a frame 0: Do not transfer GCP packet 1: Transfer GCP packet * For Progressive mode, GCP packet is transferred regardless of this bit. i.e. GCP packet in progressive mode is transferred every vsync if GCP_CON is 0b1x	0b0
ENABLE_2nd_VSYNC	[2]	For Interlace mode, Enable this bit to transfer the GCP packet on the 2nd VSYNC in a frame 0: Do not transfer GCP packet 1: Transfer GCP packet	0b1
GCP_CON	[1:0]	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync. GCP packet will be transmitted within 384 cycles after active vsync.	0b00

GCP_BYTE1 **0xF02551D0**

Name	Type	Description	Default
GCP_BYTE1	R/W	GCP data byte	0x00

Name	Bit	Description	Default
GCP_BYTE1	[7:0]	GCP packet's first data byte. It shall be either 0x10 (Clear AVMUTE) or 0x01 (Set AVMUTE). Refer to Table 5-17 of HDMI specification.	0x00

GCP_BYTE2 **0xF02551D4**

Name	Type	Description	Default
GCP_BYTE2	R/W	GCP data byte	0x00

Name	Bit	Description	Default
PP	[7:4]	PP (Packing Phase), Read Only	0x0
CD	[3:0]	CD (Color Depth)	0x0

GCP_BYTE3 **0xF02551D8**

Name	Type	Description	Default
GCP_BYTE3	R/W	GCP data byte	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
GCP_BYTE3	[0]	Default State	0

8.4.2.4 ACP Packet Registers

ACP_CON **0xF02551E0**

Name	Type	Description	Default
ACP_CON	R/W	ACP packet transmission control register	0x00

Name	Bit	Description	Default
ACP_FR_RATE	[7:3]	Transmit ACP packet once per every ACP_FR_RATE+1 frames (or fields).	0b00000
-	[2]	Reserved	0
ACP_TX_CON	[1:0]	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync with ACP_FR_RATE	0b00

ACP_TYPE **0xF02551F0**

Name	Type	Description	Default
ACP_TYPE	R/W	ACP packet header HB1 register	0x00

Name	Bit	Description	Default
ACP_TYPE	[7:0]	ACP packet header. (HB1 of ACP packet header) See Table 5-18 in HDMI v1.3 specification	0x00

ACP_DATA00~16 **0xF0255200~0xF0255400**

Name	Type	Description	Default
ACP_DATA00~16	R/W	ACP packet data registers. (17 bytes)	0x00

Name	Bit	Description	Default
ACP_DATA00~16	[7:0]	ACP packet body data. (PB0~PB16 of ACP packet body) See 9.3 in HDMI v1.3 specification	0x00

8.4.2.5 ISRC1/2 packet registers

ISRC_CON **0xF0255250**

Name	Type	Description	Default
ISRC_CON	R/W	ISRC packet transmission control register	0x00

Name	Bit	Description	Default
ISRC_FR_RATE	[7:3]	Transmit ISRC1 (with ISRC2 or not) packet once per every ISRC_FR_RATE+1 frames (or fields).	0b00000
ISRC2_EN	[2]	Transmit ISRC2 packet with ISRC1 packet	0
ISRC_TX_CON	[1:0]	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync with ISRC_FR_RATE	0b00

ISRC1_HEADER1 **0xF0255264**

Name	Type	Description	Default
ISRC1_HEADER1	R/W	ISRC packet header	0x00

Name	Bit	Description	Default
ISRC_Cont	[7]	See table 5-20 in HDMI v1.3specification	0
ISRC_Valid	[6]	See table 5-20 in HDMI v1.3 specification	0
-	[5:3]	Reserved	0b000
ISRC status	[2:0]	See table 5-20 in HDMI v1.3 specification	0b000

ISRC1_DATA 00~15 **0xF0255270~0xF02552AC**

Name	Type	Description	Default
ISRC1_DATA 00~15	R/W	ISRC1 packet data. Total 16 bytes	0x00

Name	Bit	Description	Default
ISRC1_DATA00~15	[7:0]	ISRC2 packet body data. (PB0~15 of ISRC2 packet body) See Table 5-21 in HDMI v1.3 specification.	0x00

SRC2_DATA 00~15 **0xF02552B0~0xF02552EC**

Name	Type	Description	Default
SRC2_DATA 00~15	R/W	ISRC2 packet data. Total 16 bytes	0x00

Name	Bit	Description	Default
ISRC2_DATA00~15	[7:0]	ISRC2 packet body data. (PB0~15 of ISRC2 packet body) See Table 5-21 in HDMI v1.3 specification.	0x00

8.4.2.6 AVI InfoFrame registers

AVI_CON **0xF0255300**

Name	Type	Description	Default
AVI_CON	R/W	AVI Infoframe packet transmission control register	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
AVI_TX_CON	[1:0]	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync	0b00

AVI_CHECK_SUM **0xF0255310**

Name	Type	Description	Default
AVI_CHECK_SUM	R/W	AVI InfoFrame checksum data	0x00

Name	Bit	Description	Default
AVI_CHECK_SUM	[7:0]	AVI InfoFrame checksum byte. (PB0 byte of AVI packet body)	0x00

AVI_DATA01 ~ AVI_DATA13 **0xF0255320~0xF025350**

Name	Type	Description	Default
AVI_DATA01~ AVI_DATA13	R/W	AVI Infoframe packet data	0x00

Name	Bit	Description	Default
AVI_DATA01~ AVI_DATA13	[7:0]	AVI Infoframe packet data registers. (PB1~PB13 bytes of AVI packet body)	0x00

8.4.2.7 Audio InfoFrame registers

AUI_CON **0xF0255360**

Name	Type	Description	Default
AUI_CON	R/W	Audio Infoframe (AUI) packet transmission control register	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
AUI_TX_CON	[1:0]	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync	0b00

AUI_CHECK_SUM **0xF0255370**

Name	Type	Description	Default
AUI_CHECK_SUM	R/W	AUI packet checksum byte	0x00

Name	Bit	Description	Default
AUI_CHECK_SUM	[7:0]	AUI Infoframe checksum data. (PB0 byte of AUI packet body)	0x00

AUI_DATA1 ~ AUI_DATA5 **0xF0255380~0xF0255390**

Name	Type	Description	Default
AUI_DATA1~5	R/W	AUI Infoframe packet data	0x00

Name	Bit	Description	Default
AUI_DATA1~5	[7:0]	AUI Infoframe packet body. (PB1~PB5 bytes of AUI packet body)	0x00

8.4.2.8 MPEG Source InfoFrame

MPG_CON **0xF02553A0**

Name	Type	Description	Default
MPG_CON	R/W	MPG infoframe transmission control register	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
MPG_TX_CON	[1:0]	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync	0b00

MPG_CHECK_SUM **0xF02553B0**

Name	Type	Description	Default
MPG_CHECK_SUM	R/W	MPG Infoframe checksum byte	0x00

Name	Bit	Description	Default
MPG_CHECK_SUM	[7:0]	MPG infoframe checksum register (PB0 byte of MPG packet body)	0x00

MPG_DATA1 ~ MPG_DATA5 **0xF02553C0~0xF02553D0**

Name	Type	Description	Default
MPG_DATA1~5	R/W	MPG Infoframe packet body	0x00

Name	Bit	Description	Default
MPGI_DATA1~5	[7:0]	MPG Infoframe packet data. (PB1~PB5 bytes of MPG packet body)	0x00

8.4.2.9 Source Product Descriptor InfoFrame (or general packet generation)

These registers can be used for Source Product Descriptor (SPD) packet transmission. Furthermore, they consist of full configurable header and packet body registers (3 bytes header register and 28 bytes packet body registers) so that they can be used for transmission of any type of packet.

SPD_CON **0xF0255400**

Name	Type	Description	Default
SPD_CON	R/W	SPD packet transmission control register	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
SPD_TX_CON	[1:0]	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync	0b00

SPD_HEADER0/1/2 **0xF0255410/414/418**

Name	Type	Description	Default
SPD_HEADER0 SPD_HEADER1 SPD_HEADER2	R/W	SPD packet header. (SPD_HEADER)	0x00 0x00 0x00

Name	Bit	Description	Default
SPD_HEADER0	[7:0]	HB0 byte of SPD packet header	0x00
SPD_HEADER1	[7:0]	HB1 byte of SPD packet header	0x00
SPD_HEADER2	[7:0]	HB2 byte of SPD packet header	0x00

SPD_DATA00~27 **0xF0255440~0xF025548C**

Name	Type	Description	Default
SPD_DATA00~ SPD_DATA27	R/W	SPD packet data	0x00

Name	Bit	Description	Default
SPD_DATA00~ SPD_DATA27	[7:0]	SPD packet data registers. (PB0~PB27 bytes)	0x00

8.4.2.10 HDCP Register Description

HDCP_SHA1_00~19 **0xF0255600~0xF025564C**

Name	Type	Description	Default
HDCP_SHA1_00	R/W	HDCP_SHA1[7:0]	0x00
HDCP_SHA1_01		HDCP_SHA1[15:8]	
HDCP_SHA1_02		HDCP_SHA1[23:16]	
HDCP_SHA1_03		HDCP_SHA1[31:24]	
HDCP_SHA1_04		HDCP_SHA1[39:0]	
HDCP_SHA1_05		HDCP_SHA1[47:0]	
HDCP_SHA1_06		HDCP_SHA1[55:0]	
HDCP_SHA1_07		HDCP_SHA1[63:0]	
HDCP_SHA1_08		HDCP_SHA1[71:0]	
HDCP_SHA1_09		HDCP_SHA1[79:0]	
HDCP_SHA1_10		HDCP_SHA1[87:0]	
HDCP_SHA1_11		HDCP_SHA1[95:0]	
HDCP_SHA1_12		HDCP_SHA1[103:0]	
HDCP_SHA1_13		HDCP_SHA1[111:0]	
HDCP_SHA1_14		HDCP_SHA1[119:0]	
HDCP_SHA1_15		HDCP_SHA1[127:0]	
HDCP_SHA1_16		HDCP_SHA1[135:0]	
HDCP_SHA1_17		HDCP_SHA1[143:0]	
HDCP_SHA1_18		HDCP_SHA1[151:0]	
HDCP_SHA1_19		HDCP_SHA1[159:0]	

Name	Bit	Description	Default
HDCP_SHA1	[159:0]	160-bit HDCP repeater's SHA-1 value. Least significant byte first. These registers are readable but they are not modified by HDCP H/W.	0

HDCP_SHA_RESULT **0xF0255670**

Name	Type	Description	Default
HDCP_SHA_RESULT	R/W	The result of comparing the SHA-1 values.	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b00000
Hdcp_Sha_Valid_Ready	[1]	Indicates that the SHA comparison has been done by the HW. Must be cleared by SW by writing 0 0: Not ready 1: Ready	0
Hdcp_Sha_Valid	[0]	Indicates that the SHA-1 comparison succeeds. Must be cleared by SW by writing 0 0: Valid 1: Not valid	0

HDCP_KSV_LIST_0~4 **0xF0255650~0xF0255660**

Name	Type	Description	Default
HDCP_KSV_LIST_0	R/W	HDCP_KSV_LIST[7:0]	0x00
HDCP_KSV_LIST_1		HDCP_KSV_LIST[15:8]	
HDCP_KSV_LIST_2		HDCP_KSV_LIST[23:16]	
HDCP_KSV_LIST_3		HDCP_KSV_LIST[31:24]	
HDCP_KSV_LIST_4		HDCP_KSV_LIST[39:32]	

Name	Bit	Description	Default
HDCP_KSV_LIST	[39:0]	Little endian addressing. One KSV value of the HDCP repeater's	0

		KSV list. These registers are readable.	

HDCP_KSV_LIST_CON **0xF0255664**

Name	Type	Description	Default
HDCP_KSV_LIST_CON	R/W	HDCP KSV list control register	0x00

Name	Bit	Description	Default
-	[7:4]	Reserved	0b0000
Hdcp_Ksv_Write_Done	[3]	After writing KSV data into HDCP_KSV_LIST_X registers and then writing the value “1” to this register, HW processes the written KSV value and clears this bit to “0” 0: Not yet written 1: Written	0
Hdcp_Ksv_List_Empty	[2]	If the number of KSV list is zero, set this value to make SHA-1 module to start to calculate without KSV list. 0: Not empty 1: Empty	0
Hdcp_Ksv_End	[1]	It is used to indicate that current KSV value in HDCP_KSV_LIST_X registers is the last one. 0: Not End 1: End	0
Hdcp_Ksv_Read	[0]	After writing KSV data into HDCP_KSV_LIST_X registers HDCP SHA-1 module keeps that KSV value into internal buffer and set this flag into ‘1’ to notify that it has been read. After checking that it is set to ‘1’, SW clears to ‘0’ at the same time when writing the HDCP_KSV_WRITE_DONE bit for next KSV list value. 0: Not Read 1: Read	0

HDCP_CTRL1 **0xF0255680**

Name	Type	Description	Default
HDCP_CTRL1	R/W	HDCP control register 1	0x00

Name	Bit	Description	Default
-	[7:4]	Reserved	0b0000
-	[3]	Reserved	0
Timeout	[2]	Set when Rx is repeater and its KSV list is not ready until 5 sec waiting. 0: Not timeout 1: Timeout(KSV Ready bit in the HDCP_BCAPS register is not high until 5 sec) and re-start the 1st authentication. Refer to the Figure 2-6 in the HDCP 1.3 specification	0
CP_Desired	[1]	HDCP enable 0: Not Desired 1: Desired	0
-	[0]	Reserved	0

HDCP_CTRL2 **0xF0255684**

Name	Type	Description	Default

HDCP_CTRL2	R/W	HDCP control register 2	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
Revocation_Set	[0]	KSV list is on the revocation list & Fail the 2nd authentication. 0: Revocation Not set 1: Revocation Set	0

HDCP_CHECK_RESULT 0xF0255690

Name	Type	Description	Default
HDCP_CHECK_RESULT	R/W	After SW checks Ri of Rx and that of Tx are same, the result has to be written to this register.	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
Ri_Match_Result	[1:0]	Write the result of comparison between Ri of Rx and Tx as the following values. (Ri : Tx, Ri' : Rx) Must be cleared by SW after setting 10 or 11 before next Ri Interrupt occurs. 0x : don't care 10 : Ri ≠ Ri' 11 : Ri = Ri'	0b00

HDCP_BKSV0~4 0xF02556A0~0xF02556B0

Name	Type	Description	Default
HDCP_BKSV0	R/W	HDCP_BKSV[7:0]	0x00
HDCP_BKSV1		HDCP_BKSV[15:8]	
HDCP_BKSV2		HDCP_BKSV[23:16]	
HDCP_BKSV3		HDCP_BKSV[31:24]	
HDCP_BKSV4		HDCP_BKSV[39:32]	

Name	Bit	Description	Default
HDCP_BKSV	[39:0]	Key selection vector (KSV) value from receiver.	0

HDCP_AKSV0~4 0xF02556C0~0xF02556D0

Name	Type	Description	Default
HDCP_AKSV0	R/W	HDCP_AKSV[7:0]	0x00
HDCP_AKSV1		HDCP_AKSV[15:8]	
HDCP_AKSV2		HDCP_AKSV[23:16]	
HDCP_AKSV3		HDCP_AKSV[31:24]	
HDCP_AKSV4		HDCP_AKSV[39:32]	

Name	Bit	Description	Default
HDCP_AKSV	[39:0]	KSV value of transmitter	0

HDCP_An_0~7 **0xF02556E0~0xF02556FC**

Name	Type	Description	Default
HDCP_An_0	R	HDCP_An[7:0]	0x00
HDCP_An_1		HDCP_An[15:8]	
HDCP_An_2		HDCP_An[23:16]	
HDCP_An_3		HDCP_An[31:24]	
HDCP_An_4		HDCP_An[39:32]	
HDCP_An_5		HDCP_An[47:40]	
HDCP_An_6		HDCP_An[55:48]	
HDCP_An_7		HDCP_An[63:56]	

Name	Bit	Description	Default
HDCP_An	[63:0]	64-bit Random number generated by Tx (An)	0

HDCP_BCAPS **0xF0255700**

Name	Type	Description	Default
HDCP_BCAPS	R/W	BCAPS information from Rx. This value is the data read from Rx.	0x00

Name	Bit	Description	Default
-	[7]	Reserved	0
Repeater	[6]	The receiver supports downstream connections 0: Not Repeater 1: Repeater	0
Ready	[5]	KSV FIFO, SHA-1 calculation ready 0: Not Ready 1: Ready	0
Fast	[4]	The receiver devices supports 400KHz transfer 0: Not Fast 1: Fast	0
Reserved	[3:2]	Must be 0’s	0
1.1_Features	[1]	Supports EESS, Advance cipher, Enhanced link verification 0: un-set 1: set	0
Fast_Reauthentication	[0]	ALL HDMI receiver should be capable of reauthentication 0: un-set 1: set	0

HDCP_BSTATUS_0/1 **0xF0255710/714**

Name	Type	Description	Default
HDCP_BSTATUS_0 HDCP_BSTATUS_1	R/W	HDCP_BSTATUS[7:0] HDCP_BSTATUS[15:8] HDCP Bstatus value from Rx. This is the data read from Rx.	0x00 0x00

Name	Bit	Description	Default
-	[15:13]	Reserved	0b000
Hdmi_Mode	[12]	HDMI mode indication. If set, HDCP is in HDMI mode.	0
Max_Cascade_Exceeded	[11]	Topology error	0
Depth	[10:8]	Cascade depth	0b000
Max_Devs_Exceeded	[7]	Topology error indicator 0: No Error 1: Error	0
Device_Count	[6:0]	Total number of attached downstream devices	0

HDCP_Ri_0/1 **0xF0255740/744**

Name	Type	Description	Default
HDCP_Ri_0 HDCP_Ri_1	R	HDCP_Ri[7:0] HDCP_Ri[15:8]	0x00 0x00

Name	Bit	Description	Default
HDCP_Ri	[15:0]	HDCP Ri value of the transmitter.	0x0000

HDCP_I2C_INT **0xF0255780**

Name	Type	Description	Default
HDCP_I2C_INT	R/W	HDCP I2C interrupt status	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
HDCP_I2C_INT	[0]	HDCP I2C Interrupt status. Active high. It indicates the start of I2C transaction when it is set. After active, it should be cleared by S/W by writing 0. 0: Not Occurred 1: Occurred	0

HDCP_AN_INT **0xF0255790**

Name	Type	Description	Default
HDCP_AN_INT	R/W	HDCP An interrupt status	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
HDCP_AN_INT	[0]	HDCP An Interrupt status. Active high. If An value is available, it is set. After active, it should be cleared by S/W by writing 0. 0: Not Occurred 1: Occurred	0

HDCP_WATCHDOG_INT **0xF02557A0**

Name	Type	Description	Default
HDCP_WATCHDOG_INT	R/W	HDCP Watchdog interrupt status	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
HDCP_WATCHDOG_INT	[0]	HDCP Watchdog Interrupt status. Active high. If Repeater bit value is set after 1st authentication success, it is set. After active, it should be cleared by S/W by writing 0. 0: Not Occurred 1: Occurred	0

HDCP_RI_INT **0xF02557B0**

Name	Type	Description	Default
HDCP_RI_INT	R/W	HDCP RI interrupt status	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
HDCP_RI_INT	[0]	If Ri value is updated internally (at every 128 video frames), It is set to high. After set, it should be cleared by S/W by writing 0. 0: Not Occurred 1: Occurred	0

8.4.2.11 Ri Check Registers

HDCP_Ri_Compare_0 **0xF02557D0**

Name	Type	Description	Default
HDCP_Ri_Compare_0	R/W	HDCP Ri Interrupt Frame number index register 0	0x00

Name	Bit	Description	Default
Enable	[7]	Enable the interrupt for this frame number index	1
Frame Number index	[6:0]	When the Frame count reaches this "Frame number index", an Ri Link integrity check interrupt occurs	0b0000000

HDCP_Ri_Compare_1 **0xF02557D4**

Name	Type	Description	Default
HDCP_Ri_Compare_1	R/W	HDCP Ri Interrupt Frame number index register 1	0x7F

Name	Bit	Description	Default
Enable	[7]	Enable the interrupt for this frame number index	1
Frame Number index	[6:0]	When the Frame count reaches this "Frame number index", an Ri Link integrity check interrupt occurs	0b1111111

HDCP_Frame_Count **0xF02557E0**

Name	Type	Description	Default
HDCP_Frame_Count	R	Current value of the frame count index in the hardware	0x00

Name	Bit	Description	Default
-	[7]	Reserved	0
Frame Count	[6:0]	Current value of the frame count index in the hardware	0b0000000

8.4.2.12 Gamut Metadata Packet Registers

GAMUT_CON **0xF0255500**

Name	Type	Description	Default
GAMUT_CON	R/W	Gamut Metadata packet transmission control register	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b00000
GAMUT_CON	[1:0]	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync	0b00

GAMUT_HEADER0 **0xF0255504**

Name	Type	Description	Default
GAMUT_HEADER0	R/W	Gamut metadata packet header	0x00

Name	Bit	Description	Default
HB0	[7:0]	HB0 value in the table 5-30 in HDMI 1.3 specification	0b00000

GAMUT_HEADER1 **0xF0255508**

Name	Type	Description	Default
GAMUT_HEADER1	R/W	Gamut metadata packet header	0x00

Name	Bit	Description	Default
Next_Field	[7]	Set to indicate that the GBD carried in this packet will be effective on the next video field.	0
GBD_profile	[6:4]	Transmission profile number (We only support profile 0)	0b000
Affected_Gamut_Seq_Num	[3:0]	Indicates which video fields are relevant for this metadata	0b0000

GAMUT_HEADER2 **0xF025550C**

Name	Type	Description	Default
GAMUT_HEADER2	R/W	Gamut metadata packet header	0x00

Name	Bit	Description	Default
No_Crnt_GBD	[7]	Set to indicate that there is no gamut metadata available for the currently transmitted video	0
Reserved	[6]	Reserved	0
Packet_Seq	[5:4]	Indicates whether this packet is the only, the first, an intermediate or the last packet in a Gamut metadata packet sequence	0b00
Current_Gamut_Seq_Num	[3:0]	Indicates the gamut number of the currently transmitted video stream	0b0000

GAMUT_METADATA0~27 **0xF0255510~0xF025557C**

Name	Type	Description	Default
GAMUT_METADATA	R/W	Gamut Metadata packet body data	0x00

Name	Bit	Description	Default
GAMUT_METADATA0~ GAMUT_METADATA27	[7:0]	Gamut Metadata Packet body for P0 transmission profile	0x00

8.4.2.13 Video Mode Registers

DC_CONTROL **0xF02555C0**

Name	Type	Description	Default
DC_CONTROL	R/W	Deep Color related Control Register	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
Deep Color Mode	[1:0]	00: 8 bits /pixel 01:10bits /pixel 10:12 bits / pixel 11: Not Used	0b00

VIDEO_PATTERN_GEN **0xF02555C4**

Name	Type	Description	Default
VIDEO_PATTERN_GEN	R/W	Video Pattern Generation Register	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
Ext_Video_En	[1]	0: Ext Off 1: Ext En	0
Video Pattern Enable	[0]	0: Disable 1: Use Internally generated video pattern	0

HPD_GEN **0xF02555C8**

Name	Type	Description	Default
HPD_GEN	R/W	HW HPD Duration for stable HPD input	0x01

Name	Bit	Description	Default
HPD_Duration	[7:0]	Num of cycles for determining stable HPD input Internal count = TMDS clock * HPD_Duration * 16 (cycles) Default value = 0x1 (16 TMDS clock cycles)	0x01

INTERNAL_VIDEO **0xF0255900~0xF0255A7C**

Name	Type	Description	Default
Internal_Video	R/W	Internal Video Generation	0x00

Name	Bit	Description	Default
R_00_00_0 ~ B_11_11_1	[7:0]	-Generation of Internal Video RGB Data -Can be set every 256 line and 256 pixel count values - 12bit R, G, B values are separated on 2 registers. ex) r_00_00 => R pixel data on 0~256th line & 0~256th pixel in a frame. r_00_00_0 => r_00_00[11:4], r_00_00_1 => r_00_00[3:0]	0x00

8.4.2.14 AES Registers

Name	Offset	Type	Description	Default
AES_START	0x0000	R/W	AES_START	0x00
AES_DATA_SIZE_L	0x0020	R/W	AES_DATA_SIZE_L	0x20
AES_DATA_SIZE_H	0x0024	R/W	AES_DATA_SIZE_H	0x01
AES_DATA	0x0040	W	AES_DATA	-

AES_START **0xF0256000**

Name	Type	Description	Default
AES_START	R/W	AES_START	0x00

Name	Bit	Description	Default
Reserved	[7:1]	Reserved	0b0000000
AES_Start	[0]	AES Start signal If specified amount of data is decrypted and written in memory, then AES start signal goes to 0. 0 : AES does not decrypt data. (AES decryption completed) 1 : AES starts to decrypt data from memory.	0

AES_DATA_SIZE_L/H **0xF0256020/24**

Name	Type	Description	Default
AES_DATA_SIZE_L AES_DATA_SIZE_H	R/W	AES_DATA_SIZE_L AES_DATA_SIZE_H	0x20 0x01

Name	Bit	Description	Default
AES_Data_Size_L AES_Data_Size_H	[7:0] [7:0]	AES data size (in bytes) to decrypt If the data size is not the multiple of 128 bits, zeros should be padded 128bit align Maximum number is 120h(internal memory size limits the maximum data size Default value : 288 bytes	0x20 0x01

AES_DATA **0xF0256040**

Name	Type	Description	Default
AES_DATA	W	AES_DATA	0x00

Name	Bit	Description	Default
AES_Data	[7:0]	Write buffer to store AES-encrypted data in memory before starting decryption Memory address is automatically increased. Zeros should be padded for 128 bit align.	0x00

8.4.3 SPDIF Registers

Name	Offset	Type	Description	Default
SPDIFIN_CLK_CTRL	0x0000	R/W	SPDIFIN Clock Control Register	0x02
SPDIFIN_OP_CTRL	0x0004	R/W	SPDIFIN Operation Control Register 1	0x00
SPDIFIN_IRQ_MASK	0x0008	R/W	SPDIFIN Interrupt Request Mask Register	0x00
SPDIFIN_IRQ_STATUS	0x000C	R/W	SPDIFIN Interrupt Request Status Register	0x00
SPDIFIN_CONFIG_1	0x0010	R/W	SPDIFIN Configuration Register 1	0x02
SPDIFIN_CONFIG_2	0x0014	R/W	SPDIFIN Configuration Register 2	0x00
-	0x0018	-	Reserved	-
-	0x001C	-	Reserved	-
SPDIFIN_USER_VALUE_1	0x0020	R	SPDIFIN User Value Register 1	0x00
SPDIFIN_USER_VALUE_2	0x0024	R	SPDIFIN User Value Register 2	0x00
SPDIFIN_USER_VALUE_3	0x0028	R	SPDIFIN User Value Register 3	0x00
SPDIFIN_USER_VALUE_4	0x002C	R	SPDIFIN User Value Register 4	0x00
SPDIFIN_CH_STATUS_0_1	0x0030	R	SPDIFIN Channel Status Register 0-1	0x00
SPDIFIN_CH_STATUS_0_2	0x0034	R	SPDIFIN Channel Status Register 0-2	0x00
SPDIFIN_CH_STATUS_0_3	0x0038	R	SPDIFIN Channel Status Register 0-3	0x00
SPDIFIN_CH_STATUS_0_4	0x003C	R	SPDIFIN Channel Status Register 0-4	0x00
SPDIFIN_CH_STATUS_1	0x0040	R	SPDIFIN Channel Status Register 1	0x00
-	0x0044	-	Reserved	-
SPDIFIN_FRAME_PERIOD_1	0x0048	R	SPDIFIN Frame Period Register 1	0x00
SPDIFIN_FRAME_PERIOD_2	0x004C	R	SPDIFIN Frame Period Register 2	0x00
SPDIFIN_Pc_INFO_1	0x0050	R	SPDIFIN Pc Info Register 1	0x00
SPDIFIN_Pc_INFO_2	0x0054	R	SPDIFIN Pc Info Register 2	0x00
SPDIFIN_Pd_INFO_1	0x0058	R	SPDIFIN Pd Info Register 1	0x00
SPDIFIN_Pd_INFO_2	0x005C	R	SPDIFIN Pd Info Register 2	0x00
SPDIFIN_DATA_BUF_0_1	0x0060	R	SPDIFIN Data Buffer Register 0_1	0x00
SPDIFIN_DATA_BUF_0_2	0x0064	R	SPDIFIN Data Buffer Register 0_2	0x00
SPDIFIN_DATA_BUF_0_3	0x0068	R	SPDIFIN Data Buffer Register 0_3	0x00
SPDIFIN_USER_BUF_0	0x006C	R	SPDIFIN User Buffer Register 0	0x00
SPDIFIN_DATA_BUF_1_1	0x0070	R	SPDIFIN Data Buffer Register 1_1	0x00
SPDIFIN_DATA_BUF_1_2	0x0074	R	SPDIFIN Data Buffer Register 1_2	0x00
SPDIFIN_DATA_BUF_1_3	0x0078	R	SPDIFIN Data Buffer Register 1_3	0x00
SPDIFIN_USER_BUF_1	0x007C	R	SPDIFIN User Buffer Register 1	0x00

8.4.3.1 Control Registers

SPDIFIN_CLK_CTRL **0xF0257000**

Name	Type	Description	Default
SPDIFIN_CLK_CTRL	R/W	I2S Clock Control Register	0x02

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
ready_clk_down	[1]	0: clock is enabled 1: ready for disabling clock (default)	1
power_on	[0]	0: clock will be disabled (default) 1: clock will be activated If this bit is reset, SPDIFIN stops checking the input signal just before next 'subframe' of SPDIF signal format and wait the 'acknowledge' signal from HDMI for unresolved previous 'request' toward HDMI. Then asserts 'ready_clk_down' as HIGH. To initialize internal states, you have to assert S/W reset, i.e. SPDIFIN_OP_CTRL. op_ctrl=00b right after activating clock again.	0

The spdif_clk may be gated by an external clock gating module for low-power or etc. Disabling the clock must not cause stalling HDMI data transfer. Therefore the system processor requests disabling of the clock by setting the power_on register to low and the module acknowledge this request by setting the ready_clk_down register to high after a current transaction on the I2C bus and HDMI is finished. The module must not commence a new bus transaction until the system processor sets the power_on register to high again.

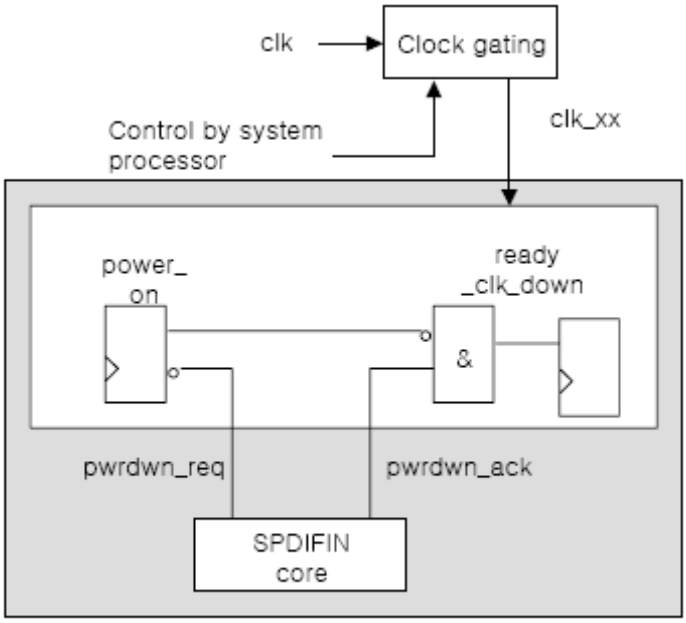


Figure 18 Structure of power down circuit

The system processor may only switch off clk_xx when the ready_clk_down bit is one and the power_on bit is zero.

The system processor can switch on the clock at any time. After having switched on clk_xx the system processor has to set the power_on bit to 1, which forces the ready_clk_down bit to zero.

After reset the clock of SPDIFIN is switched off, therefore the power_on bit is zero and the ready_clk_down bit is set to one.

SPDIFIN_OP_CTRL **0xF0257004**

Name	Type	Description	Default
SPDIFIN_OP_CTRL	R/W	SPDIFIN Operation Control Register	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
op_ctrl	[1:0]	00b : software reset 01b : status checking mode (run) 11b : status checking + HDMI operation mode (run with HDMI) Others : undefined, do not use 00b : During a software reset, all state machines are set to the idle or init state and all internal registers are set to their initial values. Interrupt status registers are cleared, all other registers that are writable by the system processor keep their values. 01b : This command should be asserted after SPDIFIN_CLK_CTRL.power_on is set. SPDIFIN starts clock recovery. When recovery is done, SPDIFIN begins detecting preambles of SPDIF signal format and stream data header, abnormal time signal input, abnormal signal input, and also reports these status via interrupts in SPDIFIN_IRQ_STATUS. 11b : “01b” case operations + checking internal buffer overflow + write received data, which can be either audio sample word of PCM or payload of stream. Data will be transferred via HDMI. - You should assert ‘op_ctrl’=11b after SPDIFIN_IRQ_STATUS. ch_status_recovered_ir was asserted at least once for linear PCM data. Or you should assert ‘op_ctrl’=11b after SPDIFIN_IRQ_STATUS.stream_header_detected_ir was asserted at least once for non-linear PCM stream data.	0

SPDIFIN_IRQ_MASK **0xF0257008**

Name	Type	Description	Default
SPDIFIN_IRQ_MASK	R/W	SPDIFIN Interrupt Request Mask Register	0x00

Name	Bit	Description	Default
buf_overflow_ir_en	[7]	mask bit for Interrupt 7	0
-	[6]	Reserved	0
-	[5]	Reserved	0
stream_header_detected_ir_en	[4]	mask bit for Interrupt 4	0
stream_header_not_detected_ir_en	[3]	mask bit for Interrupt 3	0
wrong_preamble_ir_en	[2]	mask bit for Interrupt 2	0
ch_status_recovered_ir_en	[1]	mask bit for Interrupt 1	0
wrong_signal_ir_en	[0]	mask bit for Interrupt 0	0
For every bit: ‘0’: interrupt generation is disabled. ‘1’: interrupt generation is enabled.			

SPDIFIN_IRQ_STATUS **0xF025700C**

Name	Type	Description	Default
SPDIFIN_IRQ_STATUS	R/W	SPDIFIN Interrupt Request Status Register	0x00

Name	Bit	Description	Default
buf_overflow_ir	[7]	0: no interrupt 1: internal buffer overflow SPDIFIN internal buffer(s) (SPDIFIN_DATA_BUF_x) was overflowed because HDMI did not transfer the data in the buffer(s) to memory in time. - This interrupt will be asserted only if SPDIFIN_OP_CTRL.op_ctrl was set as “011”. - If user does not handle this interrupt, SPDIFIN will overwrite next subframe data to the internal data buffer (SPDIFIN_DATA_BUF_x) and continue data transfer via HDMI.	0
-	[6]	Reserved	0
-	[5]	Reserved	0
stream_header_detected_ir	[4]	0: no interrupt 1: stream data header (Pa~Pd) detected - This interrupt will be asserted when SPDIFIN_OP_CTRL.op_ctrl=001b or 011b. - Cases for interrupt case1 : Initially after power_on case2 : Next stream header at right time when receiving stream data with SPDIFIN_CONFIG.data_type set as ‘stream mode’. case3: Initially detected stream header when receiving stream data with SPDIFIN_CONFIG.data_type set as ‘PCM mode’.	0
stream_header_not_detected_ir	[3]	0: no interrupt 1: stream data header not detected for 4096 repetition time - This interrupt will be asserted when SPDIFIN_OP_CTRL.op_ctrl=001b or 011b. - Cases for interrupt case1 : Initially after power_on case2 : SPDIFIN was receiving stream but could not find next stream header for 4096 repetition time since previous stream header case3 : Could not find stream header for 4096 repetition time since previous reset of repetition time counter after previous interrupt of ‘stream_header_not_detected_ir’.	0
wrong_preamble_ir	[2]	0: no interrupt 1: preamble was detected but there is a problem with detected time - This interrupt will be asserted when SPDIFIN_OP_CTRL.op_ctrl=001b or 011b. - Meaningless until ch_status_recovered_ir is asserted initially after SPDIFIN_OP_CTRL.op_ctrl=01b - Cases for interrupt case1: preamble was detected in the middle of a subframe audio sample word time case2: next preamble was not detected at exact time after a subframe duration case3: it was time for preamble B(or M or W) to be detected but other preamble was detected at that time	0
ch_status_recovered_ir	[1]	0: no interrupt 1: recovered channel status Detected preamble of 2 consecutive B-preamble thus recovered 192 bit wide channel status. - Only supports consumer mode, so just 36bits will be reconstructed. If a user wants to see the channel status bits through SPDIFIN_CH_STATUS_x, you’d better read two	0

		consecutive ‘ch_status_recovered_ir’ and read that register each time; if these two channel status value are same, you can rely on that value.	
wrong_signal_ir	[0]	0: no interrupt 1: clock recovery fail Can not recover clock from input because of tolerable range violation(unlock) or because of no signal from outside or because of non-biphase in non-preamble duration - Meaningless until ch_status_recovered_ir is asserted initially after SPDIFIN_OP_CTRL.op_ctrl=01b	0
For every bit the following holds: Reading returns interrupt request status. Writing ‘0’ has no effect. Writing ‘1’ clears the interrupt request.			
1) Detection of stream header Wait for matching of Pa, Pb; 0xF872, 0x4E1F respectively Wait for the repetition time (From decoded Pc value or from user-set Pc in SPDIFIN_USER_VALUE.repetition_time_manual according to SPDIFIN_CONFIG. PcPd_value_mode) Check for matching of Pa, Pb on right time.			

SPDIFIN_CONFIG_1 **0xF0257010**

Name	Type	Description	Default
SPDIFIN_CONFIG_1	R/W	SPDIFIN Configuration Register 1	0x02

Name	Bit	Description	Default
-	[7]	Reserved	0
abnormal_Pd_ir	[6]	0: filtering with 3 consecutive samples 1: filtering with 2 consecutive samples Noise filtering is done for over-sampled SPDIF input signal. This operation will be done as follows. If 'noise_filter_samples' is 0, 3 consecutive over-sampled signal will be regarded as a high or low only when those 3 samples are all high or low respectively. If 1 or 2 samples are low or high respectively for 3 over-sample duration, that noise filtered signal will keep previous value. If 'noise_filter_samples' is 1, 2 consecutive over-sampled signal will be regarded as a high or low only when those 2 samples are all high or low respectively. This setting can be used for reduced over-sampling ratio; recommended over-sampling ratio is 10 (see also 'clk_divisor')	0
-	[5]	Reserved (Must be '0')	0
PcPd_value_mode	[4]	0: automatically set 1: manually set If 0 for automatic setting, Pc, Pd values will be chosen by value of Pc, Pd from decoded stream header reported as in SPDIFIN_Px_INFO. If user sets this register, the receiver will use SPDIFIN_USER_VALUE[31:16], SPDIFIN_USER_VALUE[15:4] value as Pc, Pd respectively instead of decoded data from stream header as reported in SPDIFIN_Px_INFO. (cf) Burst payload length, whether it is automatically set or manually set, will affect the data size to be written in memory via HDMI by dumping the full sub-frame for the last bit for burst payload length. For example, if burst payload length is 257bit, i.e. (16subframe * 16bit + 1bit), then HDMI will write data in 17 consecutive sub-frames.	0
word_length_value_mode	[3]	0: automatically set 1: manually set If 0 for automatic setting, word length value will be chosen by value of channel status from decoded SPDIF format as reported in SPDIFIN_CH_STATUS_1. word_length. If user sets this register, the receiver will use SPDIFIN_USER_VALUE[3:0] value as word length instead of decoded data from channel status as reported in SPDIFIN_CH_STATUS_1.word_length.	0
U_V_C_P_report	[2]	0: neglects 'user_bit', 'validity_bit', 'channel status' 'parity_bit' of SPDIF format. 1: reports 'user_bit', 'validity_bit', 'channel status' 'parity_bit' of SPDIF format Report will be via HDMI for each subframe. Valid only if SPDIFIN_CONFIG. data_align is set for 32bit mode; see also SPDIFIN_DATA_BUF_x.	0
-	[1]	Reserved (Must be '1')	1
data_align	[0]	0: 16bit mode 1: 32bit mode 16bit: only takes 16bits from MSB in a subframe of SPDIF	0

		format then concatenates two consecutive 16bit data in one 32bit register of SPDIFIN_DATA_BUF_x. 32bit: data from one subframe with zero padding to MSB part. (ex: 0x00ffffff for 24bit data) When stream mode, you should set 'word_length_value_mode' as 1 and set SPDIFIN_USER_VALUE.word_length_manual as 0b000. - These two modes will be applied to both modes of SPDIFIN_CONFIG.data_type, i.e. PCM or stream; see also SPDIFIN_DATA_BUF_x.	

SPDIFIN_CONFIG_2 **0xF0257014**

Name	Type	Description	Default
SPDIFIN_CONFIG_2	R/W	SPDIFIN Configuration Register 2	0x02

Name	Bit	Description	Default
-	[7:4]	Reserved	0x0
clk_divisor	[3:0]	SPDIFIN_internal_clock = system_clock / (clk_divisor + 1) (SPDIFIN_internal_clock ≤ 135 Mhz) SPDIFIN over-samples the SPDIF input signal with internally made clock which is divided from system clock. Recommended over-sampling ratio is 8~10, thus following calculation holds. Recommended SPDIFIN_internal_clock = Sampling Frequency of SPDIF Input Signal * 64 bits * 10 times-over-sampling (ex) 48 kHz * 64 bits * 10 times-over-sampling = 31 Mhz	0x0

SPDIFIN_USER_VALUE_1 **0xF0257020**

Name	Type	Description	Default
SPDIFIN_USER_VALUE_1	R/W	SPDIFIN User Value 1	0x00

Name	Bit	Description	Default
repetition_time_manual_low	[7:4]	Repetition time[3:0] Repetition_time_manual register 12bits value. This register is low 4bits. Will be used for counting one block of stream data; valid only when SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode. (Unit: frames (1 frame = 2 subframes) of SPDIF format) The value should be (actual repetition time – 1). For example, if you want to use 1536 as manually set repetition time, you should write 1535.	0x0
word_length_manual	[3:0]	Word length Will be used as size for transferring data to memory via HDMI; valid only when SPDIFIN_CONFIG.word_length_value_mode is set for manual mode; see also SPDIFIN_DATA_BUF_x. [0] is 1 [0] is 0 [3:1] 101: 24 bits 20 bits 001: 23 bits 19 bits 010: 22 bits 18 bits 011: 21 bits 17 bits 100: 20 bits 16 bits	0x0

SPDIFIN_USER_VALUE_2 **0xF0257024**

Name	Type	Description	Default
SPDIFIN_USER_VALUE_2	R/W	SPDIFIN User Value 2	0x00

Name	Bit	Description	Default
repetition_time_manual_high	[7:0]	Repetition time[11:4] Repetition_time_manual register 12bits value. This register is high 8bits. Will be used for counting one block of stream data; valid only when SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode. (Unit: frames (1 frame = 2 subframes) of SPDIF format) The value should be (actual repetition time – 1). For example, if you want to use 1536 as manually set repetition time, you should write 1535.	0x00

SPDIFIN_USER_VALUE_3 **0xF0257028**

Name	Type	Description	Default
SPDIFIN_USER_VALUE_3	R/W	SPDIFIN User Value 3	0x00

Name	Bit	Description	Default
burst_payload_length_manual_low	[7:0]	Burst_payload_length_manual[7:0] Burst_payload_length register is 16bits value. This register is low 8bits Valid only when SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode. (Unit: bits)	0x00

SPDIFIN_USER_VALUE_4 **0xF025702C**

Name	Type	Description	Default
SPDIFIN_USER_VALUE_4	R/W	SPDIFIN User Value 4	0x00

Name	Bit	Description	Default
burst_payload_length_manual_high	[7:0]	Burst_payload_length_manual[15:8] Burst_payload_length register is 16bits value. this register is high 8bits Valid only when SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode. (Unit: bits)	0x00

8.4.3.2 Channel Status Registers

SPDIFIN_CH_STATUS_0_1 **0xF0257030**

Name	Type	Description	Default
SPDIFIN_CH_STATUS_0_1	R	SPDIFIN Channel Status Register 0-1 to report the channel status of input SPDIF stream.	0x00

Name	Bit	Description	Default
channel_status_mode	[7:6]	00: mode 0 others: reserved	0b00
emphasis	[5:3]	000: emphasis not indicated 100: emphasis – CD type	0b000
copyright_assertion	[2]	0: copyright 1: no copyright	0
audio_sample_word	[1]	0: linear PCM 1: non-linear PCM	0
channel_status_block	[0]	0: consumer format 1: professional format	0
This register will be updated every 192 frames(1 block) of SPDIF format. SPDIFIN_CH_STATUS_0_1 [7:0] is matched internal register SPDIFIN_CH_STATUS_0 [7:0].			

SPDIFIN_CH_STATUS_0_2 **0xF0257034**

Name	Type	Description	Default
SPDIFIN_CH_STATUS_0_2	R	SPDIFIN Channel Status Register 0-2 to report the channel status of input SPDIF stream.	0x00

Name	Bit	Description	Default
category_code	[7:0]	equipment type : [8:15] CD player: 1000_0000 DAT player: 1100_000L DCC player: 1100_001L Mini disc: 1001_001L (L : information about generation status of the material)	0x00

SPDIFIN_CH_STATUS_0_3 **0xF0257038**

Name	Type	Description	Default
SPDIFIN_CH_STATUS_0_3	R	SPDIFIN Channel Status Register 0-3 to report the channel status of input SPDIF stream.	0x00

Name	Bit	Description	Default
channel_number	[7:4]	Channel Number (bit 20 is LSB)	0x0
source_number	[3:0]	Source Number (bit 16 is LSB)	0x0

SPDIFIN_CH_STATUS_0_4 **0xF0257038**

Name	Type	Description	Default
SPDIFIN_CH_STATUS_0_4	R	SPDIFIN Channel Status Register 0-3 to report the channel status of input SPDIF stream.	0x00

Name	Bit	Description	Default
-	[7:6]	Reserved	0b00
clock_accuracy	[5:4]	Clock accuracy 00: level II, ±1000ppm 01: level I, ±50ppm 10: level III, variable pitch shifted	0b00
sampling_frequency	[3:0]	Sampling Frequency 0100: 22.05kHz 0000: 44.1 kHz 1000: 88.2kHz 1100: 176.4kHz 0110: 24kHz 0010: 48 kHz 1010: 96kHz 1110: 192kHz 0011: 32 kHz	0x0

SPDIFIN_CH_STATUS_1 **0xF0257040**

Name	Type	Description	Default
SPDIFIN_CH_STATUS_1	R	SPDIFIN Channel Status Register 1 to report the channel status of input SPDIF stream.	0x00

Name	Bit	Description	Default
-	[7:4]	Reserved	0x0
word_length	[3:1]	Word Length (field_size = 1) (field_size = 0) 000: not indicated not indicated 101: 24 bits 20 bits 100: 23 bits 19 bits 010: 22 bits 18 bits 110: 21 bits 17 bits 001: 20 bits 16 bits	0b000
field_size	[0]	Field Size 0: maximum length 20 bits 1: maximum length 24 bits	0

8.4.3.3 SPDIFIN Info Register

SPDIFIN_CH_STATUS_1 **0xF0257048**

Name	Type	Description	Default
SPDIFIN_CH_STATUS_1	R	SPDIFIN Frame Period Register 1	0x00

Name	Bit	Description	Default
frame_cnt_low	[7:0]	frame count value [7:0] Frame_cnt register is 16bits value. This is low 8bits. The period of a frame (2 sub-frames), thus this register will be updated every 2 sub-frames. It will be measured by 'SPDIFIN_internal_clk' made with SPDIFIN_CONFIG.clk_divisor. (Unit: SPDIF_internal_clk Cycles) (Recommended value for locking incoming signals : Over 0x220 (8.5timesx64bits))	0x00

SPDIFIN_CH_STATUS_2 **0xF025704C**

Name	Type	Description	Default
SPDIFIN_CH_STATUS_2	R	SPDIFIN Frame Period Register 2	0x00

Name	Bit	Description	Default
frame_cnt_high	[7:0]	frame count value [15:8] Frame_cnt register is 16bits value. This is high 8bits. The period of a frame (2 sub-frames), thus this register will be updated every 2 sub-frames. It will be measured by 'SPDIFIN_internal_clk' made with SPDIFIN_CONFIG.clk_divisor. (Unit: SPDIF_internal_clk Cycles) (Recommended value for locking incoming signals : Over 0x220 (8.5timesx64bits))	0x00

SPDIFIN_Pc_INFO_1 **0xF0257050**

Name	Type	Description	Default
SPDIFIN_Pc_INFO_1	R	SPDIFIN Pc Info Register 1	0x00

Name	Bit	Description	Default
error_flag	[7]	0: valid burst payload 1: burst payload may contain errors	0
-	[6:5]	Reserved	0b00
compressed_data_type	[4:0]	0d: null data 1d: Dolby AC-3 2d: reserved 3d: pause 4d: MPEG-1 layer 1 5d: MPEG-1 layer 2 or 3 or MPEG-2 w/o extension 6d: MPEG-2 w/ extension 7d: reserved 8d: MPEG-2 layer 1 low sampling freq. 9d: MPEG-2 layer 2 or 3 low sampling freq. 10d: reserved 11d, 12d, 13d: DTS 14d~31d: reserved	0b00000

SPDIFIN_Pc_INFO_2 **0xF0257054**

Name	Type	Description	Default
SPDIFIN_Pc_INFO_2	R	SPDIFIN Pc Info Register 2	0x00

Name	Bit	Description	Default
bit_stream_number	[7:5]	bit stream number.	0b000
data_type_dependent_info	[4:0]	data type dependent information.	0b00000

SPDIFIN_Pd_INFO_1 **0xF0257058**

Name	Type	Description	Default
SPDIFIN_Pd_INFO_1	R	SPDIFIN Pd Info Register 1	0x00

Name	Bit	Description	Default
burst_payload_length_low	[7:0]	length of burst payload [7:0] (Unit: bits)	0b000

SPDIFIN_Pd_INFO_2 **0xF025705C**

Name	Type	Description	Default
SPDIFIN_Pd_INFO_2	R	SPDIFIN Pd Info Register 2	0x00

Name	Bit	Description	Default
burst_payload_length_high	[7:0]	length of burst payload [15:8] (Unit: bits)	0b000

SPDIFIN_DATA_BUF0_1/2/3 **0xF0257060/64/68**

Name	Type	Description	Default
SPDIFIN_DATA_BUF_0_1 SPDIFIN_DATA_BUF_0_2 SPDIFIN_DATA_BUF_0_3	R	SPDIFIN Data Buffer Registers	0x00

Name	Bit	Description	Default
received_data_0_1 received_data_0_2 received_data_0_3	[7:0] [7:0] [7:0]	PCM or stream data for 1st burst of HDMI SPDIFIN_DATA_BUF_0_1 = SPDIFIN_DATA_BUF_0[7:0] SPDIFIN_DATA_BUF_0_2 = SPDIFIN_DATA_BUF_0[15:8] SPDIFIN_DATA_BUF_0_3 = SPDIFIN_DATA_BUF_0[23:16] When SPDIFIN_CONFIG.data_align is 0 for 16bit received_data = {data_(N)th, data_(N+1)th} When SPDIFIN_CONFIG.data_align is 1 for 32bit received_data = {U, V, C, P, zero-padding, data[n:0]} received_data = {zero-padding, data[n:0]} (if SPDIFIN_CONFIG.U_V_P_report is 0) ('n' is dependent on SPDIFIN_CH_STATUS_1. word_length when SPDIFIN_CONFIG. data_type is 0 for PCM. Or 'n' is 15 when SPDIFIN_CONFIG.data_type is 1 for stream) If SPDIFIN_CONFIG.HDMI_burst_size is 1 burst, HDMI accesses only this data register.	0b000

SPDIFIN_USER_BUF_0 **0xF025706C**

Name	Type	Description	Default
SPDIFIN_USER_BUF_0	R	SPDIFIN User Buffer Register 0	0x00

Name	Bit	Description	Default
received_data_user_0	[7:4]	User bit of 1st burst of HDMI received_data[7:4] = SPDIFIN_DATA_BUF_0[31:28]	0x0
-	[3:0]	Reserved	0x0

SPDIFIN_DATA_BUF1_1/2/3 **0xF0257070/74/78**

Name	Type	Description	Default
SPDIFIN_DATA_BUF_1_1 SPDIFIN_DATA_BUF_1_2 SPDIFIN_DATA_BUF_1_3	R	SPDIFIN Data Buffer Registers	0x00

Name	Bit	Description	Default
received_data_1_1 received_data_1_2 received_data_1_3	[7:0] [7:0] [7:0]	PCM or stream data for 2nd burst of HDMI SPDIFIN_DATA_BUF_0_1 = SPDIFIN_DATA_BUF_0[7:0] SPDIFIN_DATA_BUF_0_2 = SPDIFIN_DATA_BUF_0[15:8] SPDIFIN_DATA_BUF_0_3 = SPDIFIN_DATA_BUF_0[23:16] When SPDIFIN_CONFIG.data_align is 0 for 16bit received_data = {data_(N)th, data_(N+1)th} When SPDIFIN_CONFIG.data_align is 1 for 32bit received_data = {U, V, C, P, zero-padding, data[n:0]} received_data = {zero-padding, data[n:0]} (if SPDIFIN_CONFIG.U_V_P_report is 0) ('n' is dependent on SPDIFIN_CH_STATUS_1. word_length when SPDIFIN_CONFIG. data_type is 0 for PCM. Or 'n' is 15 when SPDIFIN_CONFIG.data_type is 1 for stream) If SPDIFIN_CONFIG.HDMI_burst_size is 1 burst, HDMI accesses only this data register.	0b000

SPDIFIN_USER_BUF_1 **0xF025707C**

Name	Type	Description	Default
SPDIFIN_USER_BUF_1	R	SPDIFIN User Buffer Register 1	0x00

Name	Bit	Description	Default
received_data_user_1	[7:4]	User bit of 2nd burst of HDMI received_data[7:4] = SPDIFIN_DATA_BUF_1[31:28]	0x0
-	[3:0]	Reserved	0x0

8.4.4 I2S Registers

Name	Offset	Type	Description	Default
I2S_CLK_CON	0x0000	R/W	I2S Clock Enable Register	0x00
I2S_CON_1	0x0004	R/W	I2S Control Register 1	0x00
I2S_CON_2	0x0008	R/W	I2S Control Register 2	0x00
I2S_PIN_SEL_0	0x000C	R/W	I2S Input Pin Selection Register 0	0x77
I2S_PIN_SEL_1	0x0010	R/W	I2S Input Pin Selection Register 1	0x77
I2S_PIN_SEL_2	0x0014	R/W	I2S Input Pin Selection Register 2	0x77
I2S_PIN_SEL_3	0x0018	R/W	I2S Input Pin Selection Register 3	0x07
I2S_DSD_CON	0x001C	R/W	I2S DSD Control Register	0x02
I2S_MUX_CON	0x0020	R/W	I2S In/Mux Control Register	0x60
I2S_CH_ST_CON	0x0024	R/W	I2S Channel Status Control Register	0x00
I2S_CH_ST_0	0x0028	R/W	I2S Channel Status Block 0	0x00
I2S_CH_ST_1	0x002C	R/W	I2S Channel Status Block 1	0x00
I2S_CH_ST_2	0x0030	R/W	I2S Channel Status Block 2	0x00
I2S_CH_ST_3	0x0034	R/W	I2S Channel Status Block 3	0x00
I2S_CH_ST_4	0x0038	R/W	I2S Channel Status Block 4	0x00
I2S_CH_ST_SH_0	0x003C	R	I2S Channel Status Block Shadow Register 0	0x00
I2S_CH_ST_SH_1	0x0040	R	I2S Channel Status Block Shadow Register 1	0x00
I2S_CH_ST_SH_2	0x0044	R	I2S Channel Status Block Shadow Register 2	0x00
I2S_CH_ST_SH_3	0x0048	R	I2S Channel Status Block Shadow Register 3	0x00
I2S_CH_ST_SH_4	0x004C	R	I2S Channel Status Block Shadow Register 4	0x00
I2S_VD_DATA	0x0050	R/W	I2S Audio Sample Validity Register	0x00
I2S_MUX_CH	0x0054	R/W	I2S Channel Enable Register	0x03
I2S_MUX_CUV	0x0058	R/W	I2S CUV Enable Register	0x03
I2S_IRQ_MASK	0x005C	R/W	I2S Interrupt Request Mask Register	0x03
I2S_IRQ_STATUS	0x0060	R/W	I2S Interrupt Request Status Register	0x00
I2S_CH0_L_0	0x0064	R	I2S PCM Output Data Register	0x00
I2S_CH0_L_1	0x0068	R	I2S PCM Output Data Register	0x00
I2S_CH0_L_2	0x006C	R	I2S PCM Output Data Register	0x00
I2S_CH0_L_3	0x0070	R	I2S PCM Output Data Register	0x00
I2S_CH0_R_0	0x0074	R	I2S PCM Output Data Register	0x00
I2S_CH0_R_1	0x0078	R	I2S PCM Output Data Register	0x00
I2S_CH0_R_2	0x007C	R	I2S PCM Output Data Register	0x00
I2S_CH0_R_3	0x0080	R	I2S PCM Output Data Register	0x00
I2S_CH1_L_0	0x0084	R	I2S PCM Output Data Register	0x00
I2S_CH1_L_1	0x0088	R	I2S PCM Output Data Register	0x00
I2S_CH1_L_2	0x008C	R	I2S PCM Output Data Register	0x00
I2S_CH1_L_3	0x0090	R	I2S PCM Output Data Register	0x00
I2S_CH1_R_0	0x0094	R	I2S PCM Output Data Register	0x00
I2S_CH1_R_1	0x0098	R	I2S PCM Output Data Register	0x00
I2S_CH1_R_2	0x009C	R	I2S PCM Output Data Register	0x00
I2S_CH1_R_3	0x00A0	R	I2S PCM Output Data Register	0x00
I2S_CH2_L_0	0x00A4	R	I2S PCM Output Data Register	0x00
I2S_CH2_L_1	0x00A8	R	I2S PCM Output Data Register	0x00
I2S_CH2_L_2	0x00AC	R	I2S PCM Output Data Register	0x00
I2S_CH2_L_3	0x00B0	R	I2S PCM Output Data Register	0x00
I2S_CH2_R_0	0x00B4	R	I2S PCM Output Data Register	0x00
I2S_CH2_R_1	0x00B8	R	I2S PCM Output Data Register	0x00

I2S_CH2_R_2	0x00BC	R	I2S PCM Output Data Register	0x00
I2S_Ch2_R_3	0x00C0	R	I2S PCM Output Data Register	0x00
I2S_CH3_L_0	0x00C4	R	I2S PCM Output Data Register	0x00
I2S_CH3_L_1	0x00C8	R	I2S PCM Output Data Register	0x00
I2S_CH3_L_2	0x00CC	R	I2S PCM Output Data Register	0x00
I2S_CH3_R_0	0x00D0	R	I2S PCM Output Data Register	0x00
I2S_CH3_R_1	0x00D4	R	I2S PCM Output Data Register	0x00
I2S_CH3_R_2	0x00D8	R	I2S PCM Output Data Register	0x00
I2S_CUV_L_R	0x00DC	R	I2S CUV Output Data Register	0x00

8.4.4.1 Control Registers

I2S_CLK_CON **0xF0258000**

Name	Type	Description	Default
I2S_CLK_CON	R/W	I2S Clock Enable Register	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0
i2s_en	[0]	I2S Clock Enable 0: i2s will be disabled (default) 1: i2s will be activated You must set i2s_en, after other registers are configured. when you want to reset the i2s, this register is 0□ 1.	0

I2S_CON_1 **0xF0258004**

Name	Type	Description	Default
I2S_CON_1	R/W	I2S Control Register 1	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
r_sc_pol	[1]	SDATA is synchronous to 0 : SCLK falling edge 1 : SCLK rising edge	0
r_ch_pol	[0]	LRCLK polarity 0 : Left Channel for Low polarity 1 : Left Channel for High polarity	0

I2S_CON_2 **0xF0258008**

Name	Type	Description	Default
I2S_CON_2	R/W	I2S Control Register 2	0x00

Name	Bit	Description	Default
-	[7]	Reserved	0
mlsb	[6]	0: MSB first mode 1: LSB first mode	0
bit_ch	[5:4]	Bit clock per Frame(Frame = left + right) 0b00: 32fs 0b01: 48fs 0b10: 64fs	0b00
data_num	[3:2]	Serial data bit per channel 0b01: 16bit 0b10: 20bit 0b11: 24bit	0b00
i2s_mode	[1:0]	0b00: I2S basic format 0b10: left justified format 0b11: right justified format	0b00

I2S_PIN_SEL_0 **0xF025800C**

Name	Type	Description	Default
I2S_PIN_SEL_0	R/W	I2S Input Pin Selection Register 0	0x77

Name	Bit	Description	Default
-	[7]	Reserved	0
pin_sel_1	[6:4]	SCLK(I2S) & DSD_D0(DSD) selection 0b111 : i_i2s_in[1] 0b110 : i_i2s_in[6] 0b101 : i_i2s_in[5] 0b100 : i_i2s_in[4] 0b011 : i_i2s_in[3] 0b010 : i_i2s_in[2] 0b001 : i_i2s_in[1] 0b000 : i_i2s_in[0]	0b111
-	[3]	Reserved	0
pin_sel_0	[2:0]	LRCK(I2S) & DSD_CLK(DSD) selection 0b111 : i_i2s_in[0] 0b110 : i_i2s_in[6] 0b101 : i_i2s_in[5] 0b100 : i_i2s_in[4] 0b011 : i_i2s_in[3] 0b010 : i_i2s_in[2] 0b001 : i_i2s_in[1] 0b000 : i_i2s_in[0]	0b111

I2S_PIN_SEL_1 **0xF0258010**

Name	Type	Description	Default
I2S_PIN_SEL_1	R/W	I2S Input Pin Selection Register 1	0x77

Name	Bit	Description	Default
-	[7]	Reserved	0
pin_sel_3	[6:4]	SDATA_1(I2S) & DSD_D2(DSD) selection 0b111 : i_i2s_in[3] 0b110 : i_i2s_in[6] 0b101 : i_i2s_in[5] 0b100 : i_i2s_in[4] 0b011 : i_i2s_in[3] 0b010 : i_i2s_in[2] 0b001 : i_i2s_in[1] 0b000 : i_i2s_in[0]	0b111
-	[3]	Reserved	0
pin_sel_2	[2:0]	SDATA_0(I2S) & DSD_D1(DSD) selection 0b111 : i_i2s_in[2] 0b110 : i_i2s_in[6] 0b101 : i_i2s_in[5] 0b100 : i_i2s_in[4] 0b011 : i_i2s_in[3] 0b010 : i_i2s_in[2] 0b001 : i_i2s_in[1] 0b000 : i_i2s_in[0]	0b111

I2S_PIN_SEL_2 **0xF0258014**

Name	Type	Description	Default
I2S_PIN_SEL_2	R/W	I2S Input Pin Selection Register 2	0x77

Name	Bit	Description	Default
-	[7]	Reserved	0
pin_sel_5	[6:4]	SDATA_3(I2S) & DSD_D4(DSD) selection 0b111 : i_i2s_in[5] 0b110 : i_i2s_in[6] 0b101 : i_i2s_in[5] 0b100 : i_i2s_in[4] 0b011 : i_i2s_in[3] 0b010 : i_i2s_in[2] 0b001 : i_i2s_in[1] 0b000 : i_i2s_in[0]	0b111
-	[3]	Reserved	0
pin_sel_4	[2:0]	SDATA_2(I2S) & DSD_D3(DSD) selection 0b111 : i_i2s_in[4] 0b110 : i_i2s_in[6] 0b101 : i_i2s_in[5] 0b100 : i_i2s_in[4] 0b011 : i_i2s_in[3] 0b010 : i_i2s_in[2] 0b001 : i_i2s_in[1] 0b000 : i_i2s_in[0]	0b111

I2S_PIN_SEL_3 **0xF0258018**

Name	Type	Description	Default
I2S_PIN_SEL_3	R/W	I2S Input Pin Selection Register 3	0x77

Name	Bit	Description	Default
-	[7:3]	Reserved	0
pin_sel_6	[2:0]	DSD_D5(DSD) selection 0b111 : i_i2s_in[6] 0b110 : i_i2s_in[6] 0b101 : i_i2s_in[5] 0b100 : i_i2s_in[4] 0b011 : i_i2s_in[3] 0b010 : i_i2s_in[2] 0b001 : i_i2s_in[1] 0b000 : i_i2s_in[0]	0b111

I2S_DSD_CON **0xF025801C**

Name	Type	Description	Default
I2S_DSD_CON	R/W	I2S DSD Control Register	0x02

Name	Bit	Description	Default
-	[7:2]	Reserved	0
r_dsd_pol	[1]	1 : DSD_DATA change at DSD_CLK rising edge 0 : DSD_DATA change at DSD_CLK falling edge	1
dsd_en	[0]	1 : DSD module enable 0 : DSD module disable	0

I2S_IN_MUX_CON **0xF0258020**

Name	Type	Description	Default
I2S_IN_MUX_CON	R/W	I2S In/Mux Control Register	0x60

Name	Bit	Description	Default
f_num	[7:5]	Number of stage of noise filter for I2S input pins 000: no filtering 001: 2 stage filter 010: 3 stage filter 011: 4 stage filter 100: 5 stage filter others : reserved	0b000
in_en	[4]	Enable i2s_in, a sub-module at the input stage. 0 : i2s_in module disable 1 : i2s_in module enable All output data is ‘0’ if disabled.	0
audio_sel	[3:2]	Audio selection 0b00 : SPDIF audio data enable 0b01 : I2S audio data enable 0b10 : DSD audio data enable	0b01
CUV_sel	[1]	C.U.V. Selection 0 : SPDIF C.U.V. data enable 1 : I2S C.U.V. data enable	1
mux_en	[0]	Enable i2s_mux, a sub-module for audio selection. 0 : i2s_mux module disable 1 : i2s_mux module enable All output data is ‘0’ if disabled.	0

8.4.4.2 Channel Status Register

I2S_CH_ST_CON **0xF0258024**

Name	Type	Description	Default
I2S_CH_ST_CON	R/W	I2S Channel Status Control Register	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0
channel_status_reload	[0]	0: The shadow channel status registers are updated. 1: Set this bit to update the shadow channel status registers with the values updated in I2S_CH_ST_0 ~ I2S_CH_ST_4. When the shadow channel status registers are updated, this bit is cleared.	0

Channel status information needs to be applied to the audio stream at the IEC 60958 block boundary. For this synchronization, there are two register sets for channel status block. Users can set the channel status registers, I2S_CH_ST_0~I2S_CH_ST_4, while the I2S Rx module still refers to the shadow channel status registers, I2S_CH_ST_SH_0~ I2S_CH_ST_CH4. To reflect the user configuration in the channel status registers, users should set 'channel_status_reload' bit in I2S_CH_ST_CON then I2S Rx module copies the channel status registers into the shadow channel status registers at the beginning of an IEC-60958 block.

I2S_CH_ST_0, I2S_CH_ST_SH_0 **0xF0258028/3C**

Name	Type	Description	Default
I2S_CH_ST_0	R/W	I2S Channel Status Block 0	0x00
I2S_CH_ST_SH_0	R	I2S Channel Status Block Shadow 0	0x00

Name	Bit	Description	Default
channel_status_mode	[7:6]	0b00 : Mode 0 others : Reserved	0
emphasis	[5:3]	When bit1 = 0, 0b000 : 2 audio channels without pre-emphasis* 0b001 : 2 audio channels with 50us / 15us pre-emphasis When bit1 = 1, 0b000 : default state	0
copyright	[2]	0 : copyright 1 : no copyright	0
audio_sample_word	[1]	0 : linear PCM 1 : non-linear PCM	0
channel_status_block	[0]	0 : consumer format 1 : professional format	0

Note that bits listed here in Channel Status Registers look swapped from those in IEC-60958-3 Specification, as the bit order is different (LSB is right-most bit).

I2S_CH_ST_1, I2S_CH_ST_SH_1 **0xF025802C/40**

Name	Type	Description	Default
I2S_CH_ST_1	R/W	I2S Channel Status Block 1	0x00
I2S_CH_ST_SH_1	R	I2S Channel Status Block Shadow 1	0x00

Name	Bit	Description	Default
category	[7:0]	Equipment type CD player : 0000_0001 DAT player : L000_0011 DCC player : L100_0011 Mini disc : L100_1001 (L : information about generation status of the material)	0

I2S_CH_ST_2, I2S_CH_ST_SH_2 **0xF0258030/44**

Name	Type	Description	Default
I2S_CH_ST_2	R/W	I2S Channel Status Block 2	0x00
I2S_CH_ST_SH_2	R	I2S Channel Status Block Shadow 2	0x00

Name	Bit	Description	Default
channel_number	[7:4]	Channel Number Note that bit4 is LSB.	0
source_number	[3:0]	Source Number Note that bit0 is LSB.	0

I2S_CH_ST_3, I2S_CH_ST_SH_3 **0xF0258034/48**

Name	Type	Description	Default
I2S_CH_ST_3	R/W	I2S Channel Status Block 3	0x00
I2S_CH_ST_SH_3	R	I2S Channel Status Block Shadow 3	0x00

Name	Bit	Description	Default
-	[7:6]	Reserved	0b00
Clock_Accuracy	[5:4]	Clock Accuracy as specified in IEC-60958-3 0b01 : Level I, ±50 ppm 0b00 : Level II, ±1000 ppm 0b10 : Level III, variable pitch shifted	0b00
Sampling_Frequency	[3:0]	Sampling Frequency as specified in IEC-60958-3 0b0000 : 44.1 kHz 0b0010 : 48 kHz 0b0011 : 32 kHz 0b1010 : 96 kHz …	0

I2S_CH_ST_4, I2S_CH_ST_SH_4 **0xF0258038/4C**

Name	Type	Description	Default
I2S_CH_ST_4	R/W	I2S Channel Status Block 4	0x00
I2S_CH_ST_SH_4	R	I2S Channel Status Block Shadow 4	0x00

Name	Bit	Description	Default
Org_Sampling_Freq	[7:4]	Original Sampling Frequency 0b1111 : 44.1Khz 0b0111 : 88.2Khz 0b1011 : 22.05Khz 0b0011 : 176.4Khz … For other frequencies, refer to original sampling frequency specified in IEC-60958-3	0x0
Word_Length	[3:1]	Word length Max. length 24bits 20 bits 0b000 : not defined not defined 0b001 : 20 bits 16bits 0b010 : 22 bits 18bits 0b100 : 23 bits 19bits 0b101 : 24 bits 20bits 0b110 : 21 bits 17bits	0b000
Max_Word_Length	[0]	Maximum sample word length 1 : 24 bits 0 : 20 bits	0

I2S_VD_DATA **0xF0258050**

Name	Type	Description	Default
I2S_VD_DATA	R/W	I2S Audio Sample Validity Register	0x00

Name	Bit	Description	Default
-	[7:1]	Reserved	0b0000000
validity_flag	[0]	Validity bit 0 : audio sample is reliable 1 : audio sample is unreliable	0

8.4.4.3 Mux Control Register

I2S_MUX_CH **0xF0258054**

Name	Type	Description	Default
I2S_MUX_CH	R/W	I2S Channel Enable Register	0x03

Name	Bit	Description	Default
CH3_R_en	[7]	0 : Channel 3 right audio data output is disable 1 : Channel 3 right audio data output is enable	0
CH3_L_en	[6]	0 : Channel 3 left audio data output is disable 1 : Channel 3 left audio data output is enable	0
CH2_R_en	[5]	0 : Channel 2 right audio data output is disable 1 : Channel 2 right audio data output is enable	0
CH2_L_en	[4]	0 : Channel 2 left audio data output is disable 1 : Channel 2 left audio data output is enable	0
CH1_R_en	[3]	0 : Channel 1 right audio data output is disable 1 : Channel 1 right audio data output is enable	0
CH1_L_en	[2]	0 : Channel 1 left audio data output is disable 1 : Channel 1 left audio data output is enable	0
CH0_R_en	[1]	0 : Channel 0 right audio data output is disable 1 : Channel 0 right audio data output is enable	1
CH0_L_en	[0]	0 : Channel 0 left audio data output is disable 1 : Channel 0 left audio data output is enable	1

I2S_MUX_CUV **0xF0258058**

Name	Type	Description	Default
I2S_MUX_CUV	R/W	I2S CUV Enable Register	0x03

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
CUV_R_en	[1]	0 : Right channel CUV data is disable 1 : Right channel CUV data is enable	1
CUV_L_en	[0]	0 : Left channel CUV data is disable 1 : Left channel CUV data is enable	1

8.4.4.4 Interrupt Control Registers

I2S_IRQ_MASK **0xF025805C**

Name	Type	Description	Default
I2S_IRQ_MASK	R/W	I2S Interrupt Request Mask Register	0x03

Name	Bit	Description	Default
-	[7:2]	Reserved	0b000000
int_2_mask	[1]	Disable interrupt request by int_2 interrupt 0: int_2 interrupt is disabled 1: int_2 interrupt is enabled	1
int_1_mask	[0]	Disable interrupt request by int_1 interrupt 0: int_1 interrupt is disabled 1: int_1 interrupt is enabled	1

I2S_IRQ_STATUS **0xF0258060**

Name	Type	Description	Default
I2S_IRQ_STATUS	R/W	I2S Interrupt Request Status Register	0x00

Name	Bit	Description	Default
-	[7:2]	Reserved	0
int_2	[1]	Interrupt status : Wrong register setting This interrupt is asserted when the I2S_CON_2.bit_ch is set to be 32fs while I2S_CON_2.data_num is set to either 20bit or 24bit. According to wrong register setting, Some audio data MSB bits removed. The audio data is not available.	0
int_1	[0]	Interrupt status : Bit Per Channel mismatch This interrupt is asserted when the number of SCLKs in one channel does not match with I2S_CON_2.bit_ch register value.	0
For all bits, the following holds; Reading returns interrupt request status. Writing ‘0’ has no effect. Writing ‘1’ clears the interrupt request.			

8.4.4.5 Output Buffer Registers

I2S_CHX_Y_Z **0xF0258064~0xF02580D8**

Name	Type	Description	Default
I2S_CHX_Y_Z	R	I2S PCM Output Data Register to read the PCM output data from I2S Rx module to HDMI core.	0x00

Name	Bit	Description	Default
I2S_CHX_Y_Z	[7:0]	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCMX_Y[7:0] I2S_CHX_Y_1 = PCMX_Y[15:8] I2S_CHX_Y_2 = PCMX_Y[23:16] I2S_CHX_Y_3 = PCMX_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	0x00

I2S_CUV_L_R **0xF02580DC**

Name	Type	Description	Default
I2S_CUV_L_R	R	I2S CUV Output Data to read the CUV output data from I2S Rx module to HDMI core	0x00

Name	Bit	Description	Default
-	[7]	Reserved	0
CUV_R	[6:4]	VUCP data of Right channel CUV_R[3:0] = {valid bit, user bit, channel state bit, parity bit}	0b000
-	[3]	Reserved	0
CUV_L	[2:0]	VUCP data of Left channel CUV_L[3:0] = {valid bit, user bit, channel state bit, parity bit}	0b000

8.4.5 CEC Registers

Name	Offset	Type	Description	Default
CEC Configure/Status Registers				
CEC_TX_STATUS_0	0x0000	R	CEC Tx status register 0.	0x00
CEC_TX_STATUS_1	0x0004	R	CEC Tx status register 1. Number of blocks transferred.	0x00
CEC_RX_STATUS_0	0x0008	R	CEC Rx status register 0.	0x00
CEC_RX_STATUS_1	0x000C	R	CEC Rx status register 1. Number of blocks received.	0x00
CEC_INTR_MASK	0x0010	R/W	CEC interrupt mask register	0x00
CEC_INTR_CLEAR	0x0014	R/W	CEC interrupt clear register	0x00
CEC_LOGIC_ADDR	0x0020	R/W	HDMI Tx logical address register	0x0F
CEC_DIVISOR_0	0x0030	R/W	Clock divisor for 0.05ms period count ([7:0] of 32-bit)	0x00
CEC_ DIVISOR_1	0x0034	R/W	Clock divisor for 0.05ms period count ([15:8] of 32-bit)	0x00
CEC_DIVISOR_2	0x0038	R/W	Clock divisor for 0.05ms period count ([23:16] of 32-bit)	0x00
CEC_DIVISOR_3	0x003C	R/W	Clock divisor for 0.05ms period count ([31:24] of 32-bit)	0x00
CEC Tx related Registers				
CEC_TX_CTRL	0x0040	R/W	CEC Tx control register	0x10
CEC_TX_BYTE_NUM	0x0044	R/W	Number of blocks in a message to be transferred	0x00
CEC_TX_STATUS_2	0x0060	R	CEC Tx status register 2	0x00
CEC_TX_STATUS_3	0x0064	R	CEC Tx status register 3	0x00
CEC_TX_BUFFER_0 ~ CEC_TX_BUFFER_15	0x0080 ~ 0x00BC	R/W	Byte #0 ~ #15 of CEC message to be transferred. (#0 is transferred 1st)	0x00
CEC Rx related Registers				
CEC_RX_CTRL	0x00C0	R/W	CEC Rx control register	0x00
CEC_RX_STATUS_2	0x00E0	R	CEC Rx status register 2	0x00
CEC_RX_STATUS_3	0x00E4	R	CEC Rx status register 3eived 1st)	0x00
CEC_RX_BUFFER_0 ~ CEC_RX_BUFFER_15	0x0100 ~ 0x013C	R	Byte #0 ~ #15 of CEC message received (#0 is received 1st)	0x00

8.4.5.1 CEC Configure Registers

CEC_TX_STATUS_0 **0xF0259000**

Name	Type	Description	Default
CEC_TX_STATUS_0	R	CEC Tx status register 0.	0x00

Name	Bit	Description	Default
-	[7:4]	Reserved	0b0000
Tx_Error	[3]	CEC Tx_Error interrupt flag. This bit field also indicates the status of Tx_Error interrupt. This bit is valid only when Tx_Done bit is set. O : No error has occurred. 1 : An error has occurred during CEC Tx transfer. It will be cleared: - when set to 0 Tx_Enable bit of CEC_TX_CTRL register - when set Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit in CEC_INTR_CLEAR register	0
Tx_Done	[2]	CEC Tx_Done interrupt flag. This bit field also indicates the status of Tx_Done interrupt. O : Running or Idle 1 : CEC Tx transfer finished. It will be cleared: - when reset Tx_Enable bit of CEC_TX_CTRL_0 - when set Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit in CEC_INTR_CLEAR register	0
Tx_Transferring	[1]	If set RX-Running , this field is valid O : Tx is waiting for CEC Bus 1 : CEC Tx is transferring data via CEC Bus.	0
Tx_Running	[0]	O : Tx Idle 1 : CEC Tx is enabled and is either waiting for the CEC bus or transferring message.	0

CEC_TX_STATUS_1 **0xF0259004**

Name	Type	Description	Default
CEC_TX_STATUS_1	R	CEC Tx status register 1.	0x00

Name	Bit	Description	Default
Tx_Bytes_Transferred	[7:0]	Number of blocks transferred (1 byte = 1 block in a CEC message). After sending CEC message, field will be updated. It will be cleared when set Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit in CEC_Intr_Clear register.	0b00000000

CEC_RX_STATUS_0 **0xF0259008**

Name	Type	Description	Default
CEC_RX_STATUS_0	R	CEC Rx status register 0.	0x00

Name	Bit	Description	Default
-	[7:5]	Reserved	0b000
Rx_BCast	[4]	Broadcast message flag O : Received CEC message is address to a single device. 1 : Received CEC message is a broadcast message. It will be cleared: - when reset Rx_Enable bit of CEC_RX_CTRL_0 - when set Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register	0
Rx_Error	[3]	CEC Rx_Error interrupt flag. This bit field also indicates the status of Rx_Error interrupt. This bit is valid only when Rx_Done bit is set. O : No error has occurred. 1 : An error has occurred in receiving a CEC message It will be cleared: - when reset Rx_Enable bit of CEC_RX_CTRL_0 - when set Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register	0
Rx_Done	[2]	CEC Rx done interrupt Flag. This bit field also indicates the status of Rx_Done interrupt. O : Running or Idle 1 : CEC Rx transfer finished It will be cleared: - when reset Rx_Enable bit of CEC_RX_CTRL_0 - when set Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register	0
Rx_Receiving	[1]	O : Rx is waiting for a CEC message. 1 : Rx is currently receiving data via CEC Bus.	0
Rx_Running	[0]	O : Rx disabled 1 : CEC Rx is enabled and is either waiting for a message on the CEC bus.	0

CEC_RX_STATUS_1 **0xF025900C**

Name	Type	Description	Default
CEC_RX_STATUS_1	R	CEC Rx status register 1.	0x00

Name	Bit	Description	Default
Rx_Bytes_Received	[7:0]	Number of blocks received (1 byte = 1 block in a CEC message). After receiving CEC message, field will be updated. It will be cleared when set Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_Intr_Clear register.	0b00000000

CEC_INTR_MASK **0xF0259010**

Name	Type	Description	Default
CEC_INTR_MASK	R/W	CEC interrupt mask register	0x00

Name	Bit	Description	Default
-	[7:6]	Reserved	0b00
Mask_Intr_Rx_Error	[5]	Rx_Error interrupt mask bit. 0 : Enabled 1 : Disabled.	0
Mask_Intr_Rx_Done	[4]	Rx_Done interrupt mask bit. 0 : Enabled 1 : Disabled.	0
-	[3:2]	Reserved	0b00
Mask_Intr_Tx_Error	[1]	Tx_Error interrupt mask bit. 0 : Enabled 1 : Disabled.	0
Mask_Intr_Tx_Done	[0]	Tx_Done interrupt mask bit. 0 : Enabled 1 : Disabled.	0

CEC_INTR_CLEAR **0xF0259014**

Name	Type	Description	Default
CEC_INTR_CLEAR	R/W	CEC interrupt clear register	0x00

Name	Bit	Description	Default
-	[7:6]	Reserved	0b00
Clear_Intr_Rx_Error	[5]	Rx_Error interrupt clear bit. 0 : No effect 1 : Clear Rx_Error and Rx_Bytes_Received fields in CEC_RX_STATUS_0 and 1 register. It will be cleared after one clock.	0
Clear_Intr_Rx_Done	[4]	Rx_Done interrupt clear bit. 0 : No effect 1 : Clears Rx_Done and Rx_Bytes_Received filds in CEC_RX_STATUS_0 and 1 register. Resets to 0 after one clock.	0
-	[3:2]	Reserved	0b00
Clear_Intr_Tx_Error	[1]	Tx_Error interrupt clear bit. 0 : No effect 1 : Clears Tx_Error and Tx_Bytes_Received filds in CEC_TX_STATUS_0 and 1 register. Resets to 0 after one clock.	0
Clear_Intr_Tx_Done	[0]	Tx_Done interrupt clear bit. 0 : No effect 1 : Clears Tx_Done and Tx_Bytes_Received filds in CEC_TX_STATUS_0 and 1 register. Resets to 0 after one clock.	0

CEC_LOGIC_ADDR **0xF0259020**

Name	Type	Description	Default
CEC_LOGIC_ADDR	R/W	HDMI Tx logical address register	0x00

Name	Bit	Description	Default
-	[7:4]	Reserved	0b0000
Logic_Addr	[3:0]	HDMI Tx logical address (0~15)	0b1111

CEC_ DIVISOR_0 ~ CEC_ DIVISOR_3 **0xF0259030**

Name	Type	Description	Default
CEC_ DIVISOR_0 ~ CEC_ DIVISOR_3	R/W	A 32-bit clock divisor for 0.05ms period count CEC_ DIVISOR_0 : [7: 0] of 32-bit CEC_ DIVISOR_1 : [15: 8] of 32-bit CEC_ DIVISOR_2 : [23:16] of 32-bit CEC_ DIVISOR_3 : [31:24] of 32-bit	0x00

Name	Bit	Description	Default
CEC_Divisor	[7:0]	A divisor used in counting 0.05ms period. This divisor should satisfy the following equation: (CEC_DIVISOR) x (clock cycle time(ns)) = 0.05ms	0b00000000

8.4.5.2 Tx Related Registers

CEC_TX_CTRL **0xF0259040**

Name	Type	Description	Default
CEC_TX_CTRL	R/W	CEC Tx control register 0	0x50

Name	Bit	Description	Default
Reset	[7]	CEC Tx reset bit. 0 : No effect 1 : Immediately resets CEC Tx related registers and state machines to its reset value. Resets to 0 after one clock.	0
Tx_Retrans_Num	[6:4]	Number of retransmissions tried when situations in CEC spec. page CEC-13 occurs. According to the specification, it should be set to 5.	0b001
-	[3:2]	Reserved	0b00
Tx_BCast	[1]	CEC Tx broadcast message bit. This bit indicates whether a CEC message in CEC_TX_BUFFER_00~15 is directly-addressed (addressed to a single device) or broadcast. This bit has effect on determining whether a block transfer is acknowledged or not. (following ACK scheme in CEC Spec.(section CEC 6.1.2)) 0 : Directly-addressed message 1 : Broadcast message.	0
Tx_Start	[0]	CEC Tx start bit. 0 : Tx idle. 1 : Start CEC message transfer (Resets to 0 after start)	0

CEC_TX_BYTE_NUM **0xF0259044**

Name	Type	Description	Default
CEC_TX_BYTE_NUM	R/W	Number of blocks in a message to be transferred.	0x00

Name	Bit	Description	Default
Tx_Byte_Num	[7:0]	Number of blocks in a message to be sent. (1 byte = 1 block in a CEC message).	0b00000000

CEC_TX_STATUS_2 **0xF0259060**

Name	Type	Description	Default
CEC_TX_STATUS_2	R	CEC Tx status register 2	0x00

Name	Bit	Description	Default
Tx_Wait	[7]	CEC Tx signal free time waiting flag bit 0 : Tx is in other state 1 : CEC Tx is waiting for a signal free time(time to stop sending messages after previous attempt to send a message).	0
Tx_Sending_Start_Bit	[6]	CEC Tx start bit sending flag bit 0 : Tx is in other state 1 : CEC Tx is currently sending a start bit.	0
Tx_Sending_Hdr_Blk	[5]	CEC Tx header block sending flag bit 0 : Tx is in other state 1 : CEC Tx is currently sending the header block.	0
Tx_Sending_Data_Blk	[4]	CEC Tx data block sending flag bit 0 : Tx is in other state 1 : CEC Tx is currently sending data blocks.	0
Tx_Latest_Initiator	[3]	CEC Tx last initiator flag bit 0 : This device is not the latest initiator on the CEC bus. 1 : This CEC device is the latest initiator to send a CEC message and no other CEC device sent a message. It will be cleared if Rx detects a start bit on the CEC line or Tx_Enable bit of CEC_Tx_Ctrl_0 is set(i.e. becomes a new initiator)	0
-	[2:0]	Reserved	0b000

CEC_TX_STATUS_3 **0xF0259064**

Name	Type	Description	Default
CEC_TX_STATUS_3	R	CEC Tx status register 3	0x00

Name	Bit	Description	Default
Reserved	[7]	Reserved	0
Tx_Wait_SFT_Succ	[6]	CEC Tx signal free time for successive message transfer waiting flag bit 0 : Tx is in other state 1 : Tx is waiting for a signal free time (SFT) with a precondition that Tx is the most recent initiator on the CEC bus and wants to send another frame immediately after its previous frame. (SFT ≥ 7x2.4ms)	0
Tx_Wait_SFT_New	[5]	CEC Tx signal free time for a new initiator waiting flag bit 0 : Tx is in other state 1 : Tx is waiting for a signal free time (SFT) with a precondition that Tx is a new initiator and wants to send a frame.(SFT ≥ 5x2.4ms)	0
Tx_Wait_SFT_Retran	[4]	CEC Tx signal free time for a new initiator waiting flag bit 0 : Tx is in other state 1 : Tx is waiting for a signal free time (SFT) with a precondition that Tx is attempting a retransmission of the message. (SFT ≥ 3 x2.4ms)	0
Tx_Retrans_Cnt	[3:1]	It indicates current retransmissions count. If 0, no retransmission occurred. It will be cleared when set Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit in CEC_Intr_Clear register.	0b000
Tx_ACK_Failed	[0]	CEC Tx acknowledge failed flag bit 0 : Tx is in other state 1 : Tx is not acknowledged. This bit is set when - ACK bit in a block is logical 1 in a directly-addressed message - ACK bit in a block is logical 0 in a broadcast message	0

CEC_TX_BUFFER_0 ~ CEC_ TX_BUFFER_15 **0xF0259080~0xF02590BC**

Name	Type	Description	Default
CEC_ TX_BUFFER_0 ~ CEC_ TX_BUFFER_15	R/W	Byte #0 ~ #15 of CEC message to be transferred. (#0 is transferred 1st)	0x00

Name	Bit	Description	Default
Tx_Block_0 ~ Tx_Block_15	[7:0]	Byte #0 ~ #15 of CEC message. Each byte corresponds to a block in a message. Block_0 is the header block and Block_1 ~15 are data blocks. Note that initiator and destination logical address in a header block should written by S/W.	0b00000000

8.4.5.3 Rx Related Registers

CEC_ RX_CTRL **0xF02590C0**

Name	Type	Description	Default
CEC_ RX_CTRL	R/W	Byte #0 ~ #15 of CEC message to be transferred. (#0 is transferred 1st)	0x00

Name	Bit	Description	Default
Reset	[7]	CEC Rx reset bit. 0 : No effect 1 : Immediately resets CEC Rx related registers and state machines to its reset value. It will be cleared after one clock.	0
Check_Sampling_Error	[6]	CEC Rx sampling error check enable bit 0 : Do not check sampling error. 1 : Check sampling error while receiving data bits. CEC Rx samples the CEC bus three times (at 1.00, 1.05, 1.10 ms) and checks whether three samples are identical.	0
Check_Low_Time_Error	[5]	CEC Rx low-time error check enable bit	0

		0 : Do not check low-time error. 1 : Check low-time error while receiving data bits. In receiving each bit from the CEC bus, CEC Rx checks the duration of logical 0 from the starting of one bit transfer (falling edge on the CEC bus). Rx checks whether the duration is longer than the maximum time the CEC bus can be in logical 0 (max 1.7 ms).	
Check_Start_Bit_Error	[4]	CEC Rx start bit error check enable bit 0 : Do not check start bit error. 1 : Check start bit error while receiving a start bit. In receiving a start bit from the CEC bus, CEC Rx checks the duration of logical 0 and 1 of a starting bit as specified in CEC spec. page CEC-8. Rx checks whether the duration meets the specification.	0
-	[3:2]	Reserved	0b00
Rx_Host_Busy	[1]	CEC Rx host busy bit 0 : Rx receives incoming message and send acknowledges. 1 : A host processor is unavailable to receive and process CEC messages. Rx sends not acknowledged signal to a message initiator to indicate that a host processor is unavailable to receive and process CEC messages.	0
Rx_Enable	[0]	CEC Rx start bit. 0 : Rx disabled. 1 : Enable CEC Rx module to receive a message. This bit is cleared after receiving a message	0

CEC_ RX_STATUS_2 **0xF02590E0**

Name	Type	Description	Default
CEC_ RX_STATUS_2	R	CEC Rx status register 2	0x00

Name	Bit	Description	Default
Rx_Waiting	[7]	CEC Rx waiting flag bit 0 : Rx is in other state 1 : CEC Rx is waiting for a message.	0
Rx_Receiving_Start_Bit	[6]	CEC Rx start bit receiving flag bit 0 : Rx is in other state 1 : CEC Rx is currently receiving a start bit.	0
Rx_Receiving_Hdr_Blk	[5]	CEC Rx header block receiving flag bit 0 : Rx is in other state 1 : CEC Rx is currently receiving a header block.	0
Rx_Receiving_Data_Blk	[4]	CEC Rx data block receiving flag bit 0 : Rx is in other state 1 : CEC Rx is currently receiving data blocks.	0
-	[3:0]	Reserved	0b0000

CEC_ RX_STATUS_3 **0xF02590E4**

Name	Type	Description	Default
CEC_ RX_STATUS_3	R	CEC Rx status register 3	0x00

Name	Bit	Description	Default
-	[7]	Reserved	0
Sampling_Error	[6]	CEC Rx sampling error flag bit 0 : No sampling error has occurred. 1 : A sampling error has occurred in receiving a message. CEC Rx samples the CEC bus three times (at 1.00, 1.05, 1.10 ms) and sets this bit if - Check_Sampling_Error bit in CEC_RX_CTRL_0 is set and - Three samples are not identical. It will be cleared when set to 0 when Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register.	0
Low_Time_Error	[5]	CEC Rx low-time error flag bit 0 : No low-time error has occurred. 1 : A low-time error has occurred in receiving a message. In receiving each bit from the CEC bus, CEC Rx checks the duration of logical 0 from the starting of one-bit transfer falling edge on the CEC bus). If the duration of longer than the maximum time the CEC bus can be in logical 0 (max 1.7 ms), CEC RX sets this bit. This bit field will be set to 0 when Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register is set.	0
Start_Bit_Error	[4]	CEC Rx start bit error flag bit 0 : No start bit error has occurred. 1 : A start bit error has occurred in receiving a message. In receiving a start bit from the CEC bus, CEC Rx checks the duration of logical 0 and 1 of a starting bit as specified in CEC spec. page CEC-8. If the duration does not meet the spec., CEC RX sets this bit. This bit field will be set to 0 when Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register is set.	0
-	[3:1]	Reserved	0b000
CEC_Line_Error	[0]	CEC Rx line error flag bit 0 : No line error has occurred. 1 : A start bit error line error has occurred in receiving a message. In CEC spec. page CEC-13, CEC line error is defined as a situation that period between two consecutive falling edge is smaller than a minimum data bit period. Rx check for this condition and if it occurs, sends line error notification, i.e. sending logical 0 for more than 1.4~1.6 times of the nominal data bit period (2.4ms). This bit will be cleared: When set Rx_Enable bit of CEC_RX_CTRL_0 When set Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register.	0

CEC_RX_BUFFER_0 ~ CEC_RX_BUFFER_15 **0xF0259100~0xF025913C**

Name	Type	Description	Default
CEC_RX_BUFFER_0 ~ CEC_RX_BUFFER_15	R/W	Byte #0 ~ #15 of CEC message received. (#0 is received 1st)	0x00

Name	Bit	Description	Default
Rx_Block_0 ~ Rx_Block_15	[7:0]	Byte #0 ~ #15 of CEC message. Each byte corresponds to a block in a message. Block_0 is the header block and Block_1 ~15 are data blocks.	0b00000000

8.5 HDMI PHY

8.5.1 Clock Scheme of the HDMI TX PHY Core

8.5.1.1 Using the Internal Video PLL for Pixel Clock Generation

The HDMI TX PHY has an internal video PLL and can generate a pixel clock. LCD controller uses the pixel clock to deliver its video data into HDMI LINK layer. HDMI Link layer transfers the TMDS input data (TXDn_E/O[0:9], n=0,1,2) into PHY layer with TMDS clock, which is also generated in TX PHY. For the sake of enough timing margin between LINK and PHY, the TMDS clock into PHY (TMDS_CLKHI) has a half frequency of the TMDS clock (TMDS_CLKO). 20bit interface per data channel is used to compensate the halved frequency. A timing skew between TMDS_CLKHI and TMDS_CLKO due to dividing and clock buffering is treated in de-skewing block within PHY.

One of the advantages of integrated video PLL is good pixel clock jitter. Figure 8.5 _Hlk212869678shows the clock scheme of HDMI TX PHY core using the internally generated pixel clock.

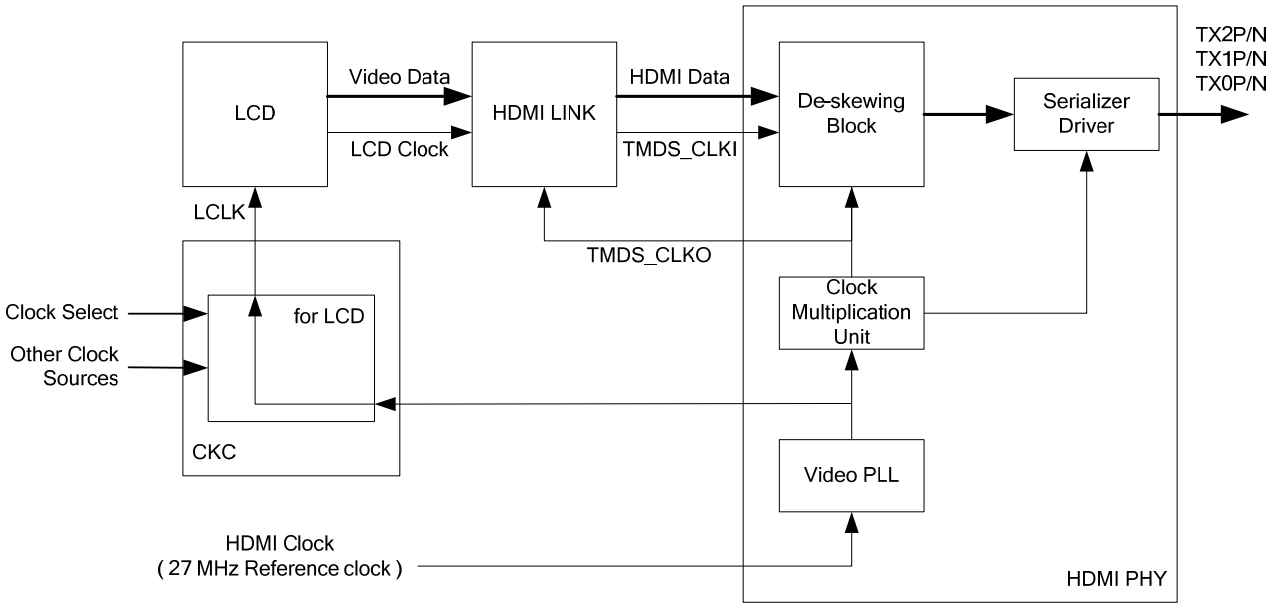


Figure 8.5 HDMI Clock Scheme Using the Integrated Video PLL for Pixel Clock Generation

8.5.1.2 Using an External Video PLL for Pixel Clock Generation

In case a separate video PLL is used for pixel clock generation, the internal video PLL can be by-passed or configured as a jitter-filtering PLL. A by-pass option can save power and jitter-filtering option can reduce pixel clock jitter. In jitter-filtering configuration, the output frequency of video PLL is adjusted for CMU. Figure 8.6 and Figure 8.7 shows the clock scheme of HDMI TX PHY core using an external video PLL.

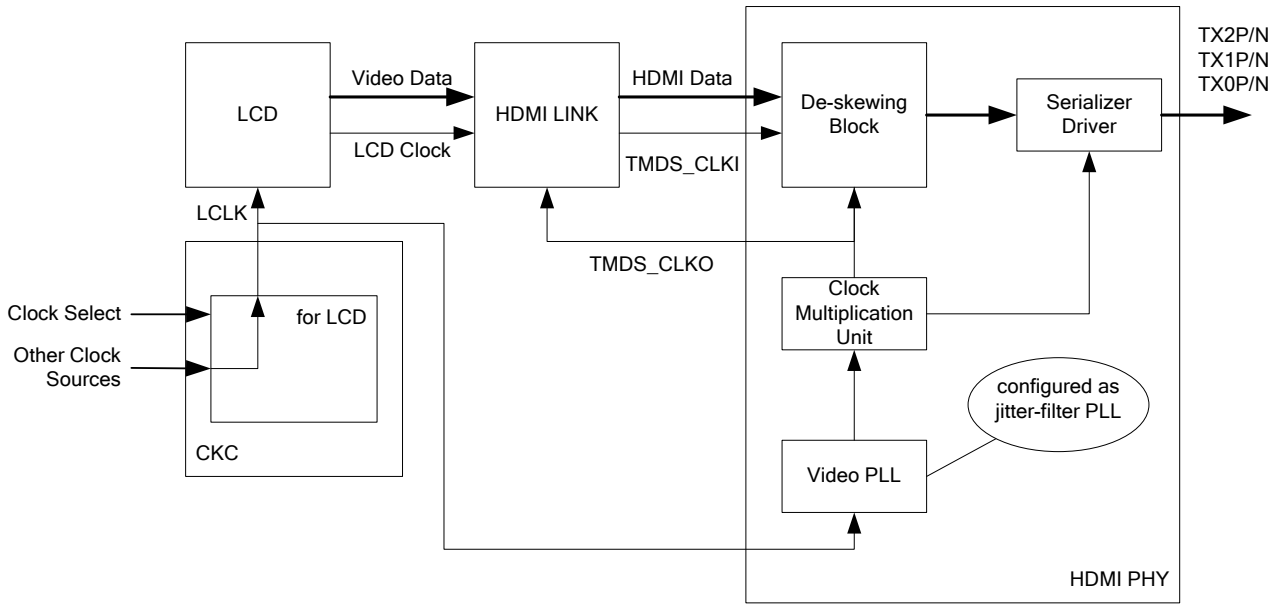


Figure 8.6 Clock Scheme Using an External Video PLL for Pixel Clock Generation (jitter-filter PLL)

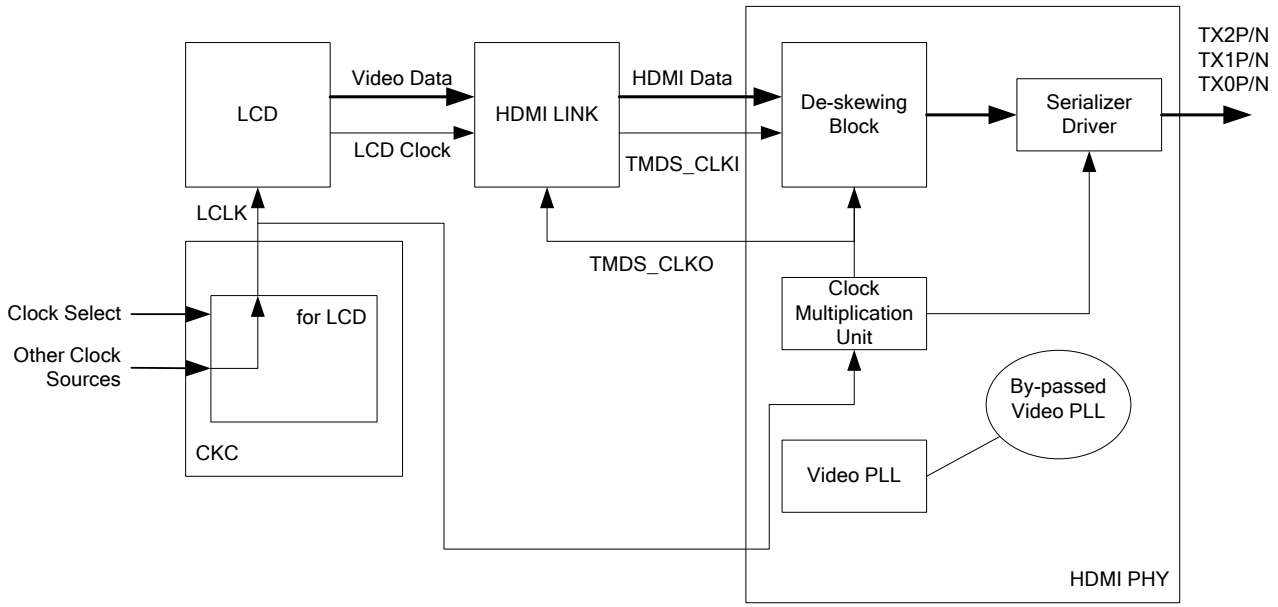


Figure 8.7 Clock Scheme Using an External Video PLL for Pixel Clock Generation (by-passed)

8.5.2 PHY Configuration Change through I2C

HDMI PHY has many internal registers to change its configuration, like pixel clock frequency or analog characteristics. These registers can be access through I2C. For secure configuration of the PHY core, MODE_SET_DONE register is used for an indicator of I2C setting state as shown in Figure 8.8.

(MODE_SET_DONE register is also controlled by I2C.)

To reconfigure PHY by new register setting, MODE_SET_DONE register should be set “0x00” instead of asserting overall RESET signal. Then PHY_READY signal goes to “0” state and PHY waits for new register setting. After new values are written on PHY registers, MODE_SET_DONE register should be set “0x80” again letting PHY start to configure its state with new register values. Once configuration is done, PHY_READY signal is automatically asserted.

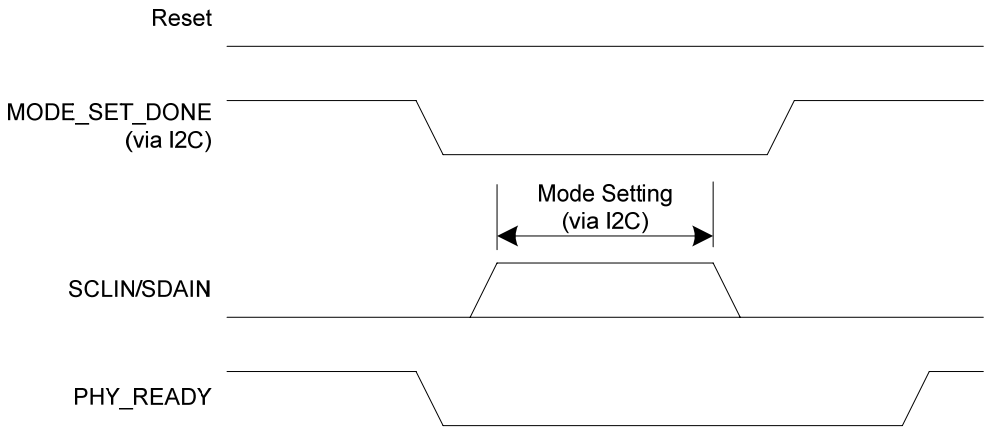


Figure 8.8 PHY Configuration through I2C with MODE_SET_DONE Register

8.5.3 PHY Reset Timing

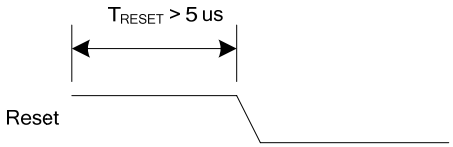


Figure 8.9 HDMI PHY Reset Timing

8.5.4 Register Map

Table 8.1 PHY Register Map (I2C Address = 0x70)

Name	Address	Description
BIAS	0x01	Bias Generation Selection
	0x02	Reserved
Video PLL	0x03	Pre-divider setting
	0x04	Pre-divider setting & Main divider setting
	0x05	Main divider setting
	0x06	Post-divider setting
	0x07	Post-divider setting
	0x08	Pixel clock divider setting
	0x09	PLL loop parameter setting
	0x0A	PLL loop parameter setting
	0x0B	PLL loop parameter setting
	0x0C	PLL loop parameter setting
	0x0D	Control code for sigma-delta modulator
	0x0E	Control code for sigma-delta modulator
	0x0F	Control code for automatic fre13huency calibration
	0x10	Control code for automatic fre13huency calibration
	0x11	Control code for automatic fre13huency calibration
Power Down	0x12	Power-down control
TX	0x13	TMDS data amplitude and pre-emphasis control
	0x14	TX source termination
Loop-back Test	0x15	Loop-back Test
CMU	0x16	Pre-divider & post-divider setting
	0x17	Main divider setting
	0x18	Control code for automatic fre13huency calibration
	0x19	Control code for automatic fre13huency

		calibration
	0x1A	Control code for automatic fre13huency calibration
	0x1B	Control code for automatic fre13huency calibration
	0x1C	PLL loop parameter setting
	0x1D	PLL loop parameter setting
	0x1E	PLL loop parameter setting
	0x1F	PLL loop parameter setting
	0x20	PLL loop parameter setting
RESET	0x21	RESET
Logic	0x22	PHY Logic
	0x23	PHY Logic
	0x24	PHY Logic
	0x25	PHY Logic
	0x26	PHY Logic
	0x27	PHY Logic
	0x28	MODE_SET_DONE

0x14 and 0x28 registers are special registers for TX source termination and I2C setting control, respectively. If TX Link layer found the SINK device can support TX source termination scheme, 0x14 register should be “60h”, which greatly reduces TX jitter in high-speed data transmission. For secure register setting between SOC and TX PHY through I2C channel, 0x28 register was used as I2C setting status monitor. If new register values are needed to be written, 0x28 register (MODE_SET_DONE) should be “00h” first to indicate start of register setting.

Then new register values are written through I2C. Finally, the 0x28 register should be set “80h” again for end of register setting.

8.5.5 Recommended Register Setting Value

8.5.5.1 Using the Integrated Video PLL for Pixel Clock Generation

	25.200	25.175	27.000	27.027	54.000	54.054	74.250	74.176
0x01	85h	85h	85h	85h	85h	85h	85h	85h
0x02	88h	88h	88h	88h	88h	88h	88h	88h
0x03	20h	20h	00h	00h	00h	00h	00h	00h
0x04	50h	50h	A0h	A0h	A0h	A0h	A0h	A0h
0x05	00h	00h	34h	34h	34h	34h	2Ch	2Ch
0x06	0Ch	0Ch	07h	07h	07h	07h	07h	07h
0x07	00h	00h	0Dh	0Dh	0Dh	0Dh	00h	00h
0x08	06h	06h	0Bh	0Bh	07h	07h	06h	06h
0x09	D8h	D8h	D8h	D8h	D8h	D8h	D8h	D8h
0x0A	C2h	C2h	C2h	C2h	C2h	C2h	C2h	C2h
0x0B	A4h	A4h	A4h	A4h	A4h	A4h	A4h	A4h
0x0C	46h	46h	06h	46h	06h	46h	06h	46h
0x0D	08h	07h	D8h	28h	D8h	28h	D8h	D8h
0x0E	0Ah	0Ah	5Bh	4Dh	5Bh	4Dh	5Bh	5Bh
0x0F	BBh	BBh	E2h	E3h	E2h	E3h	22h	23h
0x10	84h	84h	84h	84h	84h	84h	84h	84h
0x11	03h	03h	03h	03h	03h	03h	03h	03h
0x12	82h	82h	82h	82h	82h	82h	82h	82h
0x13	04h	04h	04h	04h	04h	04h	04h	04h
0x14	E0h	E0h	E0h	E0h	E0h	E0h	E0h	E0h
0x15	00h	00h	00h	00h	00h	00h	00h	00h
0x16	0Ch	0Ch	1Ch	1Ch	18h	18h	18h	18h
0x17	03h	03h	03h	03h	03h	03h	03h	03h
0x18	23h	23h	23h	23h	23h	23h	A3h	A3h
0x19	38h	38h	3Ch	3Ch	3Ch	3Ch	52h	52h
0x1A	38h	38h	38h	38h	38h	38h	38h	38h
0x1B	40h	40h	40h	40h	40h	40h	40h	40h
0x1C	88h	88h	88h	88h	88h	88h	88h	88h
0x1D	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch
0x1E	09h	09h	09h	09h	09h	09h	09h	09h
0x1F	13h	13h	13h	13h	13h	13h	13h	13h
0x20	06h	06h	06h	06h	06h	06h	06h	06h
0x21	00h	00h	00h	00h	00h	00h	00h	00h
0x22	54h	54h	54h	54h	54h	54h	54h	54h
0x23	24h	24h	FFh	FEh	FFh	FFh	B9h	B9h
0x24	96h	96h	55h	15h	54h	14h	54h	54h
0x25	6Dh	6Dh	66h	66h	66h	66h	4Ah	4Ah
0x26	02h	02h	02h	02h	02h	02h	02h	02h
0x27	00h	00h	00h	00h	00h	00h	00h	00h
0x28	80h	80h	80h	80h	80h	80h	80h	80h

	148.500	148.352	108.108	72.000	25.000	65.000	108.000	162.000
0x01	85h	85h	85h	85h	85h	85h	85h	85h
0x02	88h	88h	88h	88h	88h	88h	88h	88h
0x03	00h	00h	00h	00h	00h	40h	00h	00h
0x04	A0h	A0h	A0h	A0h	30h	A0h	A0h	10h

0x05	2Ch	2Ch	34h	30h	00h	82h	34h	00h
0x06	07h	07h	07h	0Bh	0Ch	0Bh	07h	06h
0x07	00h	00h	0Dh	00h	00h	00h	0Dh	00h
0x08	02h	02h	03h	02h	06h	02h	03h	00h
0x09	D8h	D8h	D8h	D8h	D8h	D8h	D8h	D8h
0x0A	C2h	C2h	C2h	C2h	C2h	C2h	C2h	C2h
0x0B	A4h	A4h	A4h	A4h	A4h	A4h	A4h	A4h
0x0C	06h	46h	46h	06h	06h	06h	06h	06h
0x0D	D8h	D8h	28h	D8h	D8h	D8h	D8h	D8h
0x0E	5Bh	5Bh	4Dh	5Bh	5Bh	5Bh	5Bh	5Bh
0x0F	22h	23h	E3h	82h	B2h	12h	E2h	F2h
0x10	84h	84h	84h	84h	84h	84h	84h	83h
0x11	03h	03h	03h	03h	03h	03h	03h	03h
0x12	82h	82h	82h	82h	82h	82h	82h	82h
0x13	04h	04h	04h	04h	04h	04h	04h	04h
0x14	E0h	E0h	E0h	E0h	E0h	E0h	E0h	E0h
0x15	00h	00h	00h	00h	00h	00h	00h	00h
0x16	14h	14h	14h	08h	0Ch	08h	14h	10h
0x17	03h	03h	03h	03h	03h	03h	03h	00h
0x18	A3h	A3h	23h	23h	A3h	63h	23h	23h
0x19	52h	52h	3Ch	50h	37h	48h	3Ch	2Dh
0x1A	38h	38h	38h	38h	38h	38h	38h	38h
0x1B	40h	40h	40h	40h	40h	40h	40h	40h
0x1C	88h	88h	88h	88h	88h	88h	88h	88h
0x1D	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch
0x1E	09h	09h	09h	09h	09h	09h	09h	09h
0x1F	13h	13h	13h	13h	13h	13h	13h	13h
0x20	06h	06h	06h	06h	06h	06h	06h	06h
0x21	00h	00h	00h	00h	00h	00h	00h	00h
0x22	54h	54h	54h	54h	54h	54h	54h	54h
0x23	5Ch	5Ch	7Fh	BFh	28h	D4h	7Fh	54h
0x24	54h	54h	14h	94h	56h	D4h	54h	54h
0x25	4Ah	4Ah	66h	4Ch	6Eh	54h	66h	88h
0x26	02h	02h	02h	02h	02h	02h	02h	02h
0x27	00h	00h	00h	00h	00h	00h	00h	00h
0x28	80h	80h	80h	80h	80h	80h	80h	80h

8.5.5.2 Using an External Video PLL for Pixel Clock Generation and Configuring the Integrated Video PLL as a jitter-filter PLL

	25.200	**25.175**	**27.000**	**27.027**	**54.000**	**54.054**	**74.250**	**74.176**
0x01	85h	85h	85h	85h	85h	85h	85h	85h
0x02	88h	88h	88h	88h	88h	88h	88h	88h
0x03	00h	00h	00h	00h	20h	20h	20h	20h
0x04	40h	40h	A0h	A0h	A0h	A0h	00h	00h
0x05	00h	00h	34h	34h	34h	34h	00h	00h
0x06	0Ch	0Ch	07h	07h	07h	07h	07h	07h
0x07	00h	00h	0Dh	0Dh	0Dh	0Dh	00h	00h
0x08	06h	06h	0Bh	0Bh	07h	07h	06h	06h
0x09	D8h	D8h	D8h	D8h	D8h	D8h	D8h	D8h
0x0A	E2h	E2h	E2h	E2h	E2h	E2h	E2h	E2h

0x0B	A4h	A4h	A4h	A4h	A4h	A4h	A4h	A4h
0x0C	06h	06h	06h	06h	06h	06h	06h	06h
0x0D	D8h	D8h	D8h	D8h	D8h	D8h	D8h	D8h
0x0E	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh
0x0F	BAh	BAh	E2h	E2h	E2h	E2h	22h	22h
0x10	84h	84h	84h	84h	84h	84h	84h	84h
0x11	03h	03h	03h	03h	03h	03h	03h	03h
0x12	82h	82h	82h	82h	82h	82h	82h	82h
0x13	04h	04h	04h	04h	04h	04h	04h	04h
0x14	E0h	E0h	E0h	E0h	E0h	E0h	E0h	E0h
0x15	00h	00h	00h	00h	00h	00h	00h	00h
0x16	0Ch	0Ch	1Ch	1Ch	18h	18h	18h	18h
0x17	03h	03h	03h	03h	03h	03h	03h	03h
0x18	23h	23h	23h	23h	23h	23h	A3h	A3h
0x19	38h	38h	3Ch	3Ch	3Ch	3Ch	52h	52h
0x1A	38h	38h	38h	38h	38h	38h	38h	38h
0x1B	40h	40h	40h	40h	40h	40h	40h	40h
0x1C	88h	88h	88h	88h	88h	88h	88h	88h
0x1D	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch
0x1E	09h	09h	09h	09h	09h	09h	09h	09h
0x1F	13h	13h	13h	13h	13h	13h	13h	13h
0x20	06h	06h	06h	06h	06h	06h	06h	06h
0x21	00h	00h	00h	00h	00h	00h	00h	00h
0x22	54h	54h	54h	54h	54h	54h	54h	54h
0x23	24h	24h	FFh	FEh	FFh	FFh	B9h	B9h
0x24	96h	96h	55h	15h	54h	14h	54h	54h
0x25	6Dh	6Dh	66h	66h	66h	66h	4Ah	4Ah
0x26	02h	02h	02h	02h	02h	02h	02h	02h
0x27	00h	00h	00h	00h	00h	00h	00h	00h
0x28	80h	80h	80h	80h	80h	80h	80h	80h

	148.500	**148.352**	**108.108**	**72.000**	**25.000**	**65.000**	**108.000**	**162.000**
0x01	85h	85h	85h	85h	85h	85h	85h	85h
0x02	88h	88h	88h	88h	88h	88h	88h	88h
0x03	60h	60h	60h	40h	00h	40h	60h	60h
0x04	00h	00h	A0h	40h	40h	40h	A0h	00h
0x05	00h	00h	34h	30h	00h	00h	34h	00h
0x06	07h	07h	07h	0Bh	0Ch	0Bh	07h	07h
0x07	00h	00h	0Dh	00h	00h	00h	0Dh	00h
0x08	02h	02h	03h	02h	06h	02h	03h	02h
0x09	D8h	D8h	D8h	D8h	D8h	D8h	D8h	D8h
0x0A	E2h	E2h	E2h	E2h	E2h	E2h	E2h	E2h
0x0B	A4h	A4h	A4h	A4h	A4h	A4h	A4h	A4h
0x0C	06h	06h	06h	06h	06h	06h	06h	06h
0x0D	D8h	D8h	D8h	D8h	D8h	D8h	D8h	D8h
0x0E	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh
0x0F	22h	22h	E2h	82h	B2h	12h	E2h	82h
0x10	84h	84h	84h	84h	84h	84h	84h	84h
0x11	03h	03h	03h	03h	03h	03h	03h	03h
0x12	82h	82h	82h	82h	82h	82h	82h	82h
0x13	04h	04h	04h	04h	04h	04h	04h	04h

0x14	E0h	E0h	E0h	E0h	E0h	E0h	E0h	E0h
0x15	00h	00h	00h	00h	00h	00h	00h	00h
0x16	14h	14h	14h	08h	0Ch	08h	14h	10h
0x17	03h	03h	03h	03h	03h	03h	03h	00h
0x18	A3h	A3h	23h	23h	A3h	63h	23h	23h
0x19	52h	52h	3Ch	50h	37h	48h	3Ch	2Dh
0x1A	38h	38h	38h	38h	38h	38h	38h	38h
0x1B	40h	40h	40h	40h	40h	40h	40h	40h
0x1C	88h	88h	88h	88h	88h	88h	88h	88h
0x1D	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch
0x1E	09h	09h	09h	09h	09h	09h	09h	09h
0x1F	13h	13h	13h	13h	13h	13h	13h	13h
0x20	06h	06h	06h	06h	06h	06h	06h	06h
0x21	00h	00h	00h	00h	00h	00h	00h	00h
0x22	54h	54h	54h	54h	54h	54h	54h	54h
0x23	5Ch	5Ch	7Fh	BFh	28h	D4h	7Fh	54h
0x24	54h	54h	14h	94h	56h	D4h	54h	54h
0x25	4Ah	4Ah	66h	4Ch	6Eh	54h	66h	88h
0x26	02h	02h	02h	02h	02h	02h	02h	02h
0x27	00h	00h	00h	00h	00h	00h	00h	00h
0x28	80h	80h	80h	80h	80h	80h	80h	80h

8.5.5.3 Using External Video PLL for Pixel Clock Generation and By-passing Internal Video PLL

	25.200	25.175	27.000	27.027	54.000	54.054	74.250	74.176
0x01	85h	85h	85h	85h	85h	85h	85h	85h
0x02	88h	88h	88h	88h	88h	88h	88h	88h
0x03	00h	00h	00h	00h	20h	20h	20h	20h
0x04	40h	40h	A0h	A0h	A0h	A0h	00h	00h
0x05	00h	00h	34h	34h	34h	34h	00h	00h
0x06	0Ch	0Ch	07h	07h	07h	07h	07h	07h
0x07	00h	00h	0Dh	0Dh	0Dh	0Dh	00h	00h
0x08	02h	02h	03h	03h	03h	03h	02h	02h
0x09	D8h	D8h	D8h	D8h	D8h	D8h	D8h	D8h
0x0A	E2h	E2h	E2h	E2h	E2h	E2h	E2h	E2h
0x0B	24h	24h	24h	24h	24h	24h	24h	24h
0x0C	06h	06h	06h	06h	06h	06h	06h	06h
0x0D	D8h	D8h	D8h	D8h	D8h	D8h	D8h	D8h
0x0E	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh
0x0F	B8h	B8h	E0h	E0h	E0h	E0h	20h	20h
0x10	84h	84h	84h	84h	84h	84h	84h	84h
0x11	03h	03h	03h	03h	03h	03h	03h	03h
0x12	92h	92h	92h	92h	92h	92h	92h	92h
0x13	04h	04h	04h	04h	04h	04h	04h	04h
0x14	E0h	E0h	E0h	E0h	E0h	E0h	E0h	E0h
0x15	00h	00h	00h	00h	00h	00h	00h	00h
0x16	0Ch	0Ch	0Ch	0Ch	08h	08h	08h	08h
0x17	50h	50h	50h	50h	03h	03h	03h	03h
0x18	23h	23h	23h	23h	23h	23h	A3h	A3h
0x19	38h	38h	3Ch	3Ch	3Ch	3Ch	52h	52h
0x1A	38h	38h	38h	38h	38h	38h	38h	38h

0x1B	40h	40h	40h	40h	40h	40h	40h	40h
0x1C	88h	88h	88h	88h	88h	88h	88h	88h
0x1D	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch
0x1E	09h	09h	09h	09h	09h	09h	09h	09h
0x1F	13h	13h	13h	13h	13h	13h	13h	13h
0x20	06h	06h	06h	06h	06h	06h	06h	06h
0x21	00h	00h	00h	00h	00h	00h	00h	00h
0x22	54h	54h	54h	54h	54h	54h	54h	54h
0x23	24h	24h	FFh	FEh	FFh	FFh	B9h	B9h
0x24	96h	96h	55h	15h	54h	14h	54h	54h
0x25	6Dh	6Dh	66h	66h	66h	66h	4Ah	4Ah
0x26	02h	02h	02h	02h	02h	02h	02h	02h
0x27	00h	00h	00h	00h	00h	00h	00h	00h
0x28	80h	80h	80h	80h	80h	80h	80h	80h

	148.500	**148.352**	**108.108**	**72.000**	**25.000**	**65.000**	**108.000**	**162.000**
0x01	85h	85h	85h	85h	85h	85h	85h	85h
0x02	88h	88h	88h	88h	88h	88h	88h	88h
0x03	60h	60h	60h	40h	00h	40h	60h	60h
0x04	00h	00h	A0h	40h	40h	40h	A0h	00h
0x05	00h	00h	34h	30h	00h	00h	34h	00h
0x06	07h	07h	07h	0Bh	0Ch	0Bh	07h	07h
0x07	00h	00h	0Dh	00h	00h	00h	0Dh	00h
0x08	02h	02h	03h	02h	02h	02h	03h	02h
0x09	D8h	D8h	D8h	D8h	D8h	D8h	D8h	D8h
0x0A	E2h	E2h	E2h	E2h	E2h	E2h	E2h	E2h
0x0B	24h	24h	24h	24h	24h	24h	24h	24h
0x0C	06h	06h	06h	06h	06h	06h	06h	06h
0x0D	D8h	D8h	D8h	D8h	D8h	D8h	D8h	D8h
0x0E	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh	5Bh
0x0F	20h	20h	E0h	80h	B0h	10h	E0h	80h
0x10	84h	84h	84h	84h	84h	84h	84h	84h
0x11	03h	03h	03h	03h	03h	03h	03h	03h
0x12	92h	92h	92h	92h	92h	92h	92h	92h
0x13	04h	04h	04h	04h	04h	04h	04h	04h
0x14	E0h	E0h	E0h	E0h	E0h	E0h	E0h	E0h
0x15	00h	00h	00h	00h	00h	00h	00h	00h
0x16	14h	14h	14h	08h	0Ch	08h	14h	10h
0x17	03h	03h	03h	03h	50h	03h	03h	00h
0x18	A3h	A3h	23h	23h	A3h	63h	23h	23h
0x19	52h	52h	3Ch	50h	37h	48h	3Ch	2Dh
0x1A	38h	38h	38h	38h	38h	38h	38h	38h
0x1B	40h	40h	40h	40h	40h	40h	40h	40h
0x1C	88h	88h	88h	88h	88h	88h	88h	88h
0x1D	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch	7Ch
0x1E	09h	09h	09h	09h	09h	09h	09h	09h
0x1F	13h	13h	13h	13h	13h	13h	13h	13h
0x20	06h	06h	06h	06h	06h	06h	06h	06h
0x21	00h	00h	00h	00h	00h	00h	00h	00h
0x22	54h	54h	54h	54h	54h	54h	54h	54h
0x23	5Ch	5Ch	7Fh	BFh	28h	D4h	7Fh	54h

0x24	54h	54h	14h	94h	56h	D4h	54h	54h
0x25	4Ah	4Ah	66h	4Ch	6Eh	54h	66h	88h
0x26	02h	02h	02h	02h	02h	02h	02h	02h
0x27	00h	00h	00h	00h	00h	00h	00h	00h
0x28	80h	80h	80h	80h	80h	80h	80h	80h

8.6 Examples - HDMI Register Setting

When HDMI is used, an outputted clock from HDMI PHY is recommended to be used for LCD Clock. All the example values below take HDMI PHY PLL. The below values are recommended for LCD setting value, HDMI PHY, HDMI Link according to video output format.

8.6.1 1920x1080p

1920x1080p @ 60Hz								
Pixel Clock		148.5	MHz					
H Frequency		67.5	KHz					
V Frequency		60	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	280		IV	0
0x01	85h	0x15	00h	V2_BLANK	1125		IH	0
0x02	88h	0x16	14h	V1_BLANK	45		IP	0
0x03	00h	0x17	03h	V_LINE	1125	LCTRL	DP	0
0x04	A0h	0x18	A3h	H_LINE	2200		NI	1
0x05	2Ch	0x19	52h	VSYNC_POL	0		TV	0
0x06	07h	0x1A	38h	INT_PRO_MODE	0		TFT	1
0x07	00h	0x1B	40h	V_BOT_ST	-		STN	0
0x08	02h	0x1C	88h	V_BOT_END	-	LHTIME1	LPW	43
0x09	D8h	0x1D	7Ch	HSYNC_START	86		LPC	1919
0x0A	C2h	0x1E	09h	HSYNC_END	130	LHTIME2	LSWC	147
0x0B	A4h	0x1F	13h	HSYNC_POL	0		LEWC	87
0x0C	06h	0x20	06h	VSYNC_T_END	9		VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	4	LVTIME1	VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	-		FPW	4
0x0F	22h	0x23	5Ch	VSYNC_B_ST	-		FLC	1079
0x10	84h	0x24	54h	VSYNC_H_POS_ST	-	LVTIME2	FSWC	35
0x11	03h	0x25	4Ah	VSYNC_H_POS_END	-		FEWC	3
0x12	82h	0x26	02h			LVTIME3	FPW2	4
0x13	04h	0x27	00h				FLC2	1079
0x14	E0h	0x28	80h			LVTIME4	FSWC2	35
							FEWC2	3

1920x1080p @ 59.94Hz								
Pixel Clock		148.35	MHz					
H Frequency		67.432	KHz					
V Frequency		59.939	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	280	LCTRL	IV	0

0x01	85h	0x15	00h	V2_BLANK	1125		IH	0
0x02	88h	0x16	14h	V1_BLANK	45		IP	0
0x03	00h	0x17	03h	V_LINE	1125		DP	0
0x04	A0h	0x18	A3h	H_LINE	2200		NI	1
0x05	2Ch	0x19	52h	VSYNC_POL	0		TV	0
0x06	07h	0x1A	38h	INT_PRO_MODE	0		TFT	1
0x07	00h	0x1B	40h	V_BOT_ST	-		STN	0
0x08	02h	0x1C	88h	V_BOT_END	-	LHTIME1	LPW	43
0x09	D8h	0x1D	7Ch	HSYNC_START	86		LPC	1919
0x0A	C2h	0x1E	09h	HSYNC_END	130	LHTIME2	LSWC	147
0x0B	A4h	0x1F	13h	HSYNC_POL	0		LEWC	87
0x0C	46h	0x20	06h	VSYNC_T_END	9	LVTIME1	VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	4		VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	-		FPW	4
0x0F	23h	0x23	5Ch	VSYNC_B_ST	-		FLC	1079
0x10	84h	0x24	54h	VSYNC_H_POS_ST	-	LVTIME2	FSWC	35
0x11	03h	0x25	4Ah	VSYNC_H_POS_END	-		FEWC	3
0x12	82h	0x26	02h			LVTIME3	FPW2	4
0x13	04h	0x27	00h				FLC2	1079
0x14	E0h	0x28	80h			LVTIME4	FSWC2	35
							FEWC2	3

1920x1080p @ 50Hz								
Pixel Clock		148.5	MHz					
H Frequency		56.25	KHz					
V Frequency		50	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	720		IV	0
0x01	85h	0x15	00h	V2_BLANK	1125		IH	0
0x02	88h	0x16	14h	V1_BLANK	45		IP	0
0x03	00h	0x17	03h	V_LINE	1125		DP	0
0x04	A0h	0x18	A3h	H_LINE	2640	LCTRL	NI	1
0x05	2Ch	0x19	52h	VSYNC_POL	0		TV	0
0x06	07h	0x1A	38h	INT_PRO_MODE	0		TFT	1
0x07	00h	0x1B	40h	V_BOT_ST	-		STN	0
0x08	02h	0x1C	88h	V_BOT_END	-	LHTIME1	LPW	43
0x09	D8h	0x1D	7Ch	HSYNC_START	526		LPC	1919
0x0A	C2h	0x1E	09h	HSYNC_END	570	LHTIME2	LSWC	163
0x0B	A4h	0x1F	13h	HSYNC_POL	0		LEWC	511
0x0C	46h	0x20	06h	VSYNC_T_END	9	LVTIME1	VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	4		VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	-		FPW	4
0x0F	22h	0x23	5Ch	VSYNC_B_ST	-		FLC	1079
0x10	84h	0x24	54h	VSYNC_H_POS_ST	-	LVTIME2	FSWC	35
0x11	03h	0x25	4Ah	VSYNC_H_POS_END	-		FEWC	3
0x12	82h	0x26	02h			LVTIME3	FPW2	4
0x13	04h	0x27	00h				FLC2	1079
0x14	E0h	0x28	80h			LVTIME4	FSWC2	35

							FEWC2	3

8.6.2 1920x1080i

1920x1080i @ 60Hz								
Pixel Clock		74.25	MHz					
H Frequency		33.75	KHz					
V Frequency		60	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	280	LCTRL	IV	0
0x01	85h	0x15	00h	V2_BLANK	562		IH	0
0x02	88h	0x16	18h	V1_BLANK	22		IP	0
0x03	00h	0x17	03h	V_LINE	1125		DP	0
0x04	A0h	0x18	A3h	H_LINE	2200		NI	0
0x05	2Ch	0x19	52h	VSYNC_POL	0		TV	1
0x06	07h	0x1A	38h	INT_PRO_MODE	1		TFT	0
0x07	00h	0x1B	40h	V_BOT_ST	585		STN	0
0x08	06h	0x1C	88h	V_BOT_END	1125	LHTIME1	LPW	43
0x09	D8h	0x1D	7Ch	HSYNC_START	86		LPC	1919
0x0A	C2h	0x1E	09h	HSYNC_END	130	LHTIME2	LSWC	147
0x0B	A4h	0x1F	13h	HSYNC_POL	0		LEWC	87
0x0C	06h	0x20	06h	VSYNC_T_END	7	LVTIME1	VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	2		VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	569		FPW	9
0x0F	22h	0x23	B9h	VSYNC_B_ST	564		FLC	1079
0x10	84h	0x24	54h	VSYNC_H_POS_ST	1188	LVTIME2	FSWC	29
0x11	03h	0x25	4Ah	VSYNC_H_POS_END	1188		FEWC	3
0x12	82h	0x26	02h			LVTIME3	FPW2	9
0x13	04h	0x27	00h				FLC2	1079
0x14	E0h	0x28	80h			LVTIME4	FSWC2	31
							FEWC2	3

1920x1080i @ 59.94Hz								
Pixel Clock		74.175	MHz					
H Frequency		33.716	KHz					
V Frequency		59.939	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	280	LCTRL	IV	0
0x01	85h	0x15	00h	V2_BLANK	562		IH	0
0x02	88h	0x16	18h	V1_BLANK	22		IP	0
0x03	00h	0x17	03h	V_LINE	1125		DP	0
0x04	A0h	0x18	A3h	H_LINE	2200		NI	0
0x05	2Ch	0x19	52h	VSYNC_POL	0		TV	1
0x06	07h	0x1A	38h	INT_PRO_MODE	1		TFT	0
0x07	00h	0x1B	40h	V_BOT_ST	585		STN	0

0x08	06h	0x1C	88h	V_BOT_END	1125	LHTIME1	LPW	43
0x09	D8h	0x1D	7Ch	HSYNC_START	86		LPC	1919
0x0A	C2h	0x1E	09h	HSYNC_END	130	LHTIME2	LSWC	147
0x0B	A4h	0x1F	13h	HSYNC_POL	0		LEWC	87
0x0C	46h	0x20	06h	VSYNC_T_END	7	LVTIME1	VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	2		VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	569		FPW	9
0x0F	23h	0x23	B9h	VSYNC_B_ST	564		FLC	1079
0x10	84h	0x24	54h	VSYNC_H_POS_ST	1188	LVTIME2	FSWC	29
0x11	03h	0x25	4Ah	VSYNC_H_POS_END	1188		FEWC	3
0x12	82h	0x26	02h			LVTIME3	FPW2	9
0x13	04h	0x27	00h				FLC2	1079
0x14	E0h	0x28	80h			LVTIME4	FSWC2	31
							FEWC2	3

1920x1080i @ 50Hz								
Pixel Clock		74.25	MHz					
H Frequency		28.125	KHz					
V Frequency		50	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	720	LCTRL	IV	0
0x01	85h	0x15	00h	V2_BLANK	562		IH	0
0x02	88h	0x16	18h	V1_BLANK	22		IP	0
0x03	00h	0x17	03h	V_LINE	1125		DP	0
0x04	A0h	0x18	A3h	H_LINE	2640		NI	0
0x05	2Ch	0x19	52h	VSYNC_POL	0		TV	1
0x06	07h	0x1A	38h	INT_PRO_MODE	1		TFT	0
0x07	00h	0x1B	40h	V_BOT_ST	585		STN	0
0x08	06h	0x1C	88h	V_BOT_END	1125	LHTIME1	LPW	43
0x09	D8h	0x1D	7Ch	HSYNC_START	526		LPC	1919
0x0A	C2h	0x1E	09h	HSYNC_END	570	LHTIME2	LSWC	163
0x0B	A4h	0x1F	13h	HSYNC_POL	0		LEWC	511
0x0C	06h	0x20	06h	VSYNC_T_END	7	LVTIME1	VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	2		VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	569		FPW	9
0x0F	22h	0x23	B9h	VSYNC_B_ST	564		FLC	1079
0x10	84h	0x24	54h	VSYNC_H_POS_ST	1848	LVTIME2	FSWC	29
0x11	03h	0x25	4Ah	VSYNC_H_POS_END	1848		FEWC	3
0x12	82h	0x26	02h			LVTIME3	FPW2	9
0x13	04h	0x27	00h				FLC2	1079
0x14	E0h	0x28	80h			LVTIME4	FSWC2	31
							FEWC2	3

8.6.3 1280x720p

1280x720p @ 60Hz								

Pixel Clock		74.25	MHz					
H Frequency		45	KHz					
V Frequency		60	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	370		IV	0
0x01	85h	0x15	00h	V2_BLANK	750		IH	0
0x02	88h	0x16	18h	V1_BLANK	30		IP	0
0x03	00h	0x17	03h	V_LINE	750	LCTRL	DP	0
0x04	A0h	0x18	A3h	H_LINE	1650		NI	1
0x05	2Ch	0x19	52h	VSYNC_POL	0		TV	0
0x06	07h	0x1A	38h	INT_PRO_MODE	0		TFT	1
0x07	00h	0x1B	40h	V_BOT_ST	-		STN	0
0x08	06h	0x1C	88h	V_BOT_END	-	LHTIME1	LPW	39
0x09	D8h	0x1D	7Ch	HSYNC_START	108		LPC	1279
0x0A	C2h	0x1E	09h	HSYNC_END	148	LHTIME2	LSWC	219
0x0B	A4h	0x1F	13h	HSYNC_POL	0		LEWC	109
0x0C	06h	0x20	06h	VSYNC_T_END	10		VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	5	LVTIME1	VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	-		FPW	4
0x0F	22h	0x23	B9h	VSYNC_B_ST	-		FLC	719
0x10	84h	0x24	54h	VSYNC_H_POS_ST	-	LVTIME2	FSWC	19
0x11	03h	0x25	4Ah	VSYNC_H_POS_END	-		FEWC	4
0x12	82h	0x26	02h			LVTIME3	FPW2	4
0x13	04h	0x27	00h				FLC2	719
0x14	E0h	0x28	80h			LVTIME4	FSWC2	19
							FEWC2	4

1280x720p @ 59.94Hz								
Pixel Clock		74.175	MHz					
H Frequency		44.955	KHz					
V Frequency		59.939	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	370		IV	0
0x01	85h	0x15	00h	V2_BLANK	750		IH	0
0x02	88h	0x16	18h	V1_BLANK	30		IP	0
0x03	00h	0x17	03h	V_LINE	750	LCTRL	DP	0
0x04	A0h	0x18	A3h	H_LINE	1650		NI	1
0x05	2Ch	0x19	52h	VSYNC_POL	0		TV	0
0x06	07h	0x1A	38h	INT_PRO_MODE	0		TFT	1
0x07	00h	0x1B	40h	V_BOT_ST	-		STN	0
0x08	06h	0x1C	88h	V_BOT_END	-	LHTIME1	LPW	39
0x09	D8h	0x1D	7Ch	HSYNC_START	108		LPC	1279
0x0A	C2h	0x1E	09h	HSYNC_END	148	LHTIME2	LSWC	219
0x0B	A4h	0x1F	13h	HSYNC_POL	0		LEWC	109
0x0C	46h	0x20	06h	VSYNC_T_END	10	LVTIME1	VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	5		VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	-		FPW	4

0x0F	23h	0x23	B9h	VSYNC_B_ST	-		FLC	719
0x10	84h	0x24	54h	VSYNC_H_POS_ST	-	LVTIME2	FSWC	19
0x11	03h	0x25	4Ah	VSYNC_H_POS_END	-		FEWC	4
0x12	82h	0x26	02h			LVTIME3	FPW2	4
0x13	04h	0x27	00h				FLC2	719
0x14	E0h	0x28	80h			LVTIME4	FSWC2	19
							FEWC2	4

1280x720p @ 50Hz								
Pixel Clock		74.25	MHz					
H Frequency		37.5	KHz					
V Frequency		50	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	700		IV	0
0x01	85h	0x15	00h	V2_BLANK	750		IH	0
0x02	88h	0x16	18h	V1_BLANK	30		IP	0
0x03	00h	0x17	03h	V_LINE	750	LCTRL	DP	0
0x04	A0h	0x18	A3h	H_LINE	1980		NI	1
0x05	2Ch	0x19	52h	VSYNC_POL	0		TV	0
0x06	07h	0x1A	38h	INT_PRO_MODE	0		TFT	1
0x07	00h	0x1B	40h	V_BOT_ST	-		STN	0
0x08	06h	0x1C	88h	V_BOT_END	-	LHTIME1	LPW	39
0x09	D8h	0x1D	7Ch	HSYNC_START	438		LPC	1279
0x0A	C2h	0x1E	09h	HSYNC_END	478	LHTIME2	LSWC	219
0x0B	A4h	0x1F	13h	HSYNC_POL	0		LEWC	439
0x0C	06h	0x20	06h	VSYNC_T_END	10		VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	5	LVTIME1	VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	-		FPW	4
0x0F	22h	0x23	B9h	VSYNC_B_ST	-		FLC	719
0x10	84h	0x24	54h	VSYNC_H_POS_ST	-	LVTIME2	FSWC	19
0x11	03h	0x25	4Ah	VSYNC_H_POS_END	-		FEWC	4
0x12	82h	0x26	02h			LVTIME3	FPW2	4
0x13	04h	0x27	00h				FLC2	719
0x14	E0h	0x28	80h			LVTIME4	FSWC2	19
							FEWC2	4

8.6.4 720x480p

720x480p @ 60Hz								
Pixel Clock		27.027	MHz					
H Frequency		31.5	KHz					
V Frequency		60	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	138	LCTRL	IV	1
0x01	85h	0x15	00h	V2_BLANK	525		IH	1

0x02	88h	0x16	1Ch	V1_BLANK	45		IP	0
0x03	00h	0x17	03h	V_LINE	525		DP	0
0x04	A0h	0x18	23h	H_LINE	858		NI	1
0x05	34h	0x19	3Ch	VSYNC_POL	1		TV	0
0x06	07h	0x1A	38h	INT_PRO_MODE	0		TFT	1
0x07	0Dh	0x1B	40h	V_BOT_ST	-		STN	0
0x08	0Bh	0x1C	88h	V_BOT_END	-	LHTIME1	LPW	61
0x09	D8h	0x1D	7Ch	HSYNC_START	14		LPC	719
0x0A	C2h	0x1E	09h	HSYNC_END	76	LHTIME2	LSWC	59
0x0B	A4h	0x1F	13h	HSYNC_POL	1		LEWC	15
0x0C	46h	0x20	06h	VSYNC_T_END	15	LVTIME1	VDB	0
0x0D	28h	0x21	00h	VSYNC_T_ST	9		VDF	0
0x0E	4Dh	0x22	54h	VSYNC_B_END	-		FPW	5
0x0F	E3h	0x23	FEh	VSYNC_B_ST	-		FLC	479
0x10	84h	0x24	15h	VSYNC_H_POS_ST	-	LVTIME2	FSWC	29
0x11	03h	0x25	66h	VSYNC_H_POS_END	-		FEWC	8
0x12	82h	0x26	02h			LVTIME3	FPW2	5
0x13	04h	0x27	00h				FLC2	479
0x14	E0h	0x28	80h			LVTIME4	FSWC2	29
							FEWC2	8

720x480p @ 59.94Hz								
Pixel Clock		27	MHz					
H Frequency		31.469	KHz					
V Frequency		59.94	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	138		IV	1
0x01	85h	0x15	00h	V2_BLANK	525		IH	1
0x02	88h	0x16	1Ch	V1_BLANK	45		IP	0
0x03	00h	0x17	03h	V_LINE	525		DP	0
0x04	A0h	0x18	23h	H_LINE	858	LCTRL	NI	1
0x05	34h	0x19	3Ch	VSYNC_POL	1		TV	0
0x06	07h	0x1A	38h	INT_PRO_MODE	0		TFT	1
0x07	0Dh	0x1B	40h	V_BOT_ST	-		STN	0
0x08	0Bh	0x1C	88h	V_BOT_END	-	LHTIME1	LPW	61
0x09	D8h	0x1D	7Ch	HSYNC_START	14		LPC	719
0x0A	C2h	0x1E	09h	HSYNC_END	76	LHTIME2	LSWC	59
0x0B	A4h	0x1F	13h	HSYNC_POL	1		LEWC	15
0x0C	06h	0x20	06h	VSYNC_T_END	15	LVTIME1	VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	9		VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	-		FPW	5
0x0F	E2h	0x23	FFh	VSYNC_B_ST	-		FLC	479
0x10	84h	0x24	55h	VSYNC_H_POS_ST	-	LVTIME2	FSWC	29
0x11	03h	0x25	66h	VSYNC_H_POS_END	-		FEWC	8
0x12	82h	0x26	02h			LVTIME3	FPW2	5
0x13	04h	0x27	00h				FLC2	479
0x14	E0h	0x28	80h			LVTIME4	FSWC2	29
							FEWC2	8

8.6.5 720x576p

720x576p @ 50Hz								
Pixel Clock		27	MHz					
H Frequency		31.25	KHz					
V Frequency		50	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	144	LCTRL	IV	1
0x01	85h	0x15	00h	V2_BLANK	625		IH	1
0x02	88h	0x16	1Ch	V1_BLANK	49		IP	0
0x03	00h	0x17	03h	V_LINE	625		DP	0
0x04	A0h	0x18	23h	H_LINE	864		NI	1
0x05	34h	0x19	3Ch	VSYNC_POL	1		TV	0
0x06	07h	0x1A	38h	INT_PRO_MODE	0		TFT	1
0x07	0Dh	0x1B	40h	V_BOT_ST	-		STN	0
0x08	0Bh	0x1C	88h	V_BOT_END	-	LHTIME1	LPW	63
0x09	D8h	0x1D	7Ch	HSYNC_START	10		LPC	719
0x0A	C2h	0x1E	09h	HSYNC_END	74	LHTIME2	LSWC	67
0x0B	A4h	0x1F	13h	HSYNC_POL	1		LEWC	11
0x0C	06h	0x20	06h	VSYNC_T_END	10	LVTIME1	VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	5		VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	-		FPW	4
0x0F	E2h	0x23	FFh	VSYNC_B_ST	-		FLC	575
0x10	84h	0x24	55h	VSYNC_H_POS_ST	-	LVTIME2	FSWC	38
0x11	03h	0x25	66h	VSYNC_H_POS_END	-		FEWC	4
0x12	82h	0x26	02h			LVTIME3	FPW2	4
0x13	04h	0x27	00h				FLC2	575
0x14	E0h	0x28	80h			LVTIME4	FSWC2	38
							FEWC2	4

8.6.6 720x480i

720x480i @ 60Hz								
Pixel Clock		27.027	MHz					
H Frequency		15.75	KHz					
V Frequency		60	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	276	LCTRL	IV	1
0x01	85h	0x15	00h	V2_BLANK	262		IH	1
0x02	88h	0x16	1Ch	V1_BLANK	22		IP	0
0x03	00h	0x17	03h	V_LINE	525		DP	1
0x04	A0h	0x18	23h	H_LINE	1716		NI	0
0x05	34h	0x19	3Ch	VSYNC_POL	1		TV	1
0x06	07h	0x1A	38h	INT_PRO_MODE	1		TFT	0

0x07	0Dh	0x1B	40h	V_BOT_ST	285		STN	0
0x08	0Bh	0x1C	88h	V_BOT_END	525	LHTIME1	LPW	123
0x09	D8h	0x1D	7Ch	HSYNC_START	36		LPC	1439
0x0A	C2h	0x1E	09h	HSYNC_END	160	LHTIME2	LSWC	113
0x0B	A4h	0x1F	13h	HSYNC_POL	1		LEWC	37
0x0C	46h	0x20	06h	VSYNC_T_END	7	LVTIME1	VDB	0
0x0D	28h	0x21	00h	VSYNC_T_ST	4		VDF	0
0x0E	4Dh	0x22	54h	VSYNC_B_END	269		FPW	5
0x0F	E3h	0x23	FEh	VSYNC_B_ST	266		FLC	479
0x10	84h	0x24	15h	VSYNC_H_POS_ST	896	LVTIME2	FSWC	29
0x11	03h	0x25	66h	VSYNC_H_POS_END	896		FEWC	7
0x12	82h	0x26	02h			LVTIME3	FPW2	5
0x13	04h	0x27	00h				FLC2	479
0x14	E0h	0x28	80h			LVTIME4	FSWC2	31
							FEWC2	7

720x480i @ 59.94Hz								
Pixel Clock		27	MHz					
H Frequency		15.734	KHz					
V Frequency		59.94	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	276		IV	1
0x01	85h	0x15	00h	V2_BLANK	262		IH	1
0x02	88h	0x16	1Ch	V1_BLANK	22		IP	0
0x03	00h	0x17	03h	V_LINE	525	LCTRL	DP	1
0x04	A0h	0x18	23h	H_LINE	1716		NI	0
0x05	34h	0x19	3Ch	VSYNC_POL	1		TV	1
0x06	07h	0x1A	38h	INT_PRO_MODE	1		TFT	0
0x07	0Dh	0x1B	40h	V_BOT_ST	285		STN	0
0x08	0Bh	0x1C	88h	V_BOT_END	525	LHTIME1	LPW	123
0x09	D8h	0x1D	7Ch	HSYNC_START	36		LPC	1439
0x0A	C2h	0x1E	09h	HSYNC_END	160	LHTIME2	LSWC	113
0x0B	A4h	0x1F	13h	HSYNC_POL	1		LEWC	37
0x0C	06h	0x20	06h	VSYNC_T_END	7	LVTIME1	VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	4		VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	269		FPW	5
0x0F	E2h	0x23	FFh	VSYNC_B_ST	266		FLC	479
0x10	84h	0x24	55h	VSYNC_H_POS_ST	896	LVTIME2	FSWC	29
0x11	03h	0x25	66h	VSYNC_H_POS_END	896		FEWC	7
0x12	82h	0x26	02h			LVTIME3	FPW2	5
0x13	04h	0x27	00h				FLC2	479
0x14	E0h	0x28	80h			LVTIME4	FSWC2	31
							FEWC2	7

8.6.7 720x576i

720x576i @ 50Hz

Pixel Clock		27	MHz					
H Frequency		15.625	KHz					
V Frequency		50	Hz					
HDMI PHY Setting				HDMI link setting		LCD Setting		
Address	Value	Address	Value	H_BLANK	288		IV	1
0x01	85h	0x15	00h	V2_BLANK	312		IH	1
0x02	88h	0x16	1Ch	V1_BLANK	24		IP	0
0x03	00h	0x17	03h	V_LINE	625	LCTRL	DP	1
0x04	A0h	0x18	23h	H_LINE	1728		NI	0
0x05	34h	0x19	3Ch	VSYNC_POL	1		TV	1
0x06	07h	0x1A	38h	INT_PRO_MODE	1		TFT	0
0x07	0Dh	0x1B	40h	V_BOT_ST	337		STN	0
0x08	0Bh	0x1C	88h	V_BOT_END	625	LHTIME1	LPW	125
0x09	D8h	0x1D	7Ch	HSYNC_START	22		LPC	1439
0x0A	C2h	0x1E	09h	HSYNC_END	148	LHTIME2	LSWC	137
0x0B	A4h	0x1F	13h	HSYNC_POL	1		LEWC	23
0x0C	06h	0x20	06h	VSYNC_T_END	5		VDB	0
0x0D	D8h	0x21	00h	VSYNC_T_ST	2	LVTIME1	VDF	0
0x0E	5Bh	0x22	54h	VSYNC_B_END	317		FPW	5
0x0F	E2h	0x23	FFh	VSYNC_B_ST	314		FLC	575
0x10	84h	0x24	55h	VSYNC_H_POS_ST	888	LVTIME2	FSWC	37
0x11	03h	0x25	66h	VSYNC_H_POS_END	888		FEWC	3
0x12	82h	0x26	02h			LVTIME3	FPW2	5
0x13	04h	0x27	00h				FLC2	575
0x14	E0h	0x28	80h			LVTIME4	FSWC2	39
							FEWC2	3

9 Camera Interface

9.1 Overview

The TCC8900 has the camera interface (CIF). Its features are as follows.

- INPUT FORMATS
  - CCIR 601/656, YUV4:2:2 (8bits)
  - Support Interlace mode
  - RGB555, RGB565 (8 bits)
- OUTPUT FORMATS
  - YUV4:2:2, YUV4:2:0
  - Convert YUV422 to YUV420 mode
- Skip Frame Mode
- 4-level alpha blending/Chroma-Keying (1 overlay image)
- Image effecter
  - Bias
  - Inversion of Y value
  - Strong C mode
  - Y clipping
  - Color filter
  - Sketch mode
  - Embossing
  - Gray
  - Sepia
- Image scaler (ratio : original * 256/target)
  - Upscale: 1 : 4
  - Downscale: 64 : 1
  - Each step size is 256 step

9.2 Operation

The block diagram of CIF is shown in Figure 9.1.

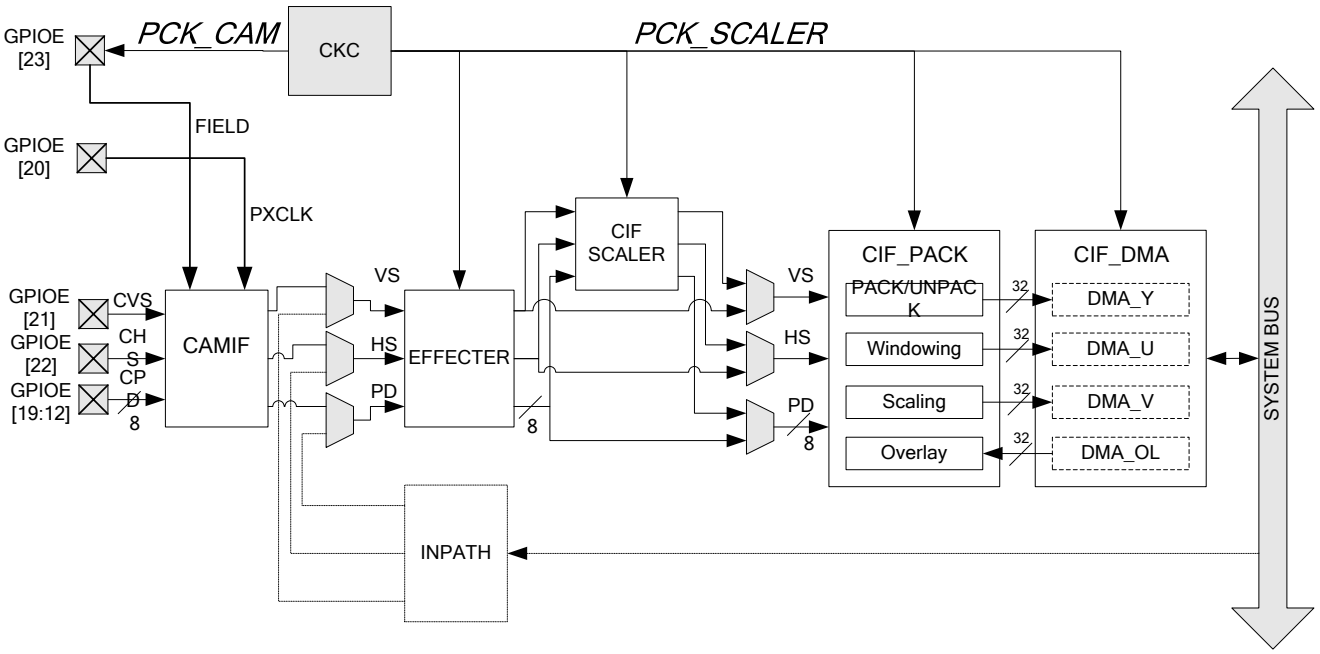


Figure 9.1 CIF Block Diagram

9.2.1 External Camera Module Interface (CAMIF)

The CIF uses CCIR-601/656-like protocol to interface with an external camera module. These signals are shown in the Figure 9.2.

The C656 bit of ICPCR1 register determines whether CPD[7:0] includes the embedded sync information or not. If the sync information is embedded in the pixel data stream, the additional sync signals, which are CVS and CHS, are not used. In this case, 656FCR1 and 656FCR2 register need to be configured. Otherwise, CVS and CHS are used for the vertical synchronous signal and the horizontal synchronous signal respectively. If the sync information is in the CVS and CHS, TV and VI bit of ICPCR1 register can be used to adjust the misaligned vertical (CVS) and horizontal sync (CHS).

When the format of input data is RGB565 or RGB555, it should be converted to YUV. Refer to CR2Y register.

PXCLK in Figure 9.1 is from a camera module. If a camera module needs the external clock input, CCKO can be used as input to one. Refer to CKC_CTRL..

The POL bit of ICPCR1 register determines whether pixel data are latched at the rising edge or the falling edge of PXCLK. And CVS and CHS polarity are determined by the VSP and the HSP bit of ICPCR1 register respectively.

The geometric property of input image is determined by CEIS register (Figure 9.3).

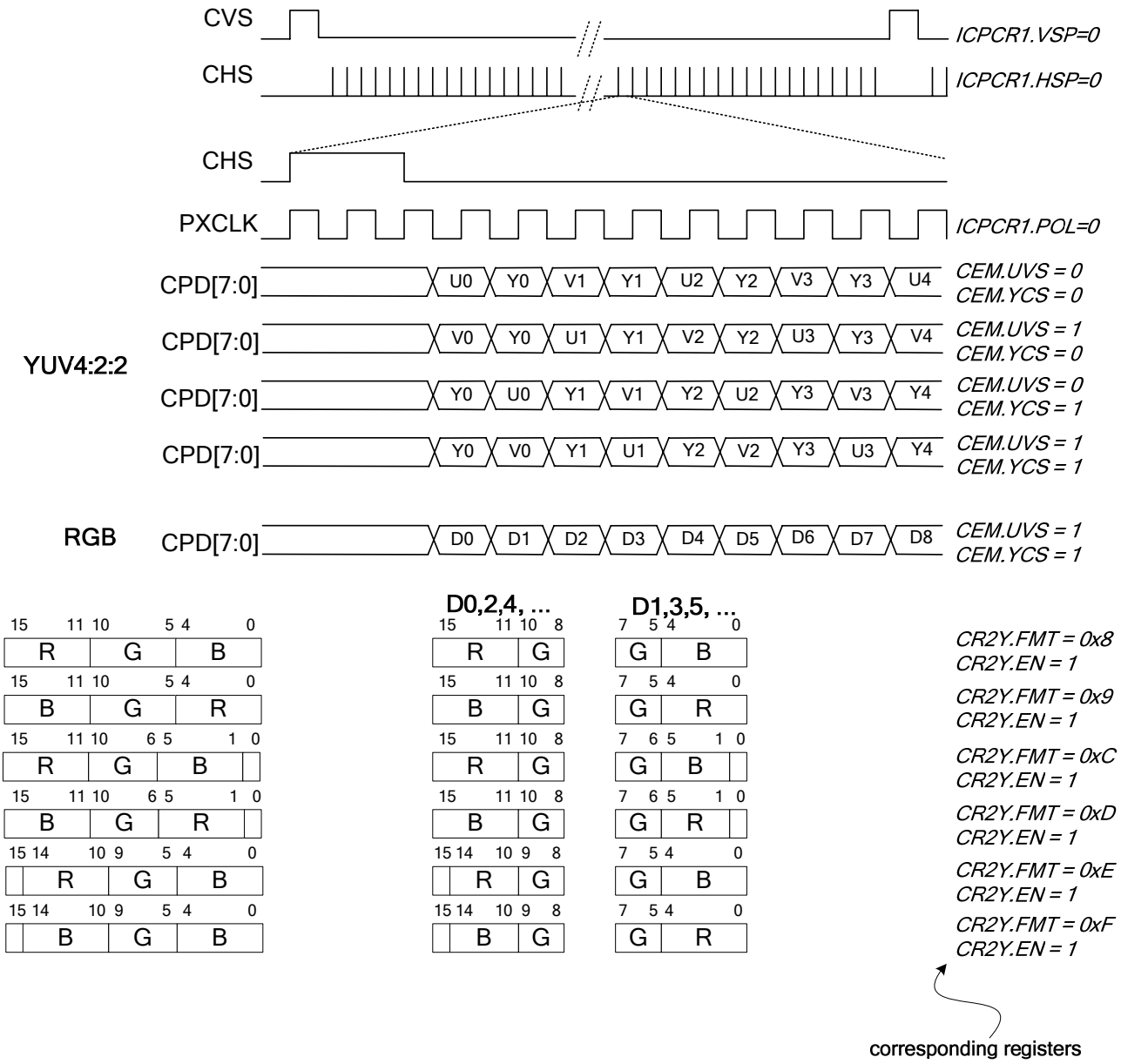


Figure 9.2 Input Data Format From the external camera module

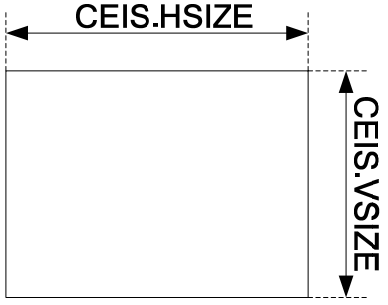


Figure 9.3 Input Image Size From the External Camera Module

In interlace mode, field signal is generated from input port signals, FIELD, HS(not same as data enable signal). PCK_CAM and FIELD signal shares the same port of TCC8900.(Figure 9.4)

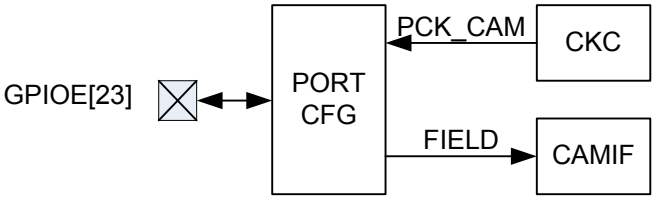
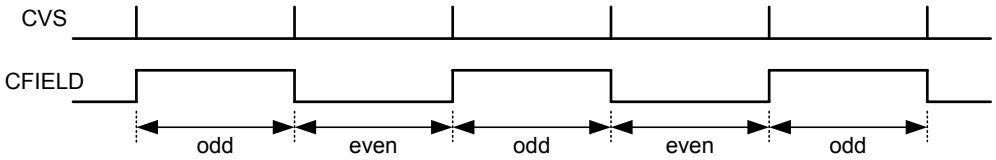
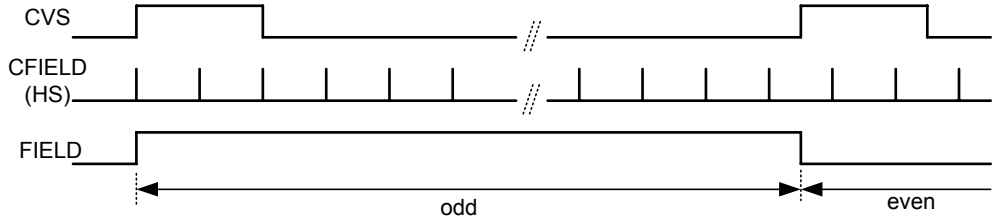


Figure 9.4 Port Control in PCK_CAM and FIELD

Field signal generation method is shown as Figure 9.5. In case Figure 9.5(a), CFIELD port input is directly used as field signal. In case (b), according to the level of CFIEID port input, HS signal(not same as CHS which indicates data enable time, and preserves it's value during vertical blanking), the field signal is generated at the positive edge of VS.

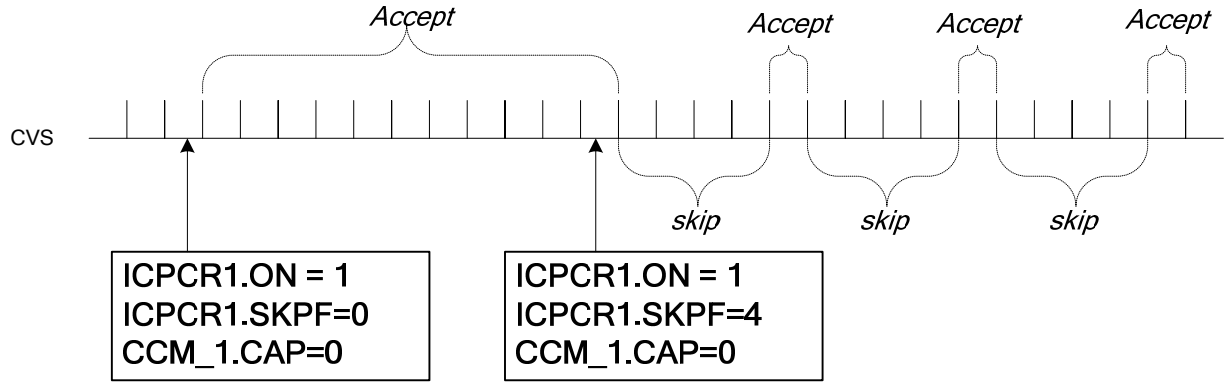


(a) CFIELD port is field signal

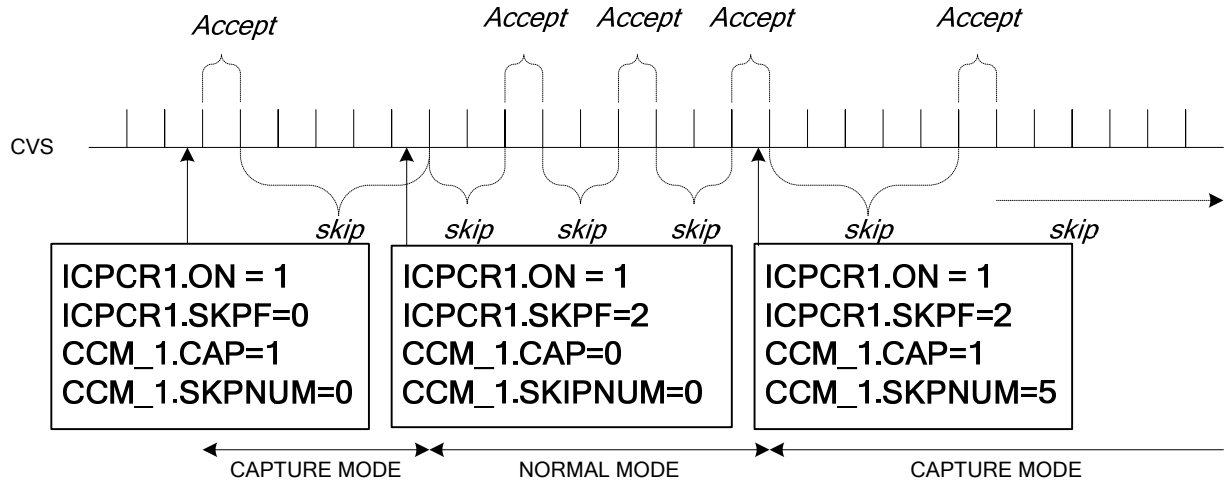


(b) Generate field signal when CFIELD port is HS signal
- HS is not data enable signal

Figure 9.5 Generate Field Signal



(a) preview mode



(b) normal and capture mode

Figure 9.6 Accepting Frame Data from the External Camera Module

The CIF has two modes to accept one frame data from the external camera module. One is the preview mode. The other is the capture mode. The preview mode accepts the successive frame data from the external camera module. But, the capture mode accepts only one frame data.

Although the camera module sends the successive frame data to the CIF, the CIF can discard several frames before accepting a frame. They are called the skip frames. When the CIF is in the preview mode, the skip frames are determined by SKPF bits of ICPCR1register. When the CIF is in the capture mode, the skip frames are determined by SKIPNUM bits of CCM_1 register. Refer to ICPCR1 register on page 9-213 and CCM_1 register on page 9-228. Figure 9.6 shows the difference between the preview mode and the capture mode.

The accepted frames are sent to the effecter and their pixel format is YUV 4:2:2.

9.2.2 Effecter

The effecter supports YUV bias (YUV offset value), Inversion of Y value, strong C mode (x2 C value), Y clipping, color filter, sketch mode, embossing (positive and negative), gray, and sepia. Refer to the effecter registers on page 9-233.

9.2.3 CIF Scaler

The TCC8900 provides the CIF scaler for scaling the image from the effecter. The supported scaling ratio is from 1 : 4 (zoom up to 4 times) to 64 : 1 (zoom down to 64 times), and scaling step is 256.

The output image of the effecter becomes the CIF scaler input. Figure 9.7 shows relationship between them and how to specify them. Refer to CSSO and CSSS register on page 9-237. When the BPS bit of ICPCR1 register is set to 1, the effecter output is not scaled by the CIF scaler.

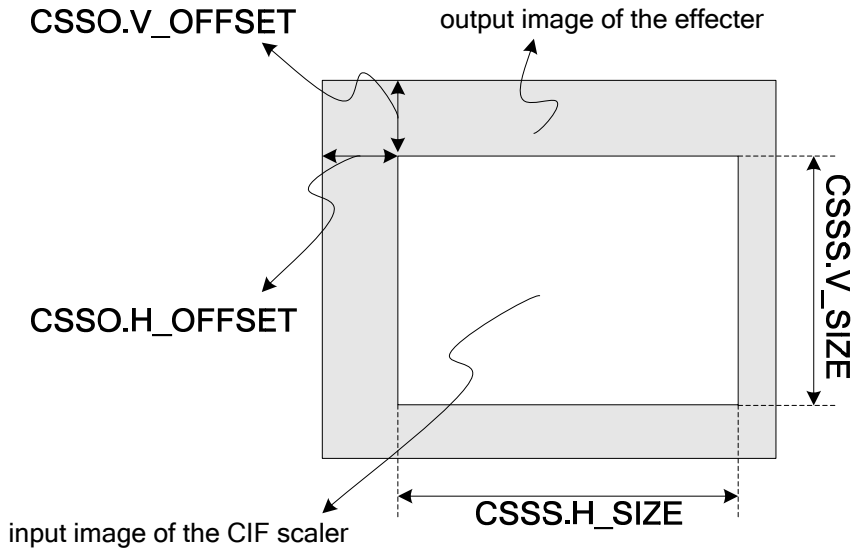


Figure 9.7 Output Image of the Effecter and Input Image of the CIF Scaler

9.2.4 Overlay

The CIF scaler output image (If the CIF scaler is disabled, it is equal to the effecter output image) can be mixed with another image in the memory. This image in the memory is called the overlay image and the CIF scaler output image is called the background image. For this operation, the CIF has the alpha-blending and the chroma-keying function.

Figure 9.8 shows how to specify the geometric property of overlay image. Refer to OIS, OIW1, and OIW2 register on page 9-227. If the size of overlay image (not active overlay image) is equal to the size of background image, the OM bit of OCTRL1 should be set to 0 (full image overlay). Otherwise, it should be set to 1 (block image overlay).

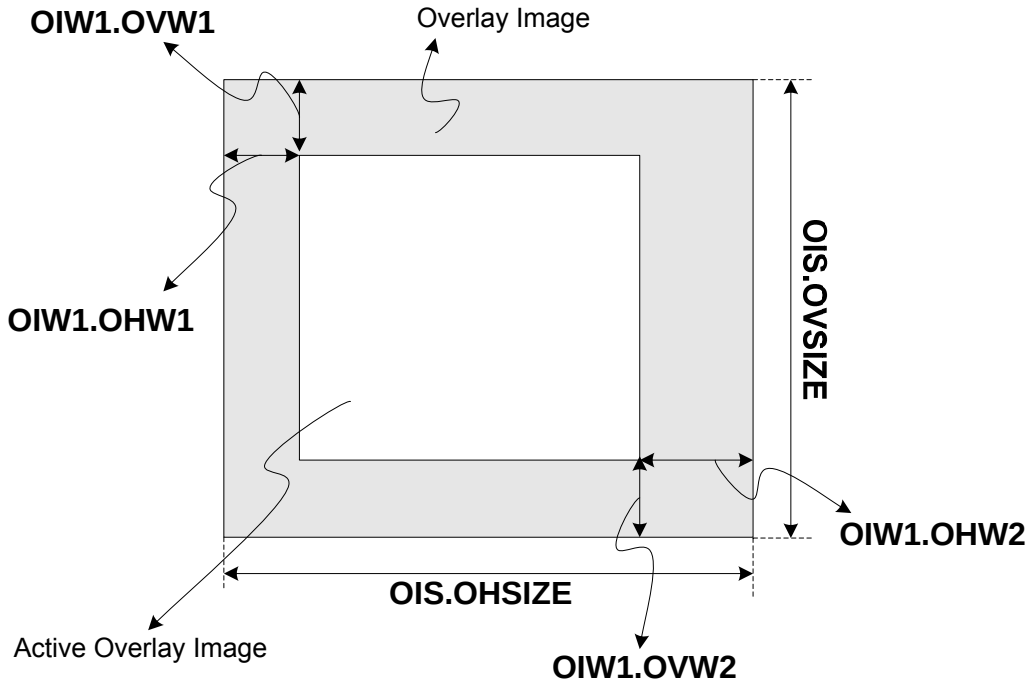


Figure 9.8 The Geometric Property of Overlay Image

9.2.5 Store to the Memory

The image data which is from the camera module through effecter and CIF scaler to overlay logic need to be stored to the memory. Figure 9.9 shows how to specify the image data to be stored. Refer to IIS, IIW1, and IIW2 register on page 9-217.

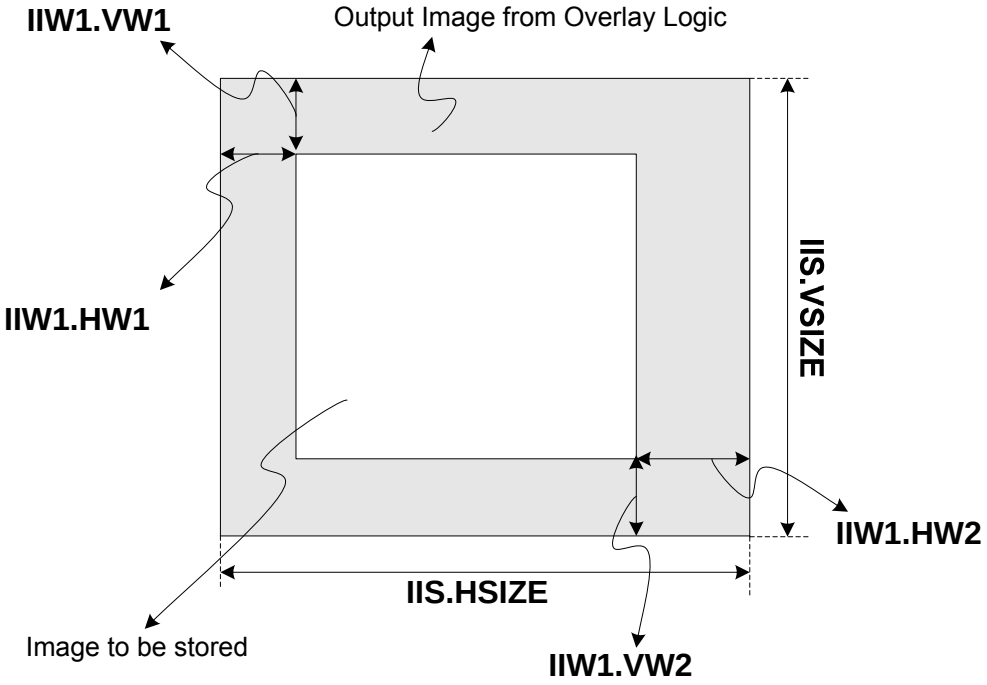


Figure 9.9 Image Data to be Stored to the Memory

The image data from the overlay logic is YUV4:2:2. The image format to be stored can be YUV4:2:2SEQ0, YUV4:2:2, or YUV4:2:0. It is determined by the BP and the M420 bit of ICPCR1 register. Refer to the ICPCR1 register on page 9-213.

The CIF has two modes to store the image data. One is the frame mode and the other is the rolling mode. The frame mode needs the memory space to store a whole frame data. But, the rolling mode needs the memory space to store the part of a frame data. It is determined by the RLY, RLU, and RLV bits of CCM_1. Therefore, in the rolling mode, before the frame data are overwritten, they should be moved to another memory space.

CDCR2, CDCR3, and CDCR4 registers are the base addresses which the image data is stored to. They are for Y, U, and V image data respectively. Additionally, the end addresses are required in the rolling mode. They are determined by CDCR5, CDCR6, and CDCR7 register. Refer to these registers on page 9-218.

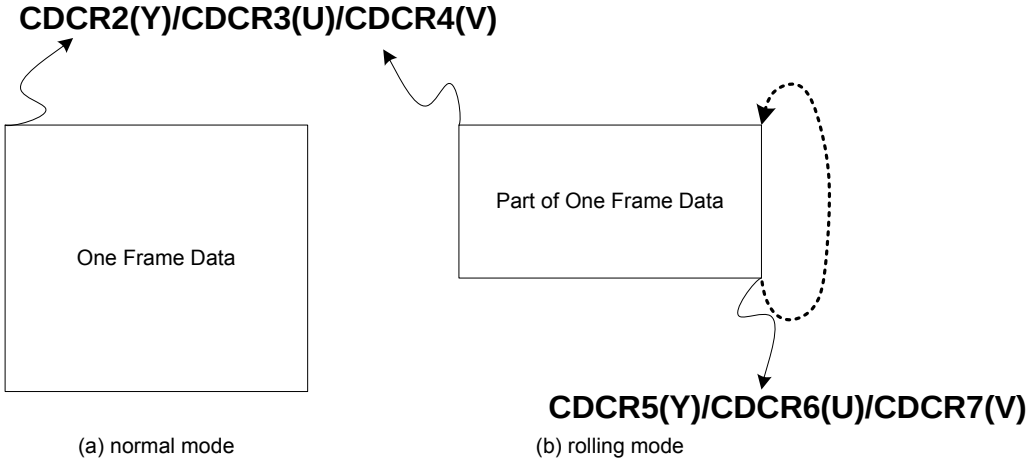


Figure 9.10 Two Modes to Store the Image Data

The CIF can inform the on-chip CPU its status during storing image data. When the CIF has stored a whole frame data to the memory, SOF of CIRQ register is set to 1. SCF and SOF of CIRQ register is the same function, but SCF is only available in the capture mode. In the rolling mode, ROLY, ROLU, and ROLV of CIRQ register represent whether the current address which is used to store the image data reaches the corresponding “end address” or not. ENS of CIRQ register is similar to ROLY, U, V but the current address is compared with CESA register. VIT of CIRQ register is set to 1 whenever the number of lines which the CIF receives data from the camera module and stores them to specified memory is equal to the multiple of 16*VCNT lines. If VCNT of CCM_2 register is set to 0, VIT bit is never set to 1. All these status bits can be the interrupt request source. Refer to CIRQ on page 9-222.

9.3 Camera Register Descriptions

Table 9.1 CIF Register Map (Base Address = 0xF0230000)

Name	Address	Type	Reset	Description
ICPCR1	0x00	W/R	0x00000000	Input Image Color/Pattern Configuration Register 1
656FCR1	0x04	W/R	0x06ff0000	CCIR656 Format Configuration Register 1
656FCR2	0x08	W/R	0x010b	CCIR656 Format Configuration Register 2
IIS	0x0C	W/R	0x00000000	Input Image Size
IIW1	0x10	W/R	0x00000000	Input Image Windowing 1
IIW2	0x14	W/R	0x00000000	Input Image Windowing 2
CDCR1	0x18	W/R	0x0003	DMA Configuration Register 1
CDCR2	0x1C	W/R	0x00000000	DMA Configuration Register 2
CDCR3	0x20	W/R	0x00000000	DMA Configuration Register 3
CDCR4	0x24	W/R	0x00000000	DMA Configuration Register 4
CDCR5	0x28	W/R	0x00000000	DMA Configuration Register 5
CDCR6	0x2C	W/R	0x00000000	DMA Configuration Register 6
CDCR7	0x30	W/R	0x00000000	DMA Configuration Register 7
CDCR8	0x34	W/R	0x00000000	DMA Configuration Register 8
FIFOSTATE	0x38	R	0x00000000	FIFO Status Register
CIRQ	0x3C	W/R	0x00000000	Interrupt & Status register
OCTRL1	0x40	W/R	0x37000000	Overlay Control 1
OCTRL2	0x44	W/R	0x00000000	Overlay Control 2
OCTRL3	0x48	W/R	0x00000000	Overlay Control 3
OCTRL4	0x4C	W/R	0x00000000	Overlay Control 4
OIS	0x50	W/R	0x00000000	Overlay Image Size
OIW1	0x54	W/R	0x00000000	Overlay Image Windowing 1
OIW2	0x58	W/R	0x00000000	Overlay Image Windowing 2
COBA	0x5C	W/R	0x00000000	Overlay Base Address
COBO	0x60	W/R	0x00000000	Overlay Base Address Offset
CDS	0x64	W/R	0x00000000	Camera Down Scaler
CCM1	0x68	W/R	0x00000000	Capture Mode Configuration 1
CCM2	0x6C	W/R	0x00000000	Capture Mode Configuration 2
CESA	0x70	W/R	0x00000000	Point Encoding Start Address
CR2Y	0x74	W/R	0x00000000	RGB2YUV Format converter Configuration
CCYA	0x78	R	-	Current Y Address
CCYU	0x7C	R	-	Current U Address
CCYV	0x80	R	-	Current V Address
CCLC	0x84	R		Current Line count

Input Image Color/Pattern Configuration Register 1 (ICPCR1) 0xF0230000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ON	-	TV	VI	UF	ITEN	FS	FP	BPS	0	POL	SKPF			M420	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BP	-	C656	CP	PF<1:0>		RGBM<1:0>		RGBBM<1:0>		CS<1:0>		0	BO	HSP	VSP

ON [31]	On/Off on CIF
0	Disable CIF
1	Enable CIF

TV [29]	TV signal
0	CIF sync signal
1	TV sync signal

VI [28]	Vertical blank insert
0	Do not inset vertical blank in even field
1	Insert vertical blank in 1 line in even field

UF [27]	Use field signal
0	Don’t use
1	Use field signal

In interlace mode, it is decided whether field signal is used or not. The field is determined from direct field input signal or relationship between VS and HS. Also the field input port must be connected with external field input signal(field or HS) obviously.

ITEN [26]	Interlace enable
0	Disable
1	Enable

This field must be set in case of interlace input mode

FS [25]	Field port signal select
0	Field signal
1	Horizontal sync

Select field port input between field signal and HS, when the field signal is used in interlace mode.

FP [25]	Field port polarity
0	Active low (odd field – high, HS - active low)
1	Active high (odd field – low, HS – active high)

BPS [23]	Bypass Scaler
0	CIF scaler is used.
1	Bypass. CIF scaler is not used.

POL [21]	PXCLK Polarity
0	Rising edge
1	Falling edge.

SKPF [20:18]	Skip frame
0	Not-skipped
1 ~ 7	Number of frames to be skipped.

M420 [17:16]	YUV4:2:2 to YUV4:2:0
00	Not-Convert
10	Odd lines of U and V image are skipped.
11	Even lines of U and V image are skipped.

BP [15]	Bypass (Non-Separate)
0	Data format to be stored to memory If M420 bit is set to 0, the data format is YUV 4:2:2. Otherwise, the data format is YUV4:2:0.

1	YUV4:2:2SEQ0

C656 [13]	Convert 656 format
0	Disable
1	Enable

CP [12]	Color Pattern
0	It should be set to 0

PF [11:10]	Pattern Format
01	“01b” SHOULD BE WRITTEN TO THIS BIT.

RGBM [9:8]	RGB Mode
00	It should be set to 0.

RGBBM [7:6]	RGB Bit Mode
00	It should be set to 0.

CS [5:4]	Color Sequence
00	It should be set to 0.

BO [2]	Bus Order
1	Switch the MSB/LSB 8bit bus. (Don’t’ change)

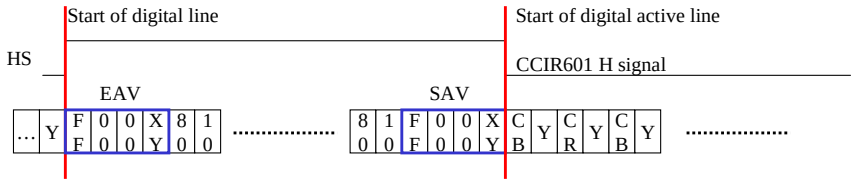
HSP [1]	Horizontal Sync Polarity
0	Active low
1	Active high (default)

VSP [0]	Vertical Sync Polarity
0	Active low (default)
1	Active high Y

CCIR656 Format Configuration Register 1 (656FCR1) 0xF0230004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					PSL<1:0>		0	FPV<7:0>							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPV<7:0>								TPV<7:0>							

This register and next register (656FCR1 and 656FCR2) define the configuration of CCIR656.



	8-bit data							
	D7 (MSB)	D6	D5	D4	D3	D2	D1	D0
preamble	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
status	1	F	V	H	P3	P2	P1	P0

•Status word define
- •F='0' for field 1, '1' for field 2 (interlace mode. If progressive, this value is '0')
- •V='1' during vertical blanking
- •H='0' at SAV, '1' at EAV

•Protection bits
- •P3=V xor H
- •P2=F xor H
- •P1=F xor V
- •P0=F xor V xor H

Figure 9.11 CCIR-656 Format Diagram

PSL [26:25]	Preamble and Status Location
00	The status word is located in the first byte of EAV & SAV
01	The status word is located in the second byte of EAV & SAV
10	The status word is located in the third byte of EAV & SAV
11	The status word is located in the forth byte of EAV & SAV

We must find the location of preamble and status for getting sync information. The total size of preamble and status is 4 bytes composed of 3 bytes of preamble and 1 byte of status. This register is used to find the location of status word.

FIELD	Description
FPV [23:16]	First preamble value Define the first preamble value. Default value is 0x00.
SPV [15:8]	Second preamble value Define the second preamble value. Default value is 0x00.
TPV [7:0]	Third preamble value Define the third preamble value. Default value is 0x00.

CCIR656 Format Configuration Register 2 (656FCR2) 0xF0230008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		FB<3:0>					HB<3:0>				0	VB<3:0>			

FIELD	Description
FB [13:10]	field information In status word, it refer to the location of 'F' and F valie at field imformation. The MSB 3 bit means the location of 'F', the other bit means value at field imformation. Default value is 0x00. but, CCRI656, this value is 0x0c.
HB [8:5]	Horizontal blank In status word, it refers to the location of 'H' and H value at blanking. The MSB 3 bit means the location of 'H', the other bit means value at blanking. Default value is 0x09.
VB [3:0]	Vertical blank In status word, it refers to the location of 'V' and V value at blanking. The MSB 3 bit means the location of 'V', the other bit means value at blanking. Default value is 0x0B.

Input Image Size (IIS) **0xF023000C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HSIZE <15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VSIZE <15:0>															

FIELD	Description
HSIZE [31:16]	Horizontal size of input image
VSIZE [15:0]	Vertical size of input image

If the CIF scaler is used, this should be the same as CSDS register. Otherwise, this should be the same as CEIS register

Input Image Windowing1 (IIW1) **0xF0230010**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HW1<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HW2<15:0>															

Input Image Windowing2 (IIW2) **0xF0230014**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VW1<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VW2 <15:0>															

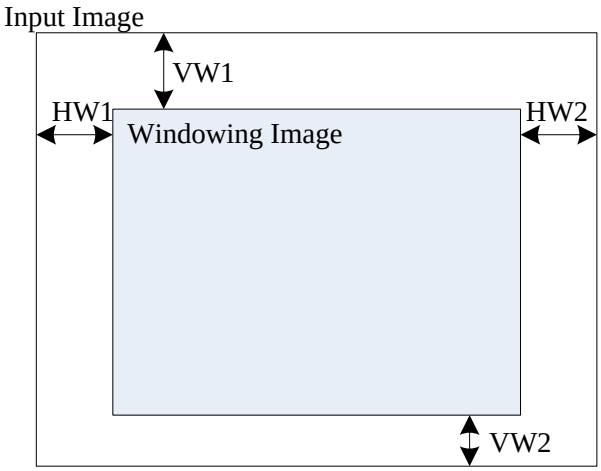


Figure 9.12 Input Image Windowing

Default value of HW1, HW2, VW1, and VW2 is 0.

CIF DMA Configuration Register 1 (CDCR1) 0xF0230018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												TM	Lock	BS <1:0>	

BS [1:0]	Preamble and Status Location
00	The DMA transfers the image data as 1 word to memory.
01	The DMA transfers the image data as 2 words to memory.
10	The DMA transfers the image data as 4 words to memory.
11	The DMA transfers the image data as 8 words to memory. (default)

LOCK [2]	Lock Transfer
0	Non-Lock (default)
1	Lock Transfer

TM [3]	Transfer Method
0	Burst Transfer(default)
1	INC Transfer

CIF DMA Configuration Register 2 (CDCR2) 0xF023001C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Input Image /Y(G) Channel Base Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Input Image / Y(G) Channel Base Address<15:0>															

FIELD	Description
CDCR2 [31:0]	Input Image Base Address. / Y(G) channel base address

CIF DMA Configuration Register 3 (CDCR3) 0xF0230020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U(R) Channel Base Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U(R) Channel Base Address<15:0>															

FIELD	Description
CDCR3 [31:0]	U(R) Channel Base Address.

CIF DMA Configuration Register 4 (CDCR4) 0xF0230024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
V(B) Channel Base Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V(B) Channel Base Address<15:0>															

FIELD	Description
CDCR4 [31:0]	V(B) Channel Base Address

CIF DMA Configuration Register 5 (CDCR5) 0xF0230028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFS1<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFS0<15:0>															

FIELD	Description
OFS1[31:16]	U,V channel base address offset. This offset means the line start point to the next line start point.
OFS0[15:0]	Y channel base address offset. This Offset means the line start point to the next line start point.

CIF DMA Configuration Register 6 (CDCR6) 0xF023002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Input Image /Y(G) Channel End Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Input Image / Y(G) Channel End Address<15:0>															

FIELD	Description
CDCR5 [31:0]	Input Image End Address. / Y(G) channel end address This mode is operated, when rolling address Y is enabled.

CIF DMA Configuration Register 7 (CDCR7) 0xF0230030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U(R) Channel End Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U(R) Channel End Address<15:0>															

FIELD	Description
CDCR6 [31:0]	U(R) Channel End Address This mode is operated, when rolling address U is enabled.

CIF DMA Configuration Register 8 (CDCR8) 0xF0230034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
V(B) Channel Base Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V(B) Channel Base Address<15:0>															

FIELD	Description
CDCR7 [31:0]	V(B) Channel End Address This mode is operated, when rolling address U is enabled.

FIFO States (FIFOSTATE) 0xF0230038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										CLR	0	REO	REV	REU	REY
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0		WEO	WEV	WEU	WEY	0	EO	EV	EU	EY	0	FO	FV	FU	FY

CLR [21]	Clear FIFO states
1	When writing 1 to this bit, all status bits are cleared.

REO [19]	Overlay FIFO Read Error
0	Normal operation.
1	It represents that FIFO underrun is occurred.

REV [18]	Cr(B) Channel FIFO Read Error
0	Normal
1	It represents that FIFO underrun is occurred.

REU [17]	Cb(R) Channel FIFO Read Error
0	Normal
1	It represents that FIFO underrun is occurred.

REO [16]	Y(G) Channel FIFO Read Error
0	Normal
1	It represents that FIFO underrun is occurred.

WEO [13]	Overlay FIFO Write Error
0	Normal
1	It represents that FIFO overrun is occurred.

WEV [12]	Cr(B) Channel FIFO Write Error
0	Normal
1	It represents that FIFO overrun is occurred.

WEU [11]	Cb(R) Channel FIFO Write Error
0	Normal
1	It represents that FIFO overrun is occurred.

WEY [10]	Y(G) Channel FIFO Write Error
0	Normal
1	It represents that FIFO overrun is occurred.

EO [8]	Overlay FIFO Empty Signal
0	It represents that FIFO is not empty.
1	It represents that FIFO is empty.

EV [7]	V(B) Channel FIFO Empty Signal
0	It represents that FIFO is not empty.
1	It represents that FIFO is empty.

EU [6]	U(R) Channel FIFO Empty Signal
0	It represents that FIFO is not empty.
1	It represents that FIFO is empty.

EY [5]	Y(G) Channel FIFO Empty Signal
0	It represents that FIFO is not empty.
1	It represents that FIFO is empty.

FO [3]	Overlay FIFO Full Signal
0	It represents that FIFO is not full.
1	It represents that FIFO is full.

FV [2]	V(B) Channel FIFO Full Signal
0	It represents that FIFO is not full.
1	It represents that FIFO is full.

FU [1]	U(R) Channel FIFO Full Signal
0	It represents that FIFO is not full.
1	It represents that FIFO is full.

FY [0]	Y(G) Channel FIFO Full Signal
0	It represents that FIFO is not full.
1	It represents that FIFO is full.

CIF Interrupt Register (CIRQ) 0xF023003C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IEN	URV	ITY	ICR	0	MVN	MVP	MVIT	MSE	MSF	MENS	MRLV	MRLU	MRLY	MSCF	MSOF
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SFEN	0	0	VSS	FL	VN	VP	VIT	SE	SF	ENS	ROLV	ROLU	ROLY	0	SOF

IEN [31]	Interrupt Enable
0	Interrupt disable
1	Interrupt enable

URV [30]	Update Register in VSYNC
0	All CIF registers are applied to CIF operation regardless of VSYNC.
1	All CIF registers are applied to CIF operation at rising edge of CVS signal.

ITY [29]	Interrupt Type.
1	“1” SHOULD BE WRITTEN TO THIS BIT.

ICR [28]	Interrupt Clear
1	When writing 1 to this bit, the CIF interrupt is cleared.

MVN [26]	Mask interrupt of VS negative edge
0	Enable When VN bit is set to 1, the CIF interrupt request is issued.
1	Disable

MVP [25]	Mask interrupt of VS positive edge
0	Enable When VP bit is set to 1, the CIF interrupt request is issued.
1	Disable

MVIT [24]	Mask interrupt of VCNT Interrupt
0	Enable When VIT bit is set to 1, the CIF interrupt request is issued.
1	Disable

MSE [23]	Mask interrupt of Scaler Error
0	Enable When SE bit is set to 1, the CIF interrupt request is issued.
1	Disable

MSF [22]	Mask interrupt of Scaler finish
0	Enable When SF bit is set to 1, the CIF interrupt request is issued.
1	Disable

MENS [21]	Mask interrupt of Encoding start
0	Enable When ENS bit is set to 1, the CIF interrupt request is issued.
1	Disable

MRLV [20]	Mask interrupt of Rolling V address.
0	Enable When RLV bit is set to 1, the CIF interrupt request is issued.
1	Disable

MRLU [19]	Mask interrupt of Rolling U address.

0	Enable When RLU bit is set to 1, the CIF interrupt request is issued.
1	Disable

MRLY [18]	Mask interrupt of Rolling Y address.
0	Enable When RLY bit is set to 1, the CIF interrupt request is issued.
1	Disable

MSCF [17]	Mask interrupt of Capture frame.
0	Enable When SCF bit is set to 1, the CIF interrupt request is issued.
1	Disable

MSOF [16]	Mask interrupt of Stored one frame.
0	Enable When SOF bit is set to 1, the interrupt request is issued.
1	Disable

SFEN [15]	Generate SOF mode in interlace
0	Generate SOF when 2-field images are stored memory.
1	Without field number, generate SOF when 1-field image is stored memory.

VSS [12]	Status of vertical sync.
0	Non- vertical sync blank area.
1	Vertical sync blank area.

FL [11]	Field signal states
0	Even field
1	Odd field

VN [10]	VS negative.
0	-
1	It represents that CVS falling-edge is detected.

VP [9]	VS positive
0	-
1	It represents that CVS rising-edge is detected.

VIT [8]	VCNT Interrupt.
0	-
1	Refer to CCM_2 register on page 9-229. When writing 1 to this bit, it is cleared.

SE [6]	Scaler Error.
0	-
1	It represents that CIF scaler has an error during scaling operation. When writing 1 to this bit, it is cleared.

SF [6]	Scaler Finish.
0	-
1	It represents that CIF scaler operation is completed. When writing 1 to this bit, it is cleared.

ENS [5]	Encoding start status.
0	-
1	Refer to CESA register on page 9-230

	When writing 1 to this bit, it is cleared.

ROLV [4]	Rolling V address status.
0	-
1	It represents that the current DMA V address reaches the DMA V rolling end address. When writing 1 to this bit, it is cleared.

ROLU [3]	Rolling U address status.
0	-
1	It represents that the current DMA U address reaches the DMA U rolling end address. When writing 1 to this bit, it is cleared.

ROLY [2]	Rolling Y address status.
0	-
1	It represents that the current DMA Y address reaches the DMA Y rolling end address. When writing 1 to this bit, it is cleared.

SOF [0]	Stored One frame
0	-
1	It represents that the CIF has received one frame data from the camera module and stored them to specified memory completely. It does not consider CAP bit.

Overlay Control 1 (OCTRL1) 0xF0230040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			OCNT<4:0>					0							OM
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			OE	0	XR1	XR0	0	AP1<1:0>		AP0<1:0>		0	AEN	0	CEN

OM[16]	Overlay Method
0	Full image overlay
1	Block image overlay

Full image overlay mode, overlay image size is equal to the input image size.

OE[12]	Overlay Enable
0	Disable
1	Enable

XR1[10]	XOR in AP1 is 3
0	XOR operation
1	100 %

When AP1 is 3 and CEN & AEN is 1, we select the 100% alpha value or XOR operation.

XR0 [9]	XOR in AP0 is 3
0	XOR operation
1	100 %

When AP0 is 3 and CEN & AEN is 1, we select the 100% alpha value or XOR operation

AP1 [7:6]	Alpha Value in alpha is 1
0	25 %
1	50 %
2	75 %
3	100 % or XOR operation (for XR value)

AP0 [5:4]	Alpha Value in alpha is 0
0	25 %

1	50 %
2	75 %
3	100 % or XOR operation

When RGB565 and AEN are set, alpha value depends on AP0 value.

AEN[2]	Alpha Enable
0	Disable
1	Enable

CEN[2]	Chroma key Enable
0	Disable
1	Enable

OCNT[29:24]	Overlay Count (FIFO)
n	

Overlay Control 2 (OCTRL2) 0xF0230044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CONV	RGB		MD

CONV[3]	Color Converter Enable
0	Disable
1	Enable

RGB[2:1]	RGB Mode
0	565RGB
1	555RGB
2	444RGB
3	332RGB

MD[0]	Color Mode
0	YUV color
1	RGB color

Overlay Control 3 – key value (OCTRL3) 0xF0230048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								KEYR<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYG<7:0>								KEYB<7:0>							

KEYR[23:16]	Chroma-key value R(U)
n	Chrome-key value in R(U) channel Default value is 0x00

KEYG[15:8]	Chroma-key value G(Y)
n	Chrome-key value in G(Y) channel. Default value is 0x00

KEYB[7:0]	Chroma-key value B(V)
n	Chrome-key value in B(V) channel. Default value is 0x00

Overlay Control 4 – mask key value (OCTRL4) 0xF002304C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								MKEYR<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MKEYG<7:0>								MKEYB<7:0>							

MKEYR[23:16]	Mask Chroma-key value R(U)
N	Chrome-key value in R(U) channel Default value is 0x00

MKEYG[15:8]	Mask Chroma-key value G(Y)
N	Chrome-key value in G(Y) channel. Default value is 0x00

MKEYB[7:0]	Mask Chroma-key value B(V)
n	Chrome-key value in B(V) channel. Default value is 0x00

Overlay Image Size (OIS) 0xF0230050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OHSIZE<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSIZE<15:0>															

FIELD	Description
OHSIZE [31:16]	Horizontal size of overlay image Default value is 0x0280. (decimal is 640)
OVSIZE [15:0]	Vertical size of overlay image Default value is 0x01E0 (decimal is 480)

Overlay Image Windowing 1 (OIW1) 0xF0230054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OHW1<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OHW2<15:0>															

Overlay Image Windowing 2 (OIW2) 0xF0230058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OVW1<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVW2 <15:0>															

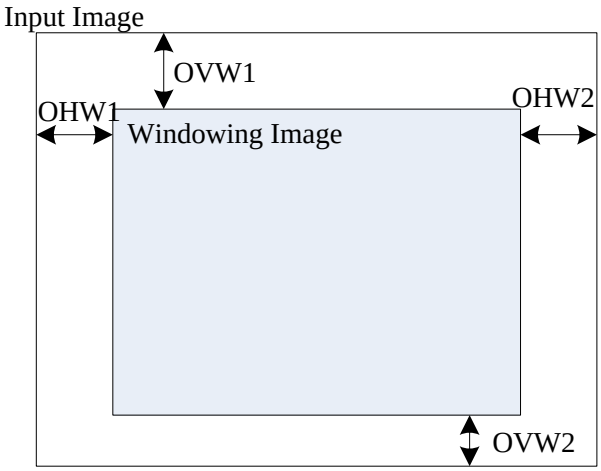


Figure 9.13 Overlay Image Windowing

CIF Overlay Base Address (COBA) 0xF023005C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Overlay Image Base Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Overlay Image Base Address<15:0>															

FIELD	Description
COBA [31:0]	Overlay Image Base Address. Default value is 0x20100000.

CIF Overlay Base Offset Address (COBO) 0xF0230060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Overlay Image Offset Address #1 <15:00>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Overlay Image Offset Address #0 <15:0>															

FIELD	Description
OIOA1 [31:16]	Overlay Image Offset Address in Chrominance.
OIOA0 [15:00]	Overlay image Offset Address in Luminance

CIF Down Scaler (CDS) 0xF0230064

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										SFH<1:0>		SFV<1:0>		0	SEN

SFH [5:4]	Horizontal Scale Factor
0	1/1 down scale
1	1/2 down scale
2	1/4 down scale
3	1/8 down scale

SFV [3:2]	Vertical Scale Factor
0	1/1 down scale
1	1/2 down scale
2	1/4 down scale
3	1/8 down scale

SEN	Scale Enable
0	Disable
1	Enable

CIF Capture mode_1 (CCM_1) 0xF0230068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENCNUM				ROLNUMV				ROLNUMU				ROLNUMY			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CB	EIT	UES	SKIPNUM				RLV	RLU	RLY	CAP

ENCNUM [31:28]	Encode INT number (using CAP mode)
0~15	Refer to EIT bit.

ROLNUMV [27:24]	Rolling Number in V (using CAP mode)
0~15	The number of times which V address is rolled while capture mode and rolling mode are enabled and one frame data is stored to the memory.

ROLNUMU [23:20]	Rolling Number in U (using CAP mode)
0~15	The number of times which U address is rolled while capture mode and rolling mode are enabled and one frame data is stored to the memory.

ROLNUMY	Rolling Number in Y (using CAP mode)

[19:16]	
0~15	The number of times which Y address is rolled while capture mode and rolling mode are enabled and one frame data is stored to the memory.

CB [10]	Capture Busy
0	-
1	It represents that frame data is storing to the memory during the capture mode.

EIT [9]	Encoding INT count
0	The ENS bit of CIRQ register is set to 1 when the current DMA Y address reaches the address to be specified in the CESA register. But, it can be occurred the only one time during storing one frame data.
1	The ENS bit of CIRQ register is set to 1 whenever the current DMA Y address reaches the address to be specified in the CESA register. ENCNUM represents the number of times which ENS bit is set to 1.

UES [8]	Using Encoding Start Address
0	The use of CESA register is disabled.
1	The use of CESA register is enabled. Refer to CESA register on page 9-230

SKIPNUM [7:4]	Skip frame number (using CAP mode)
0~15	It specifies the number of skip frames during capture mode.

RLV[3]	Rolling mode for V image
0	Disable (frame mode)
1	Enable

RLU[2]	Rolling Mode for U image
0	Disable (frame mode)
1	Enable

RLY[1]	Rolling Mode for Y image
0	Disable (frame mode)
1	Enable

CAP [0]	Capture mode
0	Preview mode
1	Capture mode

CIF CAPTURE MODE_2 (CCM_2) 0xF023006C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								VCNT							VEN

FIELD	Description (Using CAP mode)
VCNT[7:4]	VIT bit is set to 1 whenever the number of lines which the CIF receives data from the camera module and stores them to specified memory is equal to the multiple of 16*VCNT lines. If VCNT is set to 0, VIT bit is never set to 1. VCNT can be 0 ~ 7. For example, when VCNT = 3, the VIT bit is set to 1 whenever a number of line to be stored is equal to a multiple of 48 line, which is 48, 96, 144, etc.

VEN [0]	VCNT enable (Using CAP mode)
0	Disable
1	Enable (VCNT is used for setting VIT bit)

CIF Encoding Start Address (CESA) 0xF0230070

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Encoding Start Address [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Encoding Start Address [15:0]															

FIELD	Description
CESA[31:0]	Default value is 0x20100000. This is compared with the current DMA Y address. When the current DMA Y address reaches the address to be specified in the CESA register, the ENS bit of CIRQ is set to 1. When the EIT bit of CCM_1 register is equal to 0, it can be occurred the only one time during storing one frame data. This operation is enabled when UES bit of CCM_1 register is set to 1. Refer to CCM_1 register on page 9-228.

CIF R2Y Configuration (CR2Y) 0xF0230074

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											FMT				EN

FMT [4:1]				Input format
0	0	0	0	16bit 565RGB (RGB sequence)
0	0	0	1	16bit 565RGB (BGR sequence)
0	1	0	0	16bit 555RGB (RGB-garbage)
0	1	0	1	16bit 555RGB (BGR-garbage)
0	1	1	0	16bit 555RGB (garbage-RGB)
0	1	1	1	16bit 555RGB (garbage-BGR)
1	0	0	0	8bit 565RGB (RGB sequence)
1	0	0	1	8bit 565RGB (BGR sequence)
1	1	0	0	8bit 555RGB (RGB-garbage)
1	1	0	1	8bit 555RGB (BGR-garbage)
1	1	1	0	8bit 555RGB (garbage-RGB)
1	1	1	1	8bit 555RGB (garbage-BGR)

EN [0]	R2Y enable
0	Disable
1	Enable

CIF Current Y Address (CCYA) 0xF0230078

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Current Y address [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Current U address [15:0]															

FIELD	Description
CCYA[31:0]	Current Y Address.

CIF Current U Address (CCUA) 0xF023007C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Current U address [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Current U address [15:0]															

FIELD	Description
CCUA[31:0]	Current U Address

CIF Current V Address (CCVA) 0xF0230080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Current V address [31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Current V address [15:0]															

FIELD	Description
CCVA[31:0]	Current V Address

CIF Current Line Count (CCLC) 0xF0230084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Line count [15:0]															

FIELD	Description
LCNT[15:0]	Current Line Count

9.4 Effecter Register Descriptions

Table 9.2 Effect Register Map (Base Address = 0xF0230100)

Name	Address	Type	Reset	Description
CEM	0x00	W/R	0x00000000	Effect mode register
CSUV	0x04	W/R	0x00000000	Sepia UV setting
CCS	0x08	W/R	0x00000000	Color selection register
CHFC	0x0C	W/R	0x00000000	H-filter coefficent0
CST	0x10	W/R	0x00000000	Sketch threshold register
CCT	0x14	W/R	0x00000000	Clamp threshold register
CBR	0x18	W/R	0x00000000	BIAS register
CEIS	0x1C	W/R	0x00000000	Image size register
-	0x40	W/R	0x00000000	Reserved
CISA1	0x44	W/R	0x00000000	Source address in Y channel
CISA2	0x48	W/R	0x00000000	Source address in U channel
CISA3	0x4C	W/R	0x00000000	Source address in V channel
CISS	0x50	W/R	0x00000000	Source image size
CISO	0x54	W/R	0x00000000	Source image offset
CIDS	0x58	W/R	0x00000000	Destination image size
CIS	0x5C	W/R	0x00000000	Target scale

CIF Effect Mode (CEM) 0xF0230100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UVS	VB	UB	YB	YCS	IVY	STC	YCL	CS	SKT	EMM	EMB	NEGA	GRAY	SEPI	NOR

UVS [15]	UV Swap
0	U-V-U-V sequence
1	V-U-V-U sequence

VB [14]	V Bias (V channel value offset)
0	Disable
1	Enable

UB [13]	U Bias (U channel value offset)
0	Disable
1	Enable

YB [12]	Y Bias (Y channel value offset)
0	Disable
1	Enable

YCS [11]	YC Swap
0	U(or V)-Y-V(or U)-Y sequence UV sequence is determined by UVS bit.
1	Y-U(or V)-Y-V(or U) sequence UV sequence is determined by UVS bit.

IVY[10]	Invert Y
0	Disable
1	Enable

STC[9]	Strong C
0	Disable
1	Enable

YCL[8]	Y Clamp (Y value clipping)
0	Disable
1	Enable

CS[7]	C Select (Color filter)
0	Disable
1	Enable (color filter)

SKT[6]	Sketch Enable
0	Disable
1	Enable

EMM[5]	Emboss Mode
0	Positive emboss
1	Negative emboss

EMB[4]	Emboss
0	Disable
1	Enable

NEGA[3]	Negative mode
0	Disable
1	Enable

GRAY[2]	Gray mode
0	Disable
1	Enable

SEPI[1]	Sepia mode
0	Disable
1	Enable

NOR[0]	Normal mode
0	Effect mode (effecter is enabled)
1	Normal mode (effecter is disabled)

CIF Sepia UV (CSUV) 0xF0230104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEPIA_U								SEPIA_V							

FIELD	Description
SEPIA_U[15:8]	U channel threshold value for sepia
SEPIA_V[7:0]	V channel threshold value for sepia

CIF Color Selection (CCS) 0xF0230108

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U start								U end							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V start								V end							

FIELD	Description
U_Start[31:24]	Color filter range start point of U channel
U_End[23:16]	Color filter range end point of V channel
V_Start[15:8]	Color filter range start point of U channel
V_End[7:0]	Color filter range end point of V channel

CIF H Filter Coeff. (CHFC) 0xF023010C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								Coeff 0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Coeff 1								Coeff 2							

FIELD	Description
Coeff0[23:16]	Horizontal filter coefficient0 for emboss or sketch.
Coeff1[15:8]	Horizontal filter coefficient1 for emboss or sketch.
Coeff2[7:0]	Horizontal filter coefficient2 for emboss or sketch

CIF Sketch Threshold. (CST) 0xF0230110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								Sketch Threshold							

FIELD	Description
Sketch[7:0]	Sketch threshold

CIF Clamp Threshold. (CCT) 0xF0230114

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								Clamp Threshold							

FIELD	Description
Clamp[7:0]	Clamp threshold

CIF Bias Register (CBR) 0xF0230118

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								Y bias							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U bias								V bias							

FIELD	Description
Y_BIAS[23:16]	Y value offset
U_BIAS[15:8]	U value offset
V_BIAS[7:0]	V value offset

CIF Effect Image Size (CEIS) 0xF023011C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					HSIZE										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					VSIZE										

FIELD	Description
HSIZE[26:16]	Horizontal size of input image
VSIZE[10:0]	Vertical size of input image

9.5 Scaler Register Descriptions

Table 9.3 Scaler Register Map (Base Address = 0xF0230200)

Name	Address	Type	Reset	Description
CSC	0x00	W/R	0x00000000	Scaler configuration
CSSF	0x04	W/R	0x00000000	Scale factor
CSSO	0x08	W/R	0x00000000	Image offset
CSSS	0x0C	W/R	0x00000000	Source image size
CSDS	0x10	W/R	0x00000000	Destination image size

CIF Scaler CTRL (CSC) 0xF0230200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved															EN

EN [0]	Scaler Enable
0	Disable
1	Enable

CIF Scaler SCALE Factor(CSSF) 0xF0230204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		HSCALE													
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0		VSCALE													

FIELD	Description
HSCALE [29:16]	Horizontal scale factor
VSCALE [13:0]	Vertical scale factor

HSCALE = SRC_HSIZE * 256 / DST_HSIZE
VSCALE = SRC_VSIZE * 256 / DST_VSIZE

CIF Scaler Source Offset (CSSO) 0xF0230208

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				H_OFFSET											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0				V_OFFSET											

FIELD	Description
H_OFFSET [27:16]	Horizontal offset
V_OFFSET [11:0]	Vertical offset

CIF Scaler Source Size (CSSS) 0xF023020C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				H_SIZE											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0				V_SIZE											

FIELD	Description
H_SIZE [27:16]	Horizontal size in source image
V_SIZE [11:0]	Vertical size in source image

CIF Scaler DST_Size (CSDS) 0xF0230210

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				H_SIZE											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				V_SIZE											

FIELD	Description
H_SIZE [27:16]	Horizontal size in destination image
V_SIZE [11:0]	Vertical size in destination image

9.6 User guide for CAMIF Setting

9.6.1 External Camera Module Interface (CAMIF)

Figure 9.14 explains the interconnection diagram between TCC8900 and the camera sensor. In a normal camera sensor, a pixel clock is sent out by using an oscillator or external clock input as a clock source. According to the camera sensor, an input clock itself can be used or a divided input clock can be used.

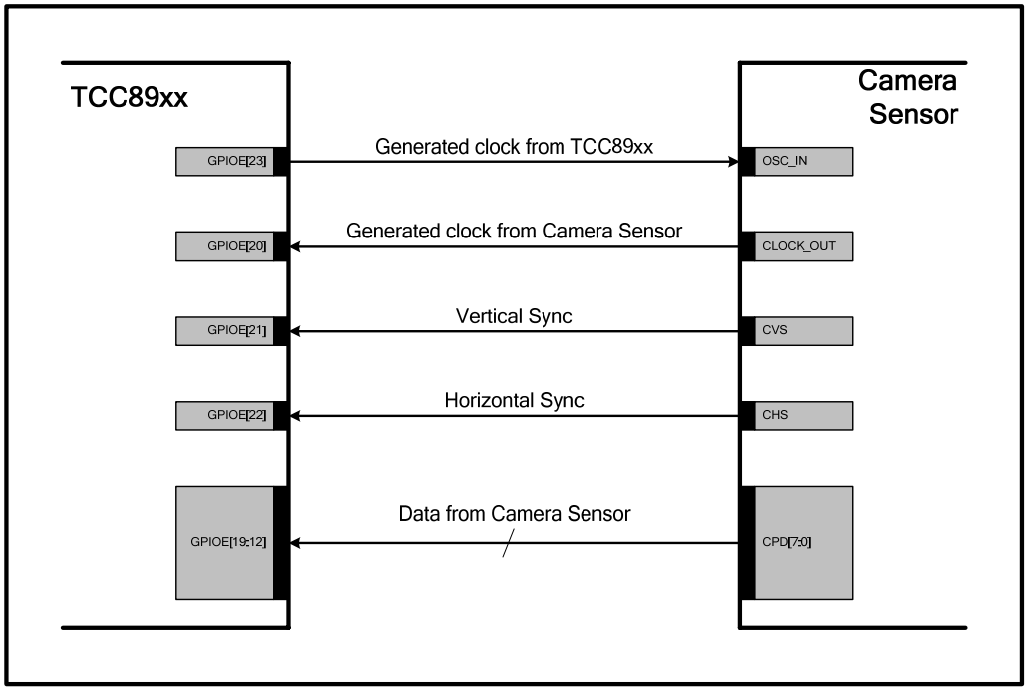


Figure 9.14 Interconnection between TCC8900 and Camera Sensor

Figure 9.15 shows the relation between each clock. The clock outputted from the Sensor is diagramed under the assumption that it is double of the input clock of the Sensor (master clock of the Sensor). In terms of the relation between each Clock, F_{SCALER} should be more than twice of FCCKI (base: inputted F_{CCKI}) as marked in the figure (precisely, twice of FCCKI is recommended). F_{IOBUS} should be bigger than at least F_{SCALER}. However, It can be changeable according to the Overlay of the Camera Block and other hardware operations. For this reason, F_{IOBUS} and F_{SCALER} should be additionally adjusted if the Overlay of the Camera Block and other hardwares are used.

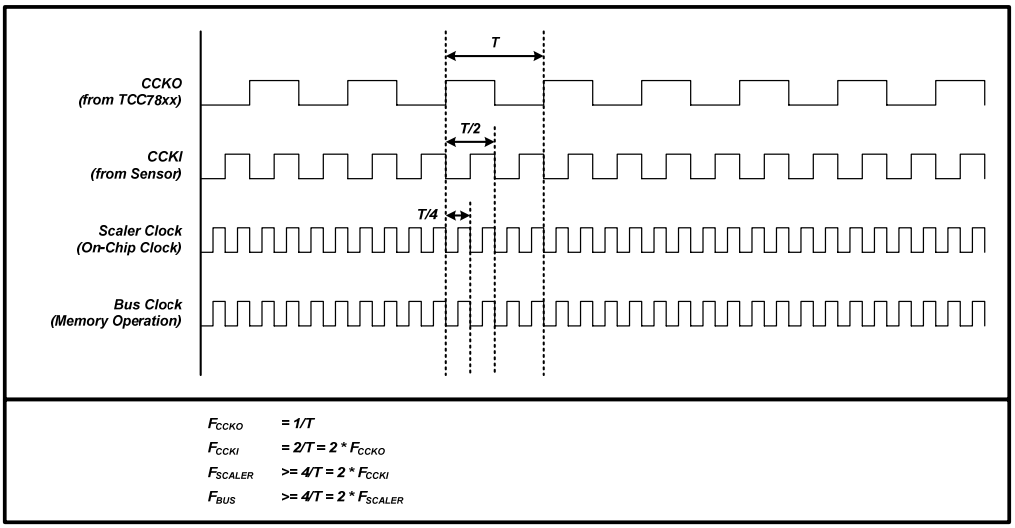


Figure 9.15 Frequency Relations for All the Clocks

9.6.2 Camera Hardware ON/OFF Sequence

Figure 9.16 shows the timing diagram for the output signal of the sensor. CVS means a vertical sync signal or frame valid signal outputted from the camera sensor. CHS means a horizontal sync or line valid signal.

At the moment when the Camera Sensor is Turned-on, a signal is continuously outputted at the timing as Figure 9.16.

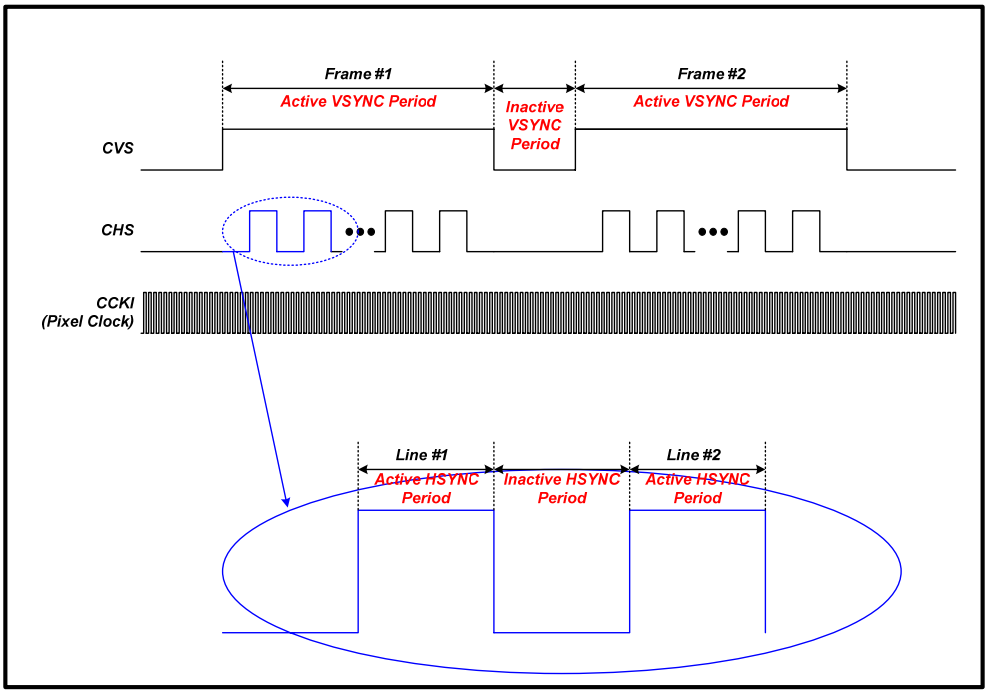


Figure 9.16 Frequency Relations for All the Clocks

Figure 9.17 explains Turn-On Sequence of Camera Interface Hardware. Before Turning on, the software of Camera Interface Hardware Block should be reset. When resetting is carried out, after the software reset register inside CKC block enters into the reset state, a camera clock and scaler clock should be set and software reset register inside CKC block should be released.

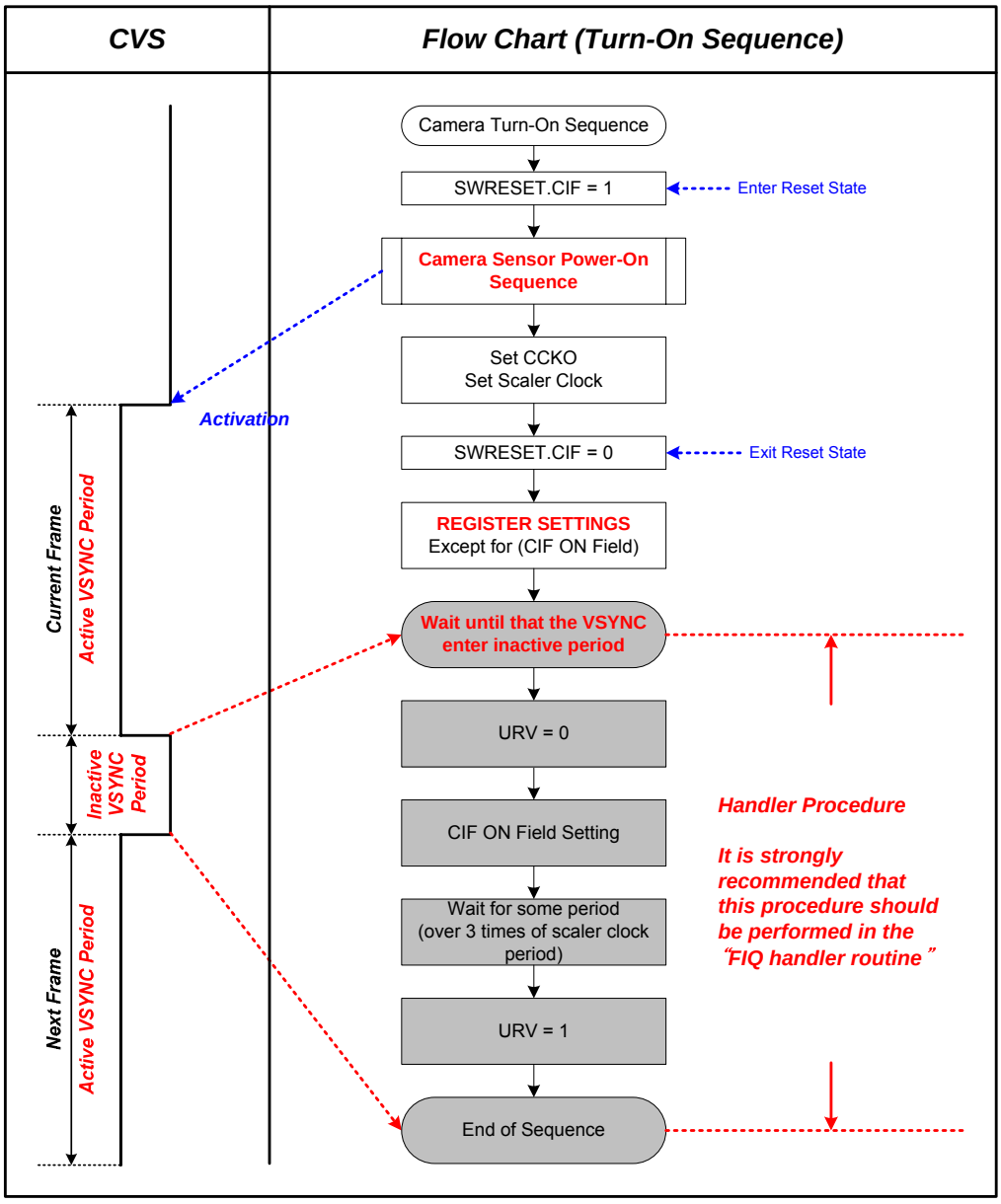


Figure 9.17 Camera Interface Hardware Turn-On Sequence

Figure 9.18 shows Turn-Off Sequence of the Camera Interface Hardware. Turning-off the camera block in TCC8900 should be done when VSYNC is in an Inactive period. If VSYNC is Turned-Off when it is Active, the remained dummy data in the memory buffer could be saved to the unintended memory.

In Turn-On Sequence and Turn-Off Sequence, a signal for an external VS should not be referred to VP or VN interrupt source inside of CAMIF hardware block. Make sure to use External Interrupt Source. – Refer to Port Multiplexor Chapter.

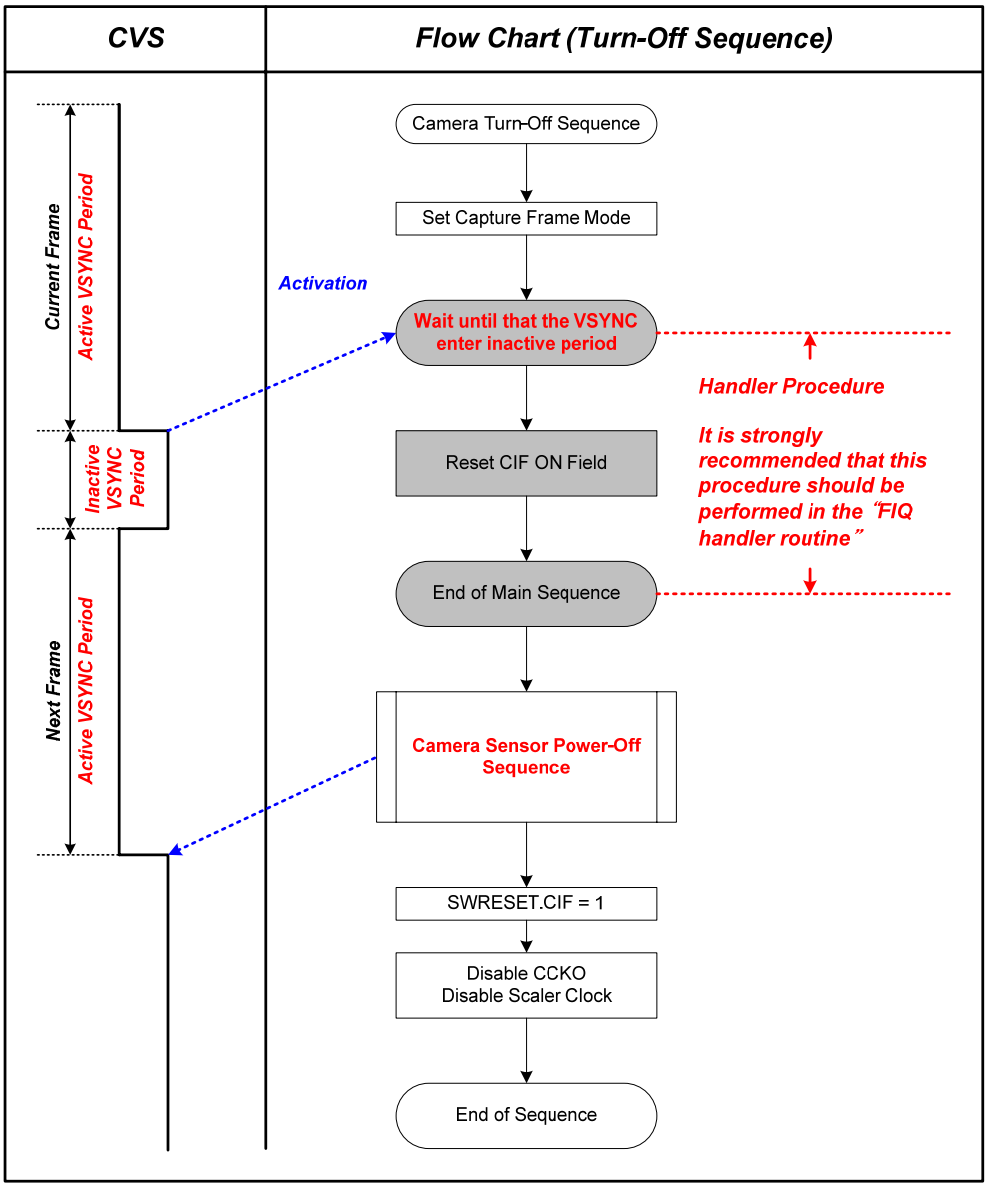


Figure 9.18 Camera Interface Hardware Turn-Off Sequence

Depending on the occasions, when Preview Mode is converted to Capture Mode in a Camera Sensor where Preview Mode and Capture Mode are separate.

9.6.3 Timing Tuning for Camera Sensor

A Camera Sensor varies by companies and pixels. Basically, it operates like Figure 9.14. However, in order to be connected to TCC8900, the below instructions should be followed for the normal operation of the CIF in TCC8900.

Figure 9.19 shows the inside scaler operation for the waveform from the Camera Sensor. Since the Scaler inside TCC8900 uses a 2 line vertical buffer for line filtering, HSYNC(CHS) signal inputted from outside occurs maximum 2 lines later. Therefore, timing should be set with the 2 line delayed operation considered.

In terms of Falling edge (a signal entering into an Inactive state) and Rising edge (a signal entering into an Active state) in CVS signal, it is designed that the preparation for the next Frame is done at the falling edge. The following is the reason for this. Overlay function is included in the Camera Interface Hardware. In order for Overlay DMA to operate normally for the first Pixel of the first Line, as soon as CVS enters into an Inactive state, the operation to process the next frame starts.
For this reason, T_{H2V} time should be obtained in order that Stored-One-Frame (SOF) Interrupt happens normally. In terms of T_{H2V} time, due to its Scaler operation structure, at least more than 2 Line time margin should be obtained for the normal operation of the CIF block in TCC8900.

In addition, T_{H2V} is relevant to an Effector block, the below chapter “Effector User Guide” explains this in details.

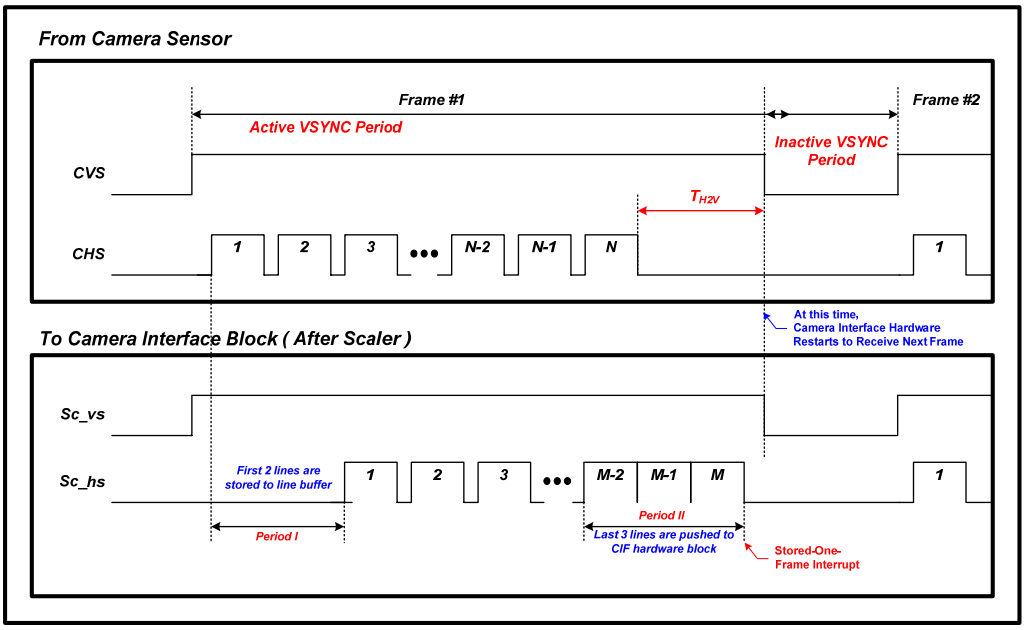


Figure 9.19 Operating Timing Diagram (Correct)

Figure 9.20 illustrates the timing diagram when an abnormal operation occurs. As can be seen in the figure, since T_{H2V} is too short, if Restart Event occurs at T_{fall} in the middle of processing (M-1) and (M) lines, this leads to that the Scaler and CIF hardware operation enters into an abnormal state., Therefore, Stored-One-Frame Interrupt does not happen.

Processing Camera Sensor Timing like Figure 9.20 is explained in the Zoom function.

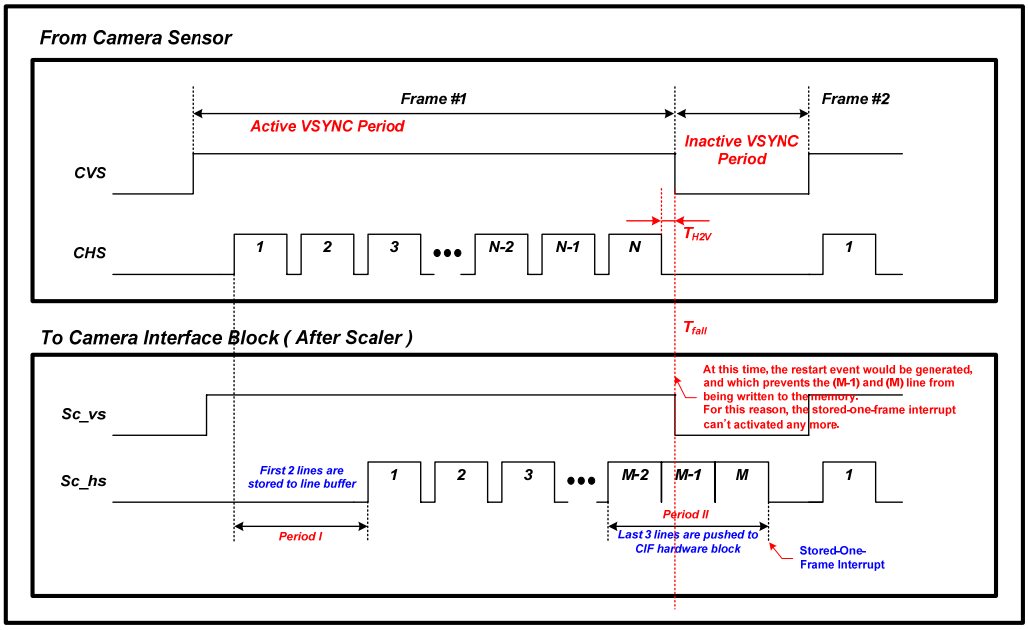


Figure 9.20 Operating Timing Diagram (Incorrect)

9.6.4 Zooming Control

Since there is a built-in Scaler in the Camera Interface Hardware of TCC8900, the inputted data from the Sensor can be Zoomed-In and Zoomed-Out.

In Figure 9.21, there is a basic image frame provided in the Camera. If the horizontal and vertical line refer to HSIZE and VSIZE., the red colored part represents Zoom-Out(Down Scale, it might not be actual Zoom-Out). The blue colored part means Zoom-In(Up-Scale). Generally, Zoom-In Region can be located anywhere within the Camera Frame since the location does not affect the operation of the scaler. However, according to the feature of a camera, Zoom-In and Zoom-out are done at the center.

If image data can be inputted from the Camera as Figure 9.19, this is absolutely fine to process the operation like Figure 9.21.

However, something is inputted as Figure 9.20, this could be fine with Zoom-In. However, it could be problematic with Zoom-Out. In the above case, if Zoom-In and Zoom-Out are done after the input data of the Camera Frame is partially windowed as Figure 9.21.

As explained in Figure 9.22, data which is inputted like in Figure 9.20 is fine in Zoom-In (Up-Scale). However, in Zoom-Out(Down-Scale), a problem occurs. Therefore, an imaginary Window should be set and a user should scale the window.
The Start Position of the Window can be the same with that of the Camera Frame. However, if symmetry is considered, 2 pixels on each side of the horizontal line and 2 lines from each ending point of the vertical line are disregarded. Due to 2 lines at the bottom, It is considered there are no data of the last 2 lines at the bottom in Figure 9.20. Therefore, the Scaler can normally operate.

Likewise, 1:1 Scaling is processed in the same way since the Window Frame is Scaled-Up to the Camera Input Frame size if there is no scaling related operation underway.

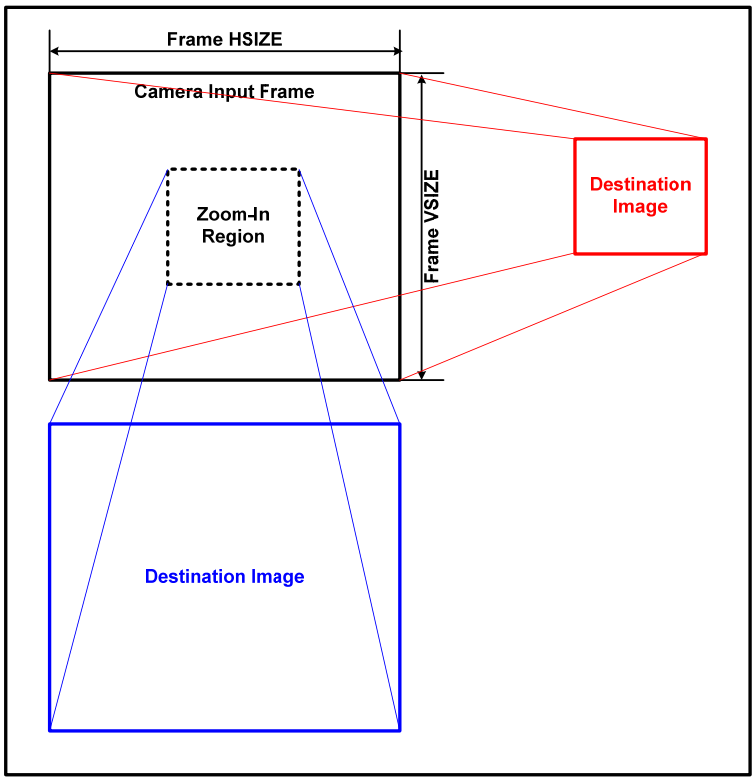


Figure 9.21 Camera Zoom-In(Up-Scale) and Zoom-Out(Down-Scale) - Correct

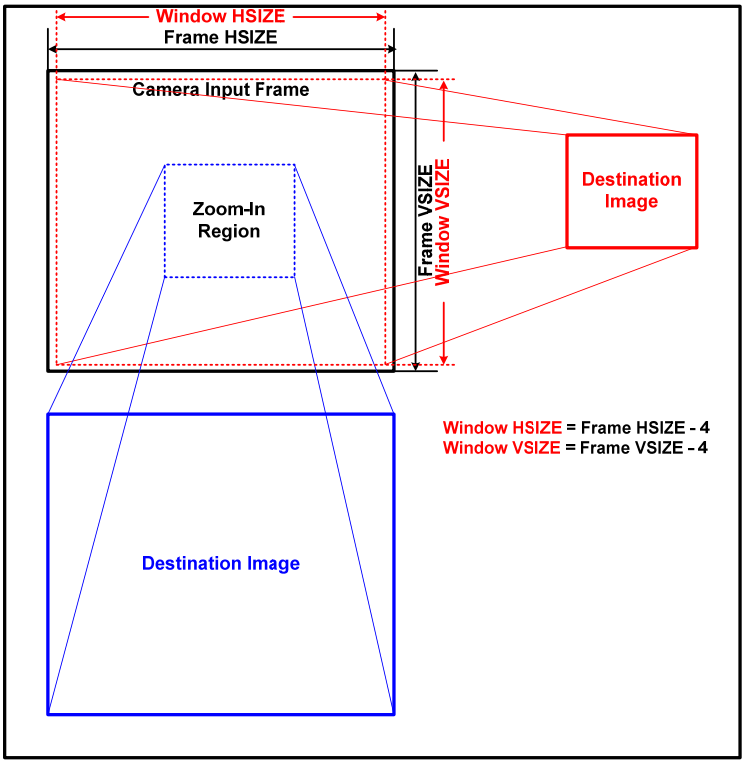


Figure 9.22 Camera Zoom-In(Up-Scale) and Zoom-Out(Down-Scale) - Incorrect

9.6.5 Using One Frame Capture Signal

Whenever one frame is received from the external Camera Sensor, a status signal is generated. The created signals are SOF, VP, VN and VSS field of CIRQ(0x3C) register of the control registers in Camera IF. Each signal creation timings are as below Figure 9.23.

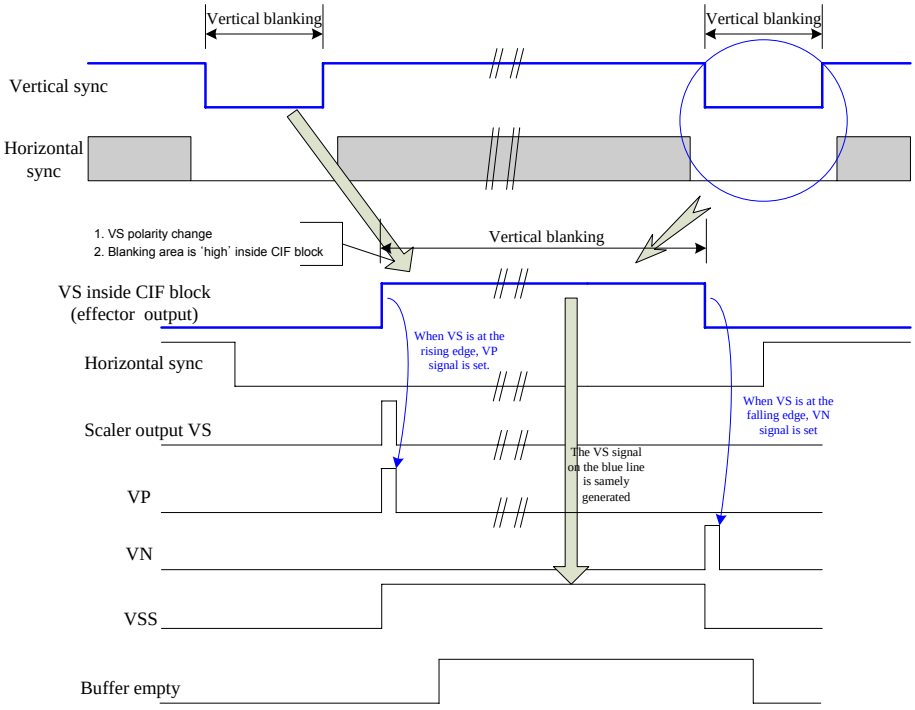


Figure 9.23 The Timing of Status Signals

The CIF operates when VS is 'low' and HS is 'high' in an active region. If VS as the above is 'high' in the active region, it operates by changing polarity. VP signal generates when a vertical sync is at a rising edge. VN signal generates when a inside vertical sync is at an falling edge. VSS signal precisely represents the current VS state. (Since the current VS state is shown, the VS state is 'low' in the active region.)

How to check that one frame is saved in both Preview Mode and Capture Mode is to use SOF signal. SOF signal is generated at the similar location to VP location (refer to Figure 9.26) For Capture Mode usage, refer to “Capture” section of the below chapter.

9.6.6 Register Update

To apply the saved register as a setting value in Camera Interface Hardware to an actual operation is called register update. There are two in register update timing. One is update is done as soon as data is written to the register. The other is update is done when the next VS is rising after data is written to the register. (Refer to CIF_Effector timing in Figure 9.24)

According to URV field (bit 30) of CIRQ register (0x38), either the first or the latter is selected.

When URV field is ‘1’, if VS is rising, register data is updated. When URV field is ‘0’, update is done as soon as register data is written.

A special attention should be given When URV field is used with Camera IF On/Off.
Before Camera IF is On, URV field should not be advancely set. This is because it is concerned with sensor operation timing. (Refer to Figure 9.17, Figure 9.18)

a. When Camera IF is on

After the register values of Camera IF are all set, the next sensor should be on and ON field of CIF should be set. Then, set URV field to ‘1’.

b. When Camera IF is off

After ON field is cleared, Clear URV field to ‘0’.

Likewise, 1:1 Scaling is processed in the same way since the Window Frame is Scaled-Up to the Camera Input Frame size if there is no scaling related operation underway.

9.6.7 Capture

When a user captures the data inputted the Camera sensor, it works based on timing diagram as Figure 9.24. As Figure 9.24, if a capture signal is inputted, from the data of the next frame is saved. In the same way, if 1 frame is saved, the next frame keeps being skipped until the capture signal is ‘low’ and VP or SOF signal notifying that saving 1 frame is completed, is generated. (However, inputted VS can be continuously monitored in the register.)

There is time lapse until writing operation is transferred to the memory. It varies depending on where or not a scaler is used. When the scaler is used, 2 line + 120*PCLK exists in “first data to memory write time”. 2 lines are occurred by the delay of the line buffer of the scaler. 120*PCLK is the cycle of input-to-memory.

“Last data to memory write time” is around maximum 50*PCLK(base: PCLK). However, this is affected by the frequency of SCLK and HCLK. Therefore, the transfer can be completed even though Last data to memory write time is under 50*PCLK since.

(Note : refer to upper chapter “zooming control”)

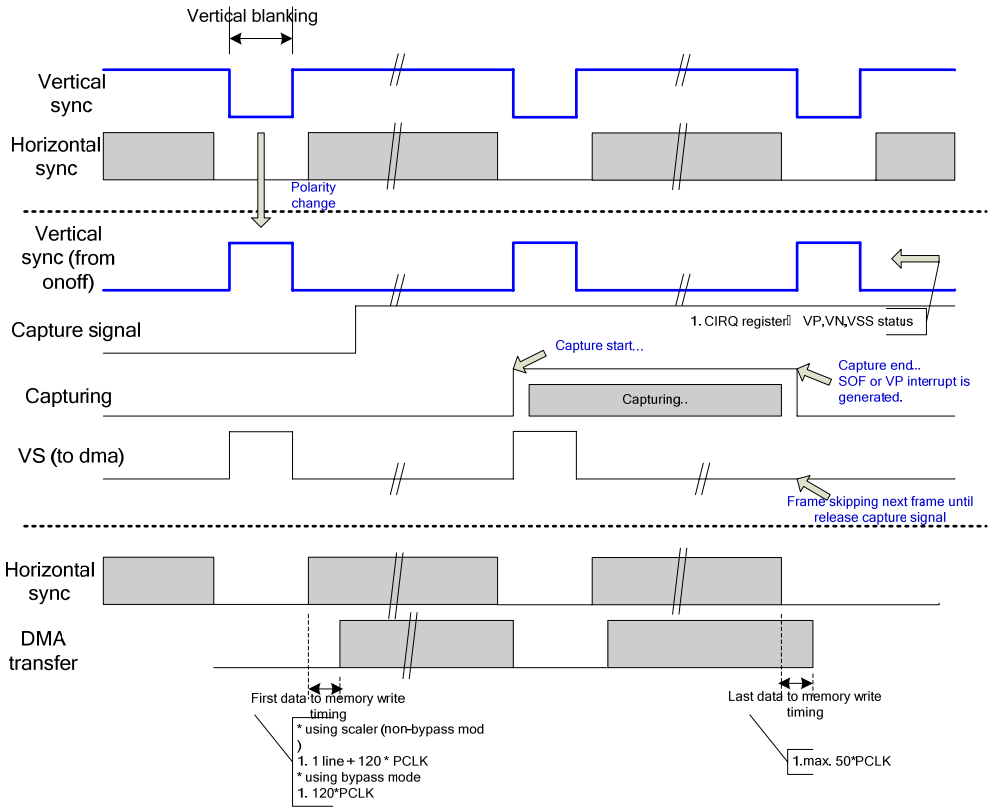


Figure 9.24 Capture Timing Diagram

Figure 9.25 represents Capture Sequence. As can be seen in the figure, check the first frame is properly inputted after turning on CIF while CVS is monitored. If SOF INT is set while SOF is used as an Interrupt source, set CAP field (bit 0 of 0x0060) when VP signal is 'high' in the handler routine. At this time, VP and SOF should be set almost at the same time. As soon as CAP field is set, 1 frame is captured.

If 1 frame capturing is completed, SOF INT is set likewise. (the second SOF waitingbox in the figure). If the SOF INT is set, the handler routine turns off CIF and processes the captured images (post processing box). JPEG Encoding, MPEG Encoding, Effect..ect are the examples about this.

(For more detailed information, refer to Figure 9.25 Flow chart)

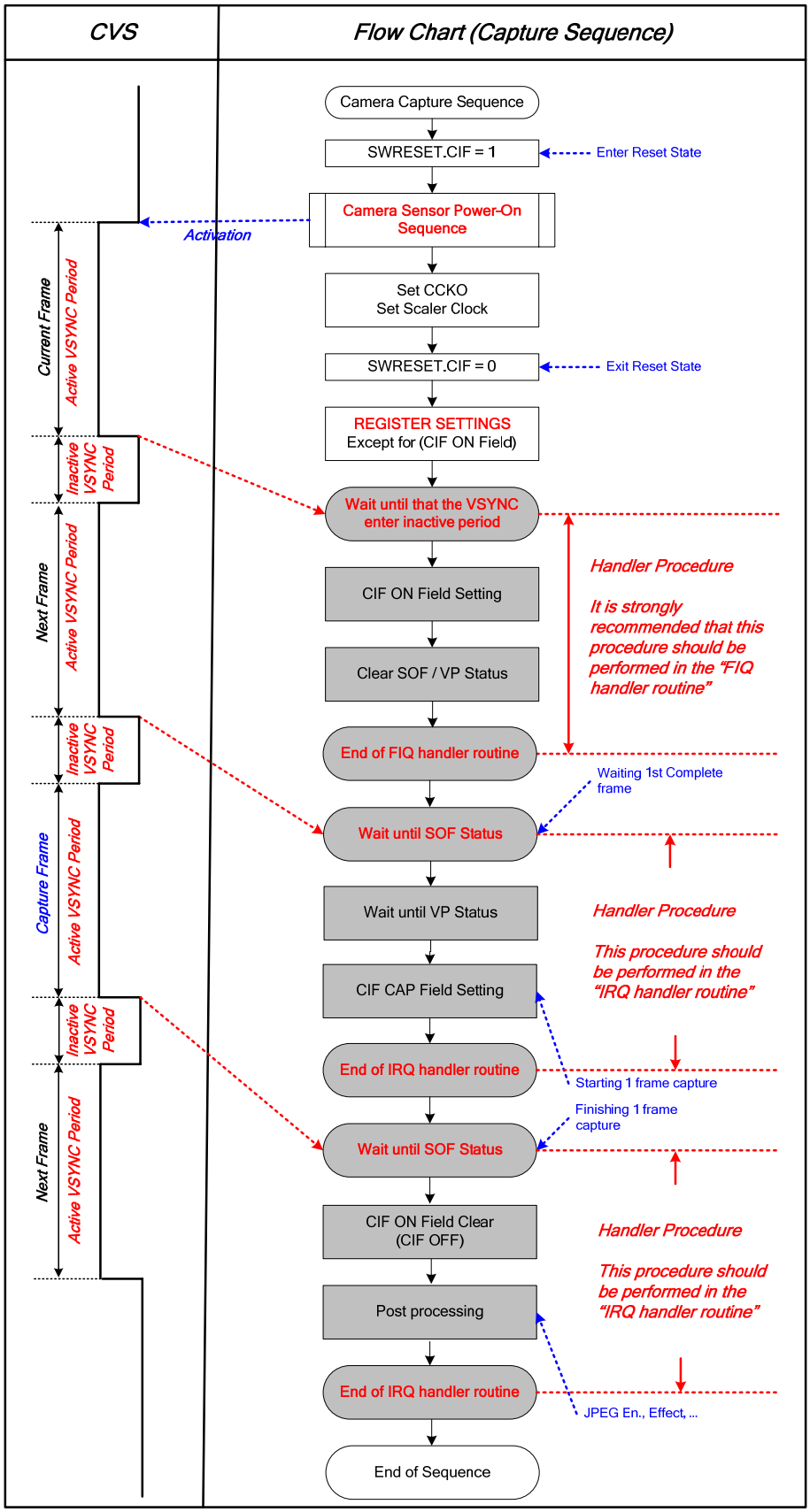


Figure 9.25 Camera Interface Hardware Capture Sequence

9.6.8 Timing and Status Diagram by CAMIF HS/VS Block

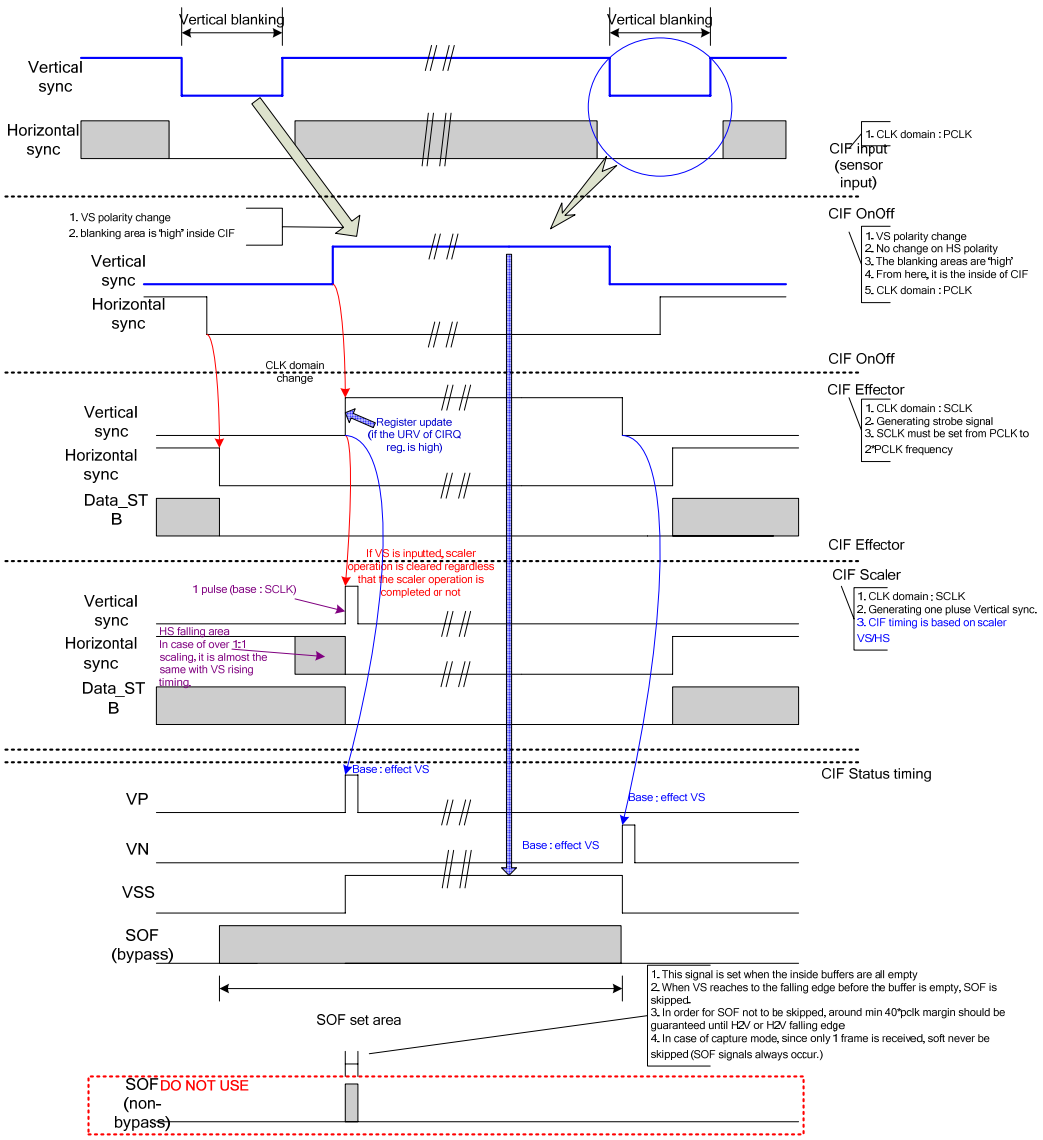


Figure 9.26 CAMIF HS/VS Timing Diagram

9.6.9 Effecter User Guide

Guide 1. Embossing/Sketch effector

When Embossing/Sketch effect is used, filter coefficient should be normalized. The best result could come out with coeff0=0xff(-1), coeff1=0x00 and coeff2=0x02. When 0x80 is set as sketch Th value, the best drawing effect occurred. After Sketch effect, an image has only 0 and 255 as Y component. This leads to increasing of a file size in JPEG encoding since compression is less efficient in Sketch effect compared to other effects.

Guide 2. H2V margin for the normal operation of Effector module

In order for Effector module to normally operate, H2V margin is required. As can be seen in Figure 1.27, 6*PCLK margin is needed. Effector blocks has delay aspects since they internally use a 3-tap filter. In addition, since the effector blocks initialize position registers when VS becomes active, H2V margin should be secured to prevent data at the end of the frames from being lost.

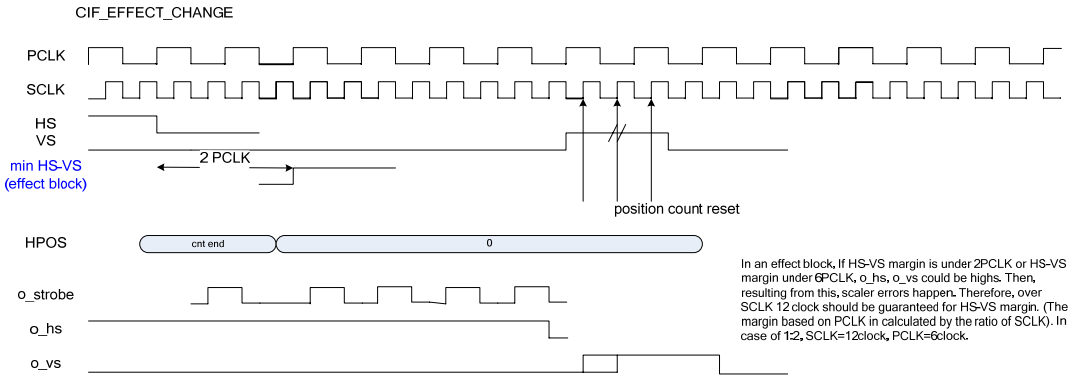


Figure 9.27 CAMIF Effector HS/VS Timing Margin

10 Video and Image Quality Enhancer (VIQE)

10.1 Overview

The VIQE(Video and Image Quality Enhancer) block is high-resolution video/image quality enhancing hardware that facilitates lot of video/image processing features. Aimed primarily at digital flat high-resolution display, it provides advanced and efficient video improvement solutions. In addition to the high quality, the pipeline architecture with specific and characteristic structure can provide minimized memory bandwidth requirement and dynamic module interoperations.

The main functions of the VIQE are listed below.

- De-interlacing
  - Edge-based/Motion-adaptive
  - File-mode processing
  - Judder Cancellation
- Remove Noise
  - Temporal/Spatial Engines
- Edge Enhancing
- Noise Measurement
- Histogram Measurement
- Contrast Enhancing
- Color Gamut Mapping

The interlaced video sources are spreading very widely, mainly for broadcasting, film sources and DVD video markets. With the need of interlaced contents to be different, nowadays the main display interface is flat panel display, so the conversion into the progressive content is indispensable. The de-interlacing hardware of VIQE has advanced and efficient features, all of which support pixel-wise processing and are composed of several algorithms corresponded to very various situations making the artifacts. The film mode detection becomes an essential feature in interlaced-to-progressive conversion as a usual de-interlacing for file mode contents tends to make the output to be contaminated and to have reduced resolution spatially and temporally. The film mode detector in the VIQE supports reverse 3:2 pull-down and also compensates pixel-by-pixel the bad-edit problem and the additional text problem having not film frame rate.

To reduce the noise over video sequence effectively, the procedure of discriminating between noise and normal information in using temporal tracing as well as spatial data referencing is necessary. The de-noising hardware in the VIQE uses a novel algorithm with neighboring data in spatial and temporal domain, and controls the strength with recursive path. Also, the VIQE has the specific hardware for reducing the memory bandwidth, and that is important feature because the recursive path for de-noiser may need additional memory bandwidth.

Some of the hardware engine utilized for de-noising procedure can be shared for the edge enhancing, which improves the contrast of the image and can be expected to make the image more clear and vivid through harmonizing the de-noising. Besides, the VIQE provides the programmability of utilizing the hardware filter structure through arbitrary filter coefficients.
Histogram measurement hardware gathers automatically histogram characteristics at each frame or for user-defined duration. This information can be used for image contrast improvements. In addition, the VIQE provides some optional functions for utilizing measured histogram information.

Lastly, color gamut mapping hardware can be used for calibration of color characteristics to be different as display panel devices. Nowadays, the memorial color concept such as skin color, blue sky, and green grass becomes more important. The color gamut mapping hardware provides the flexibility to apply various gamut mapping algorithms with LUT-based programmability. Easily accessible software tool is provided.

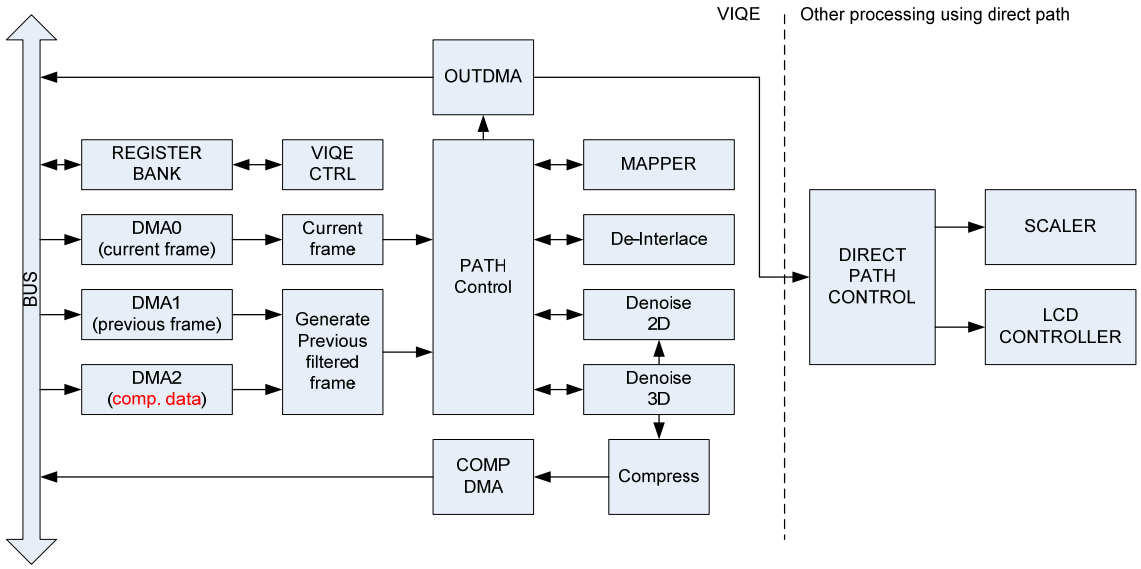


Figure 10.1 VIQE Block Diagram

The VIQE is composed to the major five modules, or DMA ports, de-Interlacer, de-noiser, spatial universal filter, and pixel-mapper. Each module can share the hardware resources and operates independently with some limitations. Any combination of enhancing procedure in the VIQE

Each module is designed to share the hardware resources with other modules to the highest degree, and to have the flexibility to select between performance and quality if the resource sharing is not possible in that combination.

The four DMA channels have specific services for transferring data. Two DMA channels for fetching input frame data, one DMA channel for writing output frame data, and one DMA channel for the recursive data have independent internal FIFO and can operate simultaneously. The output DMA channel can be used as direct path to LCDC as well as external or internal frame buffer interface. The DMAs automatically performs burst word (64-bit) transfers, filling or emptying of the FIFO.

The particular hardware for compressing and de-compressing data is attached to recursive path DMA port. This hardware is used only to reduce memory bandwidth, and the algorithm of compressing is designed to be very suitable for the recursive algorithm in de-noising hardware. This compressor/de-compressor has the flexibility of controlling the strength and performance of compressing.

10.2 De-interlacing Module

The objective of de-interlacing is to convert interlaced scanning into progressive scanning, overcoming the intrinsic nature of the interlaced video sampling structure. In interlaced scanning method, if a fast moving part exists between two fields resulting annoying jitters between lines, and an edges especially along horizontal direction exists, that performance degradation of de-interlacing becomes more noticeable. In other words, there is a theoretical limitation in high frequency components temporally or spatially.

The de-interlacing module in the VIQE has the following features.

- Pixel-wise processing
- Edge directional interpolation
- Motion adaptive processing
- 3:2 pull-down film mode detection
- Bad edition/judder detection

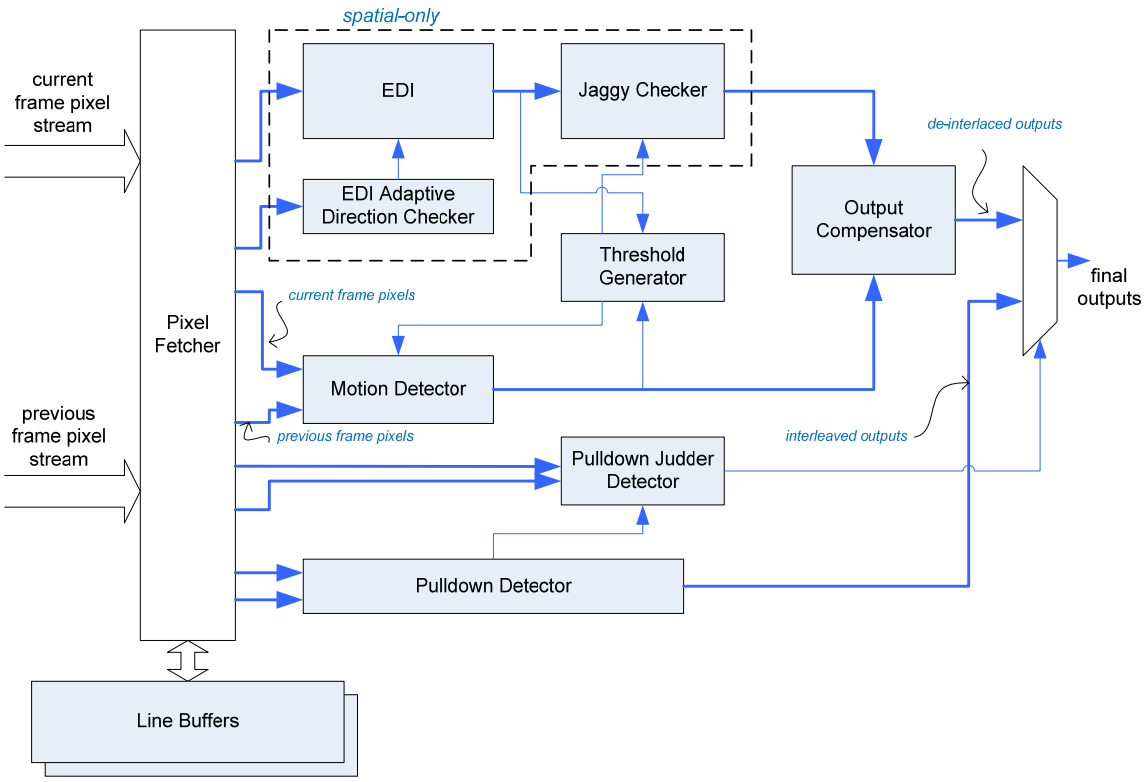


Figure 10.2 De-Interlacer Block Diagram

De-interlacing hardware is composed of several sub modules to implement various algorithms for compensating and coping with the serious negative effects from conversion scanning. Figure below describes the internal block diagram of de-interlacing module. All modules including de-interlacing module in the VIQE are pipelined in inter-module as well as inner-module, and input/output interfaces consist of actual pixel data stream and timing control signals.

The EDI, EDI adaptive direction checker, and jaggy checker have functions for spatial de-interlacing, and are default streaming path if the motion detector determines solid existence of motion or if the spatial-only mode in the control register is selected. The threshold generator block can provide the thresholds or parameter values which can be changed dynamically adapting the neighboring pixels instead of fixing as user-defined values. Motion detector decides whether the current pixel is moving or not through examination of neighboring pixels spatially or temporally. The failure of detecting motion causes serious visual artifacts, so the output compensator would check the reliability of the result of motion adaptive processing as well as spatial adaptive processing.

The pull-down detector and pull-down judder detector are designed to operate reverse 3:2 pull-down procedure, which consists of three steps in the VIQE. The first is to detect whether this stream is film source or not, and the next step is to check the existence of judder caused by a bad edition or addition of texture at normal display rate and the last step is to interleaving accurate field line or to operate de-interlacing at the detected judder pixels. The detection of film mode takes advantage of statistical information acquired during some frames.

10.3 De-noising Module

The objective of de-noising is to remove the noise or sharpening of edge from a single image or video stream. De-noiser is composed of the temporal de-noiser and spatial one that are supported the sharpening and smoothing. Two blocks of de-noiser can support several data-path connection methods as follows.

- Usage of only temporal path (blue arrow)
- Usage of only spatial path (red arrow)
- Usage of both temporal and spatial path (magenta arrow)
  - 1. In first, smoothing operation, next sharpening.
  - 2. All operation is smoothing. (But it is not recommended.)

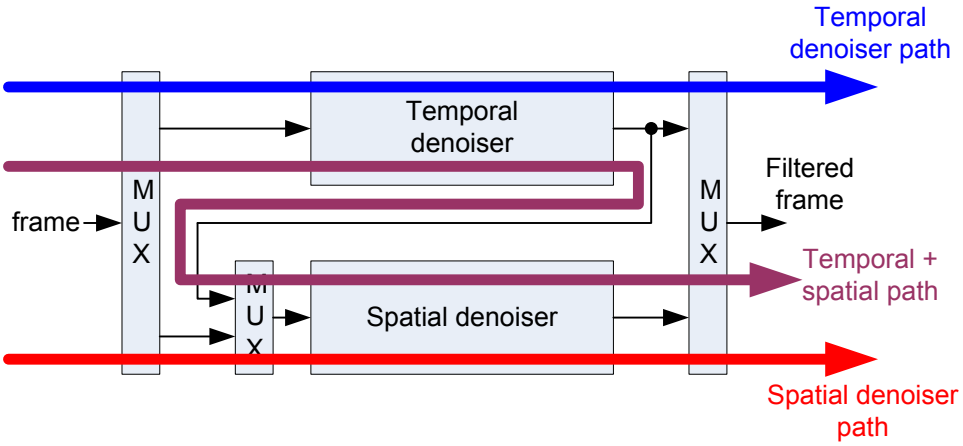


Figure 10.3 De-noiser Block Diagram

10.3.1 Temporal De-noiser

Temporal de-noiser detects and reduces noises through referencing the adjacent pixels in horizontal and vertical spatial domain and tracing the pixels of previous frame in temporal domain. The filtering method of each domain is the method of a kind of recursive filter which is fed on the current input and the immediate filtered data. Figure 10.4 describes the block diagram of the temporal de-noiser.

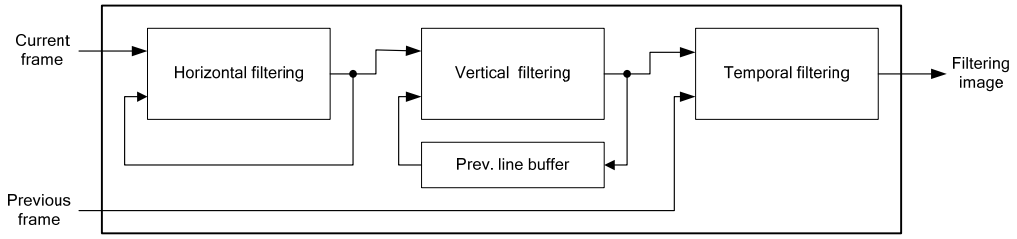


Figure 10.4 Temporal De-noiser Block Diagram

This module needs to store the filtered data for feeding at the following processing because all filter structures have the recursive path. In the temporal domain processing, because the size of frame buffer for storing the previous filtered immediate data is very huge, the special means for reducing memory bandwidth should be considered.

In advance, the differential data between the filtered frame data and original frame data are calculated. In the statistics, because the average of this differential data tends to converge into zero, the size of data to be stored is reduced effectively. Afterward, the differential data are compressed using the specific algorithm suitable for the statistical characteristics of the differential data. These procedures are implemented in fully hardware, and the controllability of adjusting compression strength through register setting is provided.

Figure 10.5 shows the flow of the modified operation.

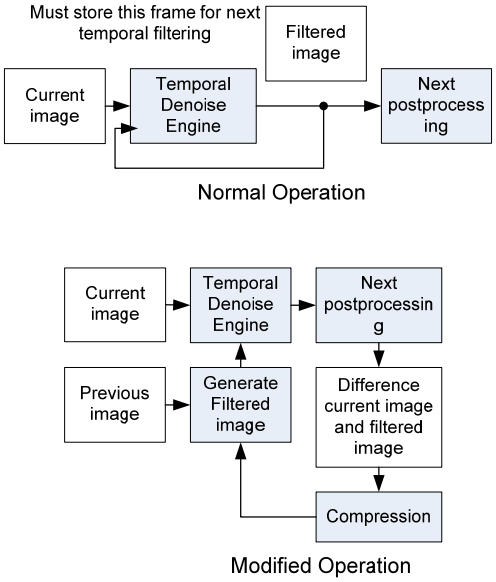


Figure 10.5 Recursive Operation

10.3.2 Spatial De-noiser

The spatial de-noiser without referencing the previous frame data in temporal domain uses only the adjacent pixels of the current pixel to be processed. Figure 10.6 shows the internal architecture in which low-pass filter and high-pass filter share the main data path. The main procedure of this block is to take consideration of the edge directional properties about the current processing window block.

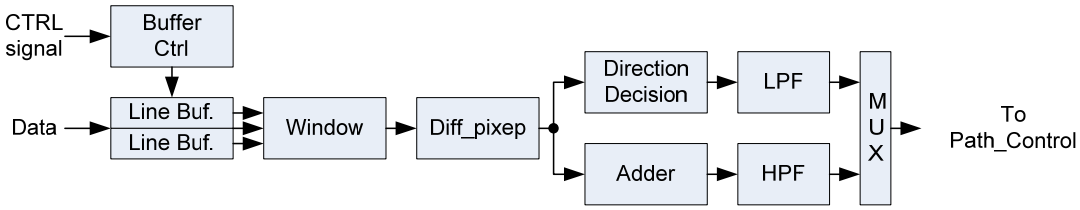


Figure 10.6 Spatial de-noising Block

In LPF mode, the smoothing filter with preserving the edge property in the current window can reduce the noise along the same direction like that of the detected edge. On the contrary, the HPF mode emphasizes the edge effectively.

10.4 RDMA (Read DMA) Module

The RDMA block is the first beginning block in the VIQE, and should be enabled before any operation in the VIQE starts. The RDMA has three channels as Figure 10.7 depicts, and two channels of three are used as normal transferring the current frame and the previous frame which is for the temporal mode in either de-interlacer or de-noiser. The other channel is designed especially for reading the compressed reference data, which is explained in the de-noiser temporal mode. The read data are stored in temporary buffer and are transferred as streaming format similar to CCIR format. To synchronize three kinds of input data, the control signals from each RDMA channel are communicated with the PATH_CONTROL block as the VIQE central block.

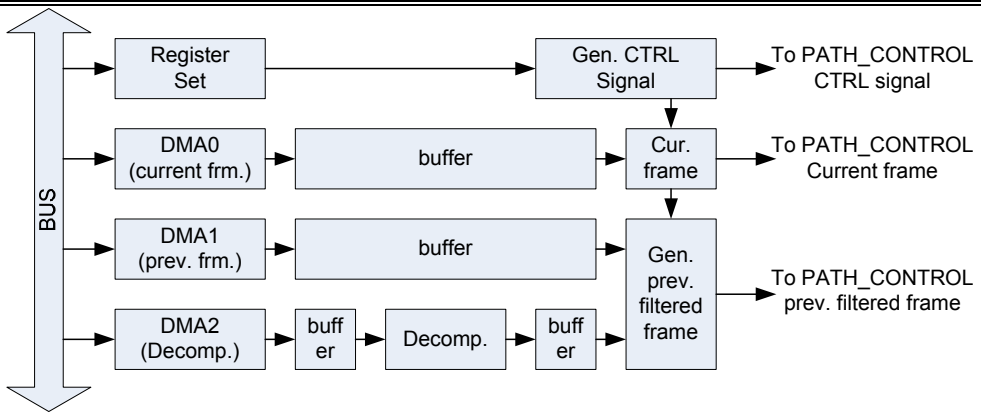


Figure 10.7 RDMA Block Diagram

10.5 ODMA (OUTPUT DMA) Module

The ODMA is for storing the final output pixel into external memory, or to transfer those results directly into display device or other processing blocks such as image scaler. The merit of the direct path is to prevent the additional memory read/write accesses just for external display. Such the on-the-fly operations as this make it possible to process all features of the VIQE without repetitive or unnecessary memory accesses from staring at the RDMA to finishing at the ODMA. Figure 10.8 shows the block diagram of the ODMA connected to external scaler and LCD controller through the direct path controller.

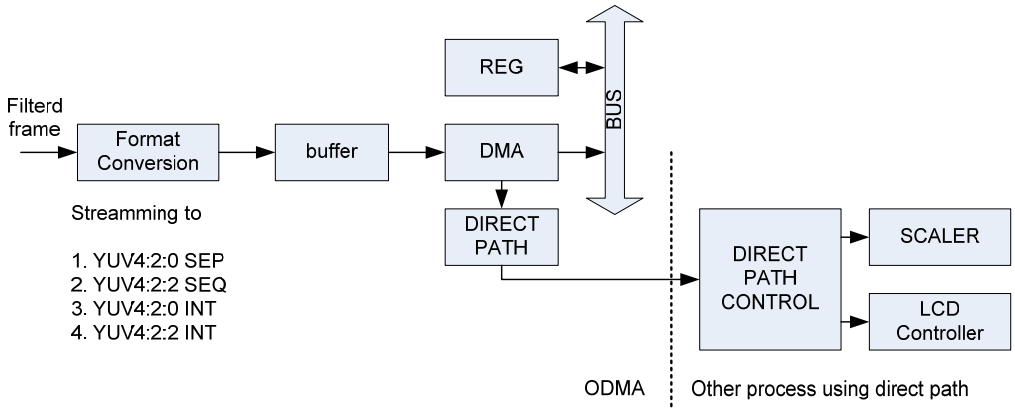


Figure 10.8 ODMA Block Diagram

10.6 Gamut Mapper Module

The gamut mapper module in the VIQE is designed to re-map three color components. The mapping method is based on several pre-programmed tables values from various algorithms and the mapping characteristic is not changed dynamically pixel-wise., but provides the flexibility of modifying the mapping values.

The programming method can be provided as software package.

10.7 Histogram Generator Module

The histogram generator is used to measure histogram of video image. It supports fast histogram processing. It enhances image contrast using automatic equalization of measured histogram.
The features of the histogram generator are as follows.

- Supports up to 1920 * 1080 (HDTV) image size
- Supports segmented histogram measure of raw video image
- Supports fast CDF generation and interpolation of measured histogram
- Supports programmable histogram equalization of video image
- Supports statistical histogram processing of multi-frame image

The following key parameters can be programmed.

- Input image width / height
- Horizontal / vertical measuring frequency
- Size of measured frame
- 16 histogram segment range
- Equalization strength of each segment

The abstract behavior of histogram generator is as follows.

Raw image data is synchronously presented with hsyn signal. vsyn signal generated at hsyn interval time. Horizontal / vertical offset counters are incremented by hsyn and vsyn signal. If offset counters are matched to each offset size, matched offset counter generates the match signal and cleared at next sync signal. If vertical and horizontal match signal is generated, the pixel data is captured and one of sixteen pixel counter is incremented. Pixel counters represent segmented histogram data. Captured pixel data is compared to each segment data. If pixel data is in the range of one segment, pixel counter of that segment is incremented. Histogram measurement can be processed in multi-frame size. Multi_frame_size is defined by the MFRM field in HI_CTRL register.

Generated pixel counter values are divided by Multi_frame_size to average multi-frame histogram data. Averaged pixel counter values are accumulated to make normalized CDF of measured histogram. Normalized CDF data is generated by equations (1)-(4). Sample size is given by external control machine.

$$avg\_pixel\_count[i] = \frac{pixel\_count[i]}{Multi\_frame\_size} \quad (1)$$

$$acc\_data[i] = \sum_{k=0}^{i} avg\_pixel\_count[i] \quad (2)$$

$$sample\_size = (\frac{image\_width}{hoffset\_size}) \times (\frac{image\_height}{voffset\_size} + 1) \quad (3)$$

$$normalized\_CDF[i] = \frac{acc\_data[i]}{sample\_size} \times 256 \quad (4)$$

Normalized CDF data can be used to make LUT. To make image preserving equalization, we make little variation to original image. Following equation shows the generation process of segmented LUT. The eq_scale values represent equalization strength. This value can be different in each segment, and its range is 0 to 128. Larger eq_scale makes stronger equalized CDF data.

$$LUT[i] = (normalized\_CDF[i] - org[i]) \times eq\_scale[i]) >> 7 + org[i] \quad (5)$$

Segmented LUT data is just composed of 16 values which can be set to be arbitrary. Histogram generator interpolates 16 LUT data to 256 LUT data. The missing value between adjacent segments is interpolated linearly. The interpolated data can be used in histogram processing or LUT mapping.

If only the histogram measurement is used, the programming of the hardware is simple. When the automatic mode is selected, the user only read the result registers at the end of one frame processing or at whenever the user wants to read. The result registers are categorized as three groups, one of which is just raw counter value at the specified segments, and the second group is the normalized CDF format, the range of which is 0 ~ 255 and which can be used as the transformation function like the histogram equalization transformation. Both of groups are not fully ranged mapping table, but user-defined segmented mapping table. The last group is the interpolated 256 LUT values, which are calculated from the normalized CDF format and the missing values inside each segment are interpolated linearly. These values can be read in the HI_LUT registers.

Otherwise, if the histogram is configured as the equalization mapping, the programming method is little complex. If the user tends to read the LUT output, the user should read that results only when the frame processing is done. Otherwise the result values would be corrupted. This checking can be through monitoring the status register or controlling the re-start control register.

10.8 Register Description

Table 10.1 VIQE Register Map (Base Address = 0xF0252000)

Name	Address	Type	Reset	Description
CTRL	0x000	R/W	0x00000000	VIQE General Control Register
SIZE	0x004	R/W	0x00000000	VIQE SIZE Register
TIMEGEN	0x008	R/W	0x00000000	VIQE Time Generator Register
LUMADLY	0x00C	R/W	0x00000000	VIQE Luma Delay Register
IMGCONF	0x010	R/W	0x00000000	VIQE Image Configuration Register
IMGFMT	0x014	R/W	0x00000000	VIQE Image Format Register
MISCC	0x018	R/W	0x00000000	VIQE Misc, Control Register
FRMC	0x01C	R/W	0x00000000	VIQE Frame Control Register
INT	0x020	R/W	0x00000000	VIQE Interrupt Register
INTMASK	0x024	R/W	0x00000000	VIQE Interrupt Mask Register
VERSION	0x03c	R	0x4d2b3401	VIQE Version Register

Name	Address	Type	Reset	Description
DI_CTRL	0x080	R/W	0x00000000	De-interlacer Control Register
DI_ENGINE0	0x084	R/W	0x00000000	De-interlacer Engine 0 Register
DI_ENGINE1	0x088	R/W	0x00000000	De-interlacer Engine 1 Register
PD_THRES0	0x08C	R/W	0x00000000	De-interlacer Pulldown Threshold 0 Register
PD_THRES1	0x090	R/W	0x00000000	De-interlacer Pulldown Threshold 1 Register
PD_JUDDER	0x094	R/W	0x00000000	De-interlacer Pulldown Judder Register
DI_MISCC	0x098	R/W	0x00000000	De-interlacer Misc. Control Register
DI_STATUS	0x0A0	R		De-interlacer Status Register
PD_STATUS	0x0A4	R		De-interlacer Pulldown Status Register
DI_REGION0	0x0A8	R/W	0x00000000	De-interlacer Region 0 Register
DI_REGION1	0x0AC	R/W	0x00000000	De-interlacer Region 1 Register
DI_INT	0x0BC	R/W	0x00000000	De-interlacer Interrupt Register

Name	Address	Type	Reset	Description
DN_C_H_Y0	0x100	R/W	0xbffffa4	Temporal De-noiser horizontal coefficient #0 in luminance
DN_C_H_Y1	0x104	R/W	0x15556aaa	Temporal De-noiser horizontal coefficient #1 in luminance
DN_C_V_Y0	0x108	R/W	0xaaaaaaa4	Temporal De-noiser vertical coefficient #0 in luminance
DN_C_V_Y1	0x10C	R/W	0x15556aaa	Temporal De-noiser vertical coefficient #1 in luminance
DN_C_T_Y0	0x110	R/W	0xaaaaaaa4	Temporal De-noiser temporal coefficient #0 in luminance
DN_C_T_Y1	0x114	R/W	0x15556aaa	Temporal De-noiser temporal coefficient #1 in luminance
DN_C_H_C0	0x118	R/W	0xbffffa4	Temporal De-noiser horizontal coefficient #0 in chrominance
DN_C_H_C1	0x11C	R/W	0x15556aaa	Temporal De-noiser horizontal coefficient #1 in chrominance
DN_C_V_C0	0x120	R/W	0xaaaaaaa4	Temporal De-noiser vertical coefficient #0 in chrominance
DN_C_V_C1	0x124	R/W	0x15556aaa	Temporal De-noiser vertical coefficient #1 in chrominance
DN_C_T_C0	0x128	R/W	0xaaaaaaa4	Temporal De-noiser temporal coefficient #0 in chrominance
DN_C_T_C1	0x12C	R/W	0x15556aaa	Temporal De-noiser temporal coefficient #1 in chrominance
DN_STATE0_TEM	0x130	R/W	0x00000000	Temporal De-noiser count states and int. mask
DN_STATE1_TEM	0x134	R	-	Temporal De-noiser count states
DN_DIV_IMG_TEM	0x138	R/W	0x00000168	Temporal De-noiser image divide
DN_C_SPA_Y0	0x140	R/W	0x12320e0a	Spatial De-noiser coefficient #0 in luminance
DN_C_ SPA _Y1	0x144	R/W	0x373c051d	Spatial De-noiser coefficient #1 in luminance
DN_C_ SPA _Y2	0x148	R/W	0x4a0640ff	Spatial De-noiser coefficient #2 in luminance
DN_C_ SPA _Y3	0x14C	R/W	0x003100fb	Spatial De-noiser coefficient #3 in luminance
DN_C_ SPA _C0	0x150	R/W	0x12190805	Spatial De-noiser coefficient #0 in chrominance
DN_C_ SPA _C1	0x154	R/W	0x373c0507	Spatial De-noiser coefficient #1 in chrominance
DN_C_ SPA _C2	0x158	R/W	0x4a0640ff	Spatial De-noiser coefficient #2 in chrominance
DN_C_ SPA _C3	0x15C	R/W	0x003100fb	Spatial De-noiser coefficient #3 in chrominance

Name	Address	Type	Reset	Description
DN_FIFOSTATE	0x160	R/W	0x00000000	De-noiser FIFO states
DN_STATE0_ SPA	0x164	R/W	0x00000000	Spatial De-noiser count states and int.mask
DN_STATE1_ SPA	0x168	R	-	Spatial De-noiser count states
DN_CTRL	0x16C	R/W	0x00000000	De-noiser FIFO and coefficient ctrl
DN_DIV_IMG_ SPA	0x170	R/W	0x00000168	Spatial De-noiser image divide

Name	Address	Type	Reset	Description
RD_IMG0_BASE0	0x180	R/W	0x00000000	RDMA image #0 base address in Y channel
RD_IMG0_BASE1	0x184	R/W	0x00000000	RDMA image #0 base address in U channel
RD_IMG0_BASE2	0x188	R/W	0x00000000	RDMA image #0 base address in V channel
RD_IMG0_OFS	0x18C	R/W	0x00000000	RDMA image #0 address offset
RD_IMG1_BASE0	0x190	R/W	0x00000000	RDMA image #1 base address in Y channel
RD_IMG1_BASE1	0x194	R/W	0x00000000	RDMA image #1 base address in U channel
RD_IMG1_BASE2	0x198	R/W	0x00000000	RDMA image #1 base address in V channel
RD_IMG1_OFS	0x19C	R/W	0x00000000	RDMA image #1 address offset
RD_IMG2_BASE0_0	0x1A0	R/W	0x00000000	RDMA decomp. data #0 base address in Y channel
RD_IMG2_BASE1_0	0x1A4	R/W	0x00000000	RDMA decomp. data #0 base address in U channel
RD_IMG2_BASE2_0	0x1A8	R/W	0x00000000	RDMA decomp. data #0 base address in V channel
RD_IMG2_BASE0_1	0x1AC	R/W	0x00000000	RDMA decomp. data #1 base address in Y channel
RD_IMG2_BASE1_1	0x1B0	R/W	0x00000000	RDMA decomp. data #1 base address in U channel
RD_IMG2_BASE2_1	0x1B4	R/W	0x00000000	RDMA decomp. data #1 base address in V channel
RD_CUR_ADDR0	0x1B8	R	-	RDMA image #0 current address
RD_CUR_ADDR1	0x1BC	R	-	RDMA image #1 current address
RD_CUR_ADDR2	0x1C0	R	-	RDMA decomp. data current address
RD_FIFOSTATE	0x1C4	R/W	0x00000000	RDMA FIFO States
RD_LINE_STATE0	0x1C8	R	-	RDMA count states #0
RD_LINE_STATE1	0x1CC	R/W	0x00000000	RDMA count states #1
RD_CTRL	0x1D0	R/W	0x00000000	RDMA control register
RD_COMP_PL0	0x1D4	R/W	0x00000000	RDMA decomp. data number in Y channel
RD_COMP_PL1	0x1D8	R/W	0x00000000	RDMA decomp. data number in C channel

Name	Address	Type	Reset	Description
CD_BASE0_0	0x200	R/W	0x00000000	Comp. DMA #0 base address in Y channel
CD_BASE1_0	0x204	R/W	0x00000000	Comp. DMA #0 base address in U channel
CD_BASE2_0	0x208	R/W	0x00000000	Comp. DMA #0 base address in V channel
CD_BASE0_1	0x20C	R/W	0x00000000	Comp. DMA #1 base address in Y channel
CD_BASE1_1	0x210	R/W	0x00000000	Comp. DMA #1 base address in U channel
CD_BASE2_1	0x214	R/W	0x00000000	Comp. DMA #1 base address in V channel
CD_CUR_ADDR	0x218	R	-	Comp. DMA current address
CD_STATE	0x21C	R/W	0x00000000	Comp. DMA states
CD_CTRL	0x220	R/W	0x00000000	Comp. DMA control register
CD_HUFF_CNT0	0x230	R	-	Comp. DMA compressed data count in Y channel
CD_HUFF_CNT1	0x234	R	-	Comp. DMA compressed data count in U channel
CD_HUFF_CNT2	0x238	R		Comp. DMA compressed data count in V channel

Name	Address	Type	Reset	Description
OD_BASE0	0x280	R/W	0x00000000	ODMA base address in Y channel
OD_BASE1	0x284	R/W	0x00000000	ODMA base address in U channel
OD_BASE2	0x288	R/W	0x00000000	ODMA base address in V channel
OD_SIZE	0x28C	R/W	0x00000000	ODMA image size
OD_OFS	0x290	R/W	0x00000000	ODMA address offset
OD_CFG	0x294	R/W	0x00000000	ODMA image type
OD_CTRL	0x2A4	R/W	0x00000000	ODMA control register
OD_STATE	0x2A8	R/W	0x00000000	ODMA States

Name	Address	Type	Reset	Description
GM_CTRL	0x400	R/W	0x00000000	Gamut-mapper Control Register
GM_STATUS	0x404	R/W	0x00000000	Gamut-mapper Status Register
GM_INT	0x41C	R/W	0x00000000	Gamut-mapper Interrupt Register

Name	Address	Type	Reset	Description
HI_CTRL	0x600	R/W	0x00000000	Histogram Control Register
HI_STATUS	0x604	R/W	0x00000000	Histogram Status Register
HI_CONFIG	0x608	R/W	0x00000000	Histogram Configuration Register

Name	Address	Type	Reset	Description
HI_INT	0x61C	R/W	0x00000000	Histogram Interrupt Register
HI_SEGS	0x620 ~ 0x62C	R/W	0x00000000	Histogram Segments Register
HI_CDFS	0x630 ~ 0x63C	R	0x00000000	Histogram CDF Register
HI_CNTS	0x640 ~ 0x65C	R	0x00000000	Histogram CNT Register
HI_SCALE	0x660 ~ 0x66C	R/W	0x00000000	Histogram Scale Register
HI_LUTS	0x700 ~ 0x7FC	R/W	0x00000000	Histogram LUT Table Register

CTRL (Control Register) 0xF0252000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											FMTDIS	HILUT	HIEN	GMEN	OEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		FDI	DISP		HPFMODE	DEMODE		RCMEN	DN2EN	DN3EN	DIEN	REN2	REN1	REN0	RGEN

FMTDIS [20]	Image Format Automatic Conversion Disable
0	Auto Conversion is Enable by default. If the LCDC direct path is used, the output image format should be 4:2:2 format. This bit by default enables to make the output image format 4:2:2 format.
1	Disable

HILUT [19]	Histogram CDF or LUT Enable
0	Disable
1	This bit enables the main PATH Controller to check the status of Histogram Generator in order to dynamically insert the Histogram Generator into pipeline path. When this bit is only enabled, the interpolation engine or LUT in Histogram Generator can be used.

HIEN [18]	Histogram Generator Enable
0	Disable
1	Enable

GMEN [17]	Gamut Mapper Enable
0	Disable
1	Enable

OEN [16]	OutDMA Enable
0	Disable
1	Enable

FDI [13]	De-interlacer Order Prioty
0	De-interlacer processing shall be done after De-noising processing.
1	De-interlacer processing shall be done before De-noising processing.

DISP [25]	Spatial De-interlacer Only Mode
0	De-interlacer operates using both current frame and previous frame information
1	De-interlacer operates using only current frame information

HPFMODE [10]	HighPass Filter Mode
0	LowPass Filter Mode
1	HighPass Filter Mode

Spatial de-noiser supports high-pass filter and low-pass filter. If the user wants to use the spatial de-noiser, this field should be selected.

DEMODE [9:8]	De-noiser Mode
00	Use only temporal de-noiser
01	Use only spatial de-noiser
1x	Use both of temporal and spatial de-noiser. In this mode, temporal de-noiser is low-pass filter, and spatial de-noiser is high-pass filter.

RCMEN [7]	Recursive Pixel Compressor Enable
0	Disable
1	This bit turns compressor/decompressor on for recursive pixel data. This feature is meaningful only when RDMA3 port is enabled and the temporal de-noiser is enabled.

DN2EN [6]	Spatial De-noiser Enable

0	Disable
1	Enable

DN3EN [5]	Temporal De-noiser Enable
0	Disable
1	Enable

DIEN [4]	De-interlacer Enable
0	Disable
1	Enable

REN2 [3]	RDMA 2 Enable
0	Disable
1	This bit starts the RDMA2 to transfer the frame data. This bit is meaningless unless RGEN bit is enabled.

This channel 2 of RDMA transfers the compressed differential data of previous filtered frame and previous frame. This bit should be set when the temporal de-noiser mode is on.

REN1 [2]	RDMA 1 Enable
0	Disable
1	This bit starts the RDMA1 to transfer the frame data. This bit is meaningless unless RGEN bit is enabled.

This channel 1 of RDMA transfers the previous frame. This field should be set when the temporal mode of either de-noiser or de-interlacer is on.

REN0 [1]	RDMA 0 Enable
0	Disable
1	This bit starts the RDMA0 to transfer the frame data. This bit is meaningless unless RGEN bit is enabled.

This channel 0 of RDMA should be enabled when the VIQE is operated.

RGEN [0]	Global RDMA Enable
0	Disable
1	Enable

SIZE (Image Size Register) **0xF0252004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					HEIGHT										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					WIDTH										

HEIGHT [26:16]	Image Height
-	Input Image Height by pixel

WIDTH [10:0]	Image Width
-	Input Image Width by pixel

TIMEGEN (Time Generator Register) **0xF0252008**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								H2H							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				H2V				V2H				P2P			

The RDMA transfers the frame data to the processing modules such as de-noiser, de-interlacer, and histogram using the interface similar to CCIR streaming format for on-the-fly operation. This register is used as the configuration of interface timing.

H2H [23:16]	Pixel count of line-end to line-start
-	This field is the pixel count between the ending pixel of the line and the first pixel of the next line. Refer to Figure 10.9,

H2V [11:8]	Line count of line-end to frame-start
-	This field is the line count between the ending of the last line and the starting of the next frame. Refer to Figure 10.9,

V2H [7:4]	Line count of frame-start to line-start
-	This field is the line count between the starting of the frame and the starting of the first line. Refer to Figure 10.9,

P2P [3:0]	Clock count of current to next pixel
-	This field is the cycle count between the valid pixels. Refer to Figure 10.9,

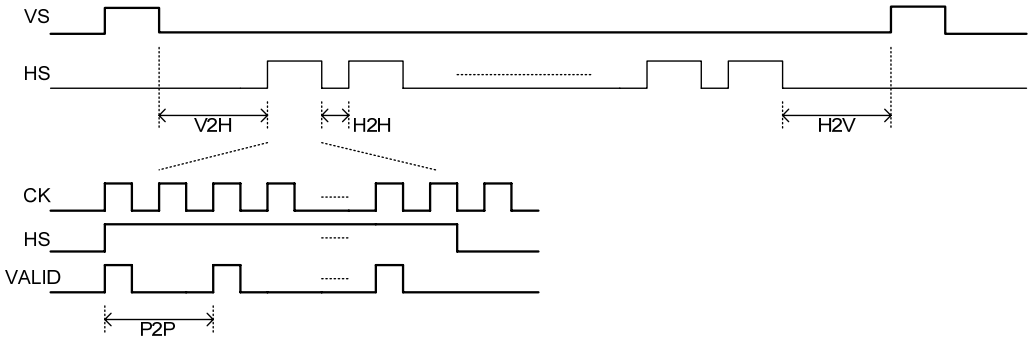


Figure 10.9 Timing Diagram in Internal Operating

LUMADLY (Luminance Delay Register) **0xF025200C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												RDDLY		DNDLY	

Each module has the delay factor for filtering method. For example, spatial de-noiser needs one line delay, de-interlacer needs two line delays. But if the image format is 4:2:0, the output data delay of luminance and chrominance are different because chrominance data is processed once in either of odd or even line. So, the VIQE could not be operated in the on-the-fly mode because of this limitation. For solving this problem, the module is fed on the chrominance data in one to three lines beforehand. Refer to Figure 10.10.

- RDDLY value is 1 : only use the spatial de-noiser
- RDDLY value is 2 : use the de-interlacer or the composed de-interlaces and temporal de-noiser
- RDDLY value is 3 : user the composed de-interlacer and spatial de-noiser.
- DNDLY : DNDLY is the same value in RDDLY, but DNDLY is ‘1’ when the spatial de-noiser is inputted from output data of de-intelacer.

If the image format is not 4:2:0, this register does not need to be set to ‘1’.

RDDLY [3:2]	Luma Delay Value for RDMA
-	Luminance data delay value from output chrominance data.

DNDLY [1:0]	Luma Delay Value for Denoiser
	Luminance data delay value from output chrominance data.

If FMTDIS field of CTRL register is set HIGH, this delay values are set automatically.

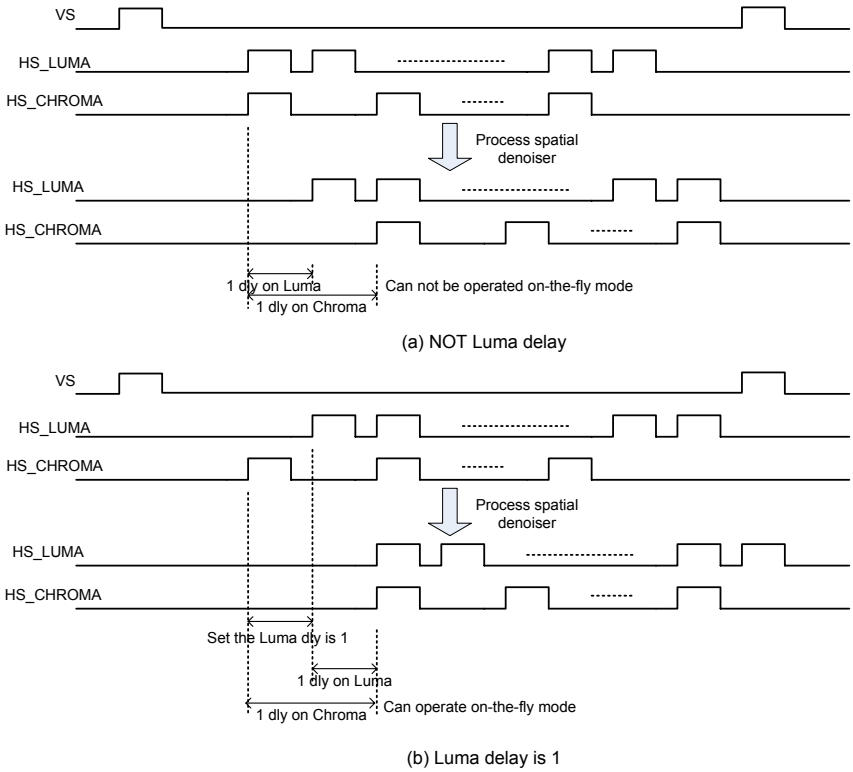


Figure 10.10 Timing Diagram in Luminance Delay Register

IMGCONF (Image Configuration Register) **0xF0252010**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															ODDF

ODDF [0]	Odd Field First Indicator for De-interlacer
0	Even Field First Line in Interlaced Frame
1	Odd Field First Line in Interlaced Frame

This register should be set correctly. Otherwise, de-interlacing output should be deteriorated seriously. This register has an influence on only de-interlacing.

IMGFMT (Image Format Register) 0xF0252014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									FPF	CRT	CNF		IMGFMT		

FPF [6]	First Previous Frame
0	The first previous reference frame is read from the memory buffer described in RDMA1 address register.
1	The first previous reference frame is copied from the first current frame.

CRT [5]	Chroma Line Read Twice
0	Chroma Lines are read just once
1	Chroma Lines are read twice in 4:2:0 format. This bit makes the input image in 4:2:0 format to be manipulated as 4:2:2 input format.

CNF [4]	Chroma Not First Line
0	Chroma Valid Line is in first luminance line in 4:2:0 format
1	Chroma Valid Line is in second luminance line in 4:2:0 format

IMGFMT [3:0]	Image Format
000	YUV4:2:0 Separate mode
001	YUV4:2:2 Separate mode
010	YUV4:2:2 Sequence mode The stored image format is V-Y1-U-Y0(MSB-LSB sequence) in word unit
011	NOT USED
100	YUV4:2:0 interleave 0 mode
101	YUV4:2:0 interleave 1 mode
110	YUV4:2:2 interleave 0 mode
111	YUV4:2:2 interleave 1mode

In interleave mode, the stored image is separated into two channels – luminance(Y) and chrominance(C).

Luminance format is Y3-Y2-Y1-Y0 (MSB-LSB sequence) in a word unit.

Chrominance format is V2-U2-V0-U0 (MSB-LSB sequence) in interleave0 mode, but in the other mode, U2-V2-U0-V0 (MSB-LSB sequence).

MISCC (Miscellaneous Control Register) 0xF0252018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFOF			RESERVED					OVLDIS	RESERVED						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															FRMUPM

FLUSH [31:29]	Internal FIFOs Flush Control
FIFOF[31]	This bit is the de-interlacer Input FIFO Flush control. When it is set, the FIFO shall be flushed.
FIFOF[30]	This bit is the Luma Delayer Input FIFO Flush control. When it is set, the FIFO shall be flushed.
FIFOF[29]	This bit is the Format Converter FIFO Flush Control. When it is set, the FIFO shall be flushed.

OVLDIS [23]	Overlapped Block Processing Disable
0	All Blocks can be processed overlappedly each other.
1	All Blocks cannot be processed overlappedly each other Or, each block processing shall wait until other block processing is done

If this bit is enabled, the processing performance may be declined slightly.

FRMUPM [0]	Frame Restart Register Manual Update
0	Frame Restart Registers is updated automatically. In other words, the next frame processing starts immediately after the current frame processing is done.
1	Frame Registers are updated manually. Only FRMRST bit in FRMCTRL register can make restart the VIQE processing. This bit can be set even during the current frame processing, and after the current frame is done, the next processing shall starts immediately.

The VIQE has the temporal register groups for restoring the configuration changes not to be effected during the current frame processing.

FRMCTRL (Frame Control Register) **0xF025201C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															FRMRST

FRMRST [0]	Frame Restart Register
0	Writing ‘0’ is meaningless
1	VIQE starts to process the next frame data

INT (Interrupt Register) **0xF0252020**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRREG				RESERVED						PM	ODMA	CDMA	DI	DN	RDMA

ERRREG [15:12]	Register Setting Error Interrupt
ERRREG[15]	The interrupt of the reference frame HPF_DI_FIRST setting error. This happens when the HPF mode and DI_FIRST mode are simultaneously enabled.
ERRREG[14]	The interrupt of the reference frame DI_DNS setting error. This happens when the de-interlacer and spatial-only de-noiser mode are simultaneously enabled.
ERRREG[13]	The interrupt of the reference frame RDMA3 setting error. This happens when the de-noiser is enabled but either the RDMA3 or the WDMA is not simultaneously enabled.
ERRREG[12]	The interrupt of the reference frame RDMA2 setting error. This happens when the reference frame data is needed – either temporal de-interlacer or temporal de-noiser mode is enabled, the RDMA2 is not enabled.

PM [5]	Histogram/Gamut Mapper Interrupt
-	An interrupt from Histogram Generator/Gamut Mapper.

ODMA [4]	Output DMA Interrupt
-	An interrupt from Output DMA.

CDMA [3]	Compression DMA Interrupt
-	An interrupt from Compression DMA

DI [2]	De-interlacer Interrupt
-	An interrupt from De-interlacer.

DN [1]	De-noiser Interrupt
-	An interrupt from De-noiser

RDMA [0]	Read DMA Interrupt
-	An interrupt from RDMA.

In above all interrupt register fields, reading ‘1’ means the occurrence of the according interrupt, and writing ‘1’ means the clearing bit of the according interrupt

INTMASK (Interrupt Mask Register) 0xF0252024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRREG				RESERVED						PM	ODMA	CDMA	DI	DN	RDMA

ERRREG [15:12]	Register Setting Error Interrupt Mask
ERRREG[15]	The interrupt mask of the reference frame HPF_DI_FIRST setting error. This happens when the HPF mode and DI_FIRST mode are simultaneously enabled.
ERRREG[14]	The interrupt mask of the reference frame DI_DNS setting error. This happens when the de-interlacer and spatial-only de-noiser mode are simultaneously enabled.
ERRREG[13]	The interrupt mask of the reference frame RDMA3 setting error. This happens when the de-noiser is enabled but either the RDMA3 or the WDMA is not simultaneously enabled.
ERRREG[12]	The interrupt mask of the reference frame RDMA2 setting error. This happens when the reference frame data is needed – either temporal de-interlacer or temporal de-noiser mode is enabled, the RDMA2 is not enabled.

PM [5]	Histogram/Gamut Mapper Interrupt Mask
0	Interrupt Disable
1	Interrupt Enable

ODMA [4]	Output DMA Interrupt Mask
0	Interrupt Disable
1	Interrupt Enable

CDMA [3]	Compression DMA Interrupt Mask
0	Interrupt Disable
1	Interrupt Enable

DI [2]	De-interlacer Interrupt Mask
0	Interrupt Disable
1	Interrupt Enable

DN [1]	De-noiser Interrupt Mask
0	Interrupt Disable
1	Interrupt Enable

RDMA [0]	Read DMA Interrupt Mask
0	Interrupt Disable
1	Interrupt Enable

The interrupt of each module is described in the register fields of each module.

Actually, this register is checked or used only for where the interrupt comes from. The interrupt processing such as clearing interrupts should be checked in each module register.

DI_CTRL (De-interlacer Control Register) 0xF0252080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRIR	FLSFF	BYPASS	RESERVED												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDCLFI	ERRREG			PDPF	PDGTHJ	PDJUD	PDEN	SPJAG	GTHSJ	MRSP	MRTM	STTHDIS			

CLRIR [31]	Clear Internal Registers
0	No meaning
1	Clear Internal Registers in De-interlacer

FLSFF [30]	Flush Synchronizer FIFO
0	No meaning
1	Flush Synchronizer FIFO

BYPASS [29]	Bypass Register
0	Normal Process
1	De-interlacer Bypass Enable If this bit is set, the input data to the de-interlacer will be intact through hardware block.

PDCLFI [15]	Clear Internal Frame Index Register
0	Normal Process
1	Clear Internal Frame Index Register in Pulldown Detector

PDPF [11]	Use Prevention-Flag in Pulldown Detector
0	Use Prevention-Flag algorithm in Pulldown Detector The Prevention-Flag algorithm can reduce the possibility of miss-detection in film mode checking.
1	Don't use Prevention-Flag Algorithm

PDGTHJ [10]	Generate Judder Detector Threshold
0	Use only pre-defined threshold for judder detection.
1	Use generated threshold for judder detection. This is optional feature that the thresholds for judder detection can be generated adaptively by the specific hardware. This feature is recommended strongly.

PDJUD [9]	User Judder Detection in Pulldown Detector
0	Don't use judder detection in Pulldown detector.
1	Use judder detection in Pulldown detector. The judder in the reverse pull-down processing means the artifacts from bad-edit or flowing text that be edited additionally at the normal frame rate. These artifacts can be removed using the judder detection algorithm in the VIQE.

PDEN [8]	Pulldown Detector Enable
0	Pulldown detector is disable.
1	Pulldown detector is enable. Pulldown detector can be used for detecting the film source, which is transferred into digital format using 3:2 pull-down method.

SPJAG [7]	Jaggy Checker in Spatial Deinterlacing
0	Jaggy Checker Disable
1	Jaggy Checker Enable. The jaggy artifact comes from EDI-based spatial de-interlacing. This option can compensate the flaw from the faulty edge direction.

GTHSJ [6]	Generate Jaggy Checker Threshold
0	Use only pre-defined threshold for jaggy checker.
1	Use generated threshold for jaggy checker. This is optional feature that the thresholds for jaggy checker can be generated adaptively by the specific hardware. This feature is recommended strongly.

MRSP [5]	Pixel Output Range Enable in Spatial Processing
0	Don't use output range limitation algorithm.
1	Use output range limitation algorithm. This option can be used only when the spatial de-interlacer only mode(DISP bit in CTRL register) is off.

MRTM [4]	Pixel Output Range Enable in Temporal Processing
0	Don't use output range limitation algorithm.
1	Use output range limitation algorithm. This option can be used only when the spatial de-interlacer only mode(DISP bit in CTRL register) is off.

STTHDIS [3]	Adaptive Stationary Checker Enable
0	Don't use adaptive stationary checker algorithm
1	Use adaptive stationary checker algorithm. The stationary checker in the VIQE does not determine only whether this pixel is stationary or not, but also analyzes the degree of stationary quantitatively. This value can be used in the compensating the outputs. These procedures shall process in the adaptive stationary checker hardware.

DI_ENGINE0 (De-interlacer Engine 0 Register) 0xF0252084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EDIDIRT								DMTSAD							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMTPXL								EDITAD							

EDIDIRT [31:24]	EDI Horizontal Direction Threshold
-	Threshold value for Horizontal Direction Checker in EDI processing

DMTSAD [23:16]	SAD Threshold in Motion Detection
-	Threshold for SAD value in Motion Detection

DMTPXL [15:8]	Pixel Threshold in Motion Detection
-	Threshold for Pixel value in Motion Detection

EDITAD [7:0]	Threshold of EDI Adaptive Direction Checker
-	Threshold value in EDI adaptive direction checker

DI_ENGINE1 (De-interlacer Engine 1 Register) 0xF0252088

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
THSJMAX								THSJMIN							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STTHW				STTHM							EDIADEN	RESERVED			

THSJMAX [31:24]	Maximum Threshold in Jaggy Threshold Generator
-	This value means the maximum boundary for the jaggy threshold generator.

THSJMIN [23:16]	Minimum Threshold in Jaggy Threshold Generator
-	This value means the minimum boundary for the jaggy threshold generator.

STTHW [15:12]	Threshold Weight Parameter of Stationary Detection
-	Threshold Weight Parameter of Stationary Detection Increasing one bit means 0.5 weight addition

STTHM [11:8]	Threshold Multiplier Parameter of Stationary Detection
-	Threshold Multiplier Parameter of Stationary Detection Increasing one bit means 0.5 multiplier addition

EDIADEN [4]	EDI Adaptive Direction Checker
0	Don’t use EDI adaptive direction checker algorithm
1	Use EDI adaptive direction checker algorithm

PD_THRES0 (Pulldown Detector Threshold 0 Register) 0xF025208C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CNTSCO						CO	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UVALDIS			WEIGHT			CNTS									

CNTCO [23:20]	Threshold for counter value of Pulldown Prevention-Flag
-	Threshold for counter value of pulldown Prevention-Flag algorithm

CO [17:16]	Value of Pulldown Detection type
00	Not interleaving, but normal de-interlacing process
01	Reserved
10	Interleaving fields from the previous fields
11	Interleaving fields from the next fields

Through this field, the user can control manually the method of interleaving in the reverse pull-down process.

UVALDIS [15]	Not Use User-defined Threshold Value in Prevention-Flag
0	Use an user-defined threshold value
1	Use hardware-generated threshold value Its range can be controlled by WEIGHT field in PD_THRES0 register.

WEIGHT [12:10]	Weight Value for Threshold of Pulldown Prevention-Flag
-	This means the ratio of relative amount between minimum detection value and the second smaller detection value. This bit is meaningless when UVALDIS is ‘0’.

CNTS [9:0]	Threshold for counter value of Pulldown checker
-	Threshold for counter value of pulldown checker If the number of continuous success of pulldown detection is over that threshold, field interleaving operates instead of de-interlacing.

PD_THRES1 (Pulldown Detector Threshold 1 Register) 0xF0252090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								THRES2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRES1								THRES0							

THRES2 [23:16]	Threshold for pixel difference value of third level
-	This means the threshold value for pixel difference between the current frame and the previous frame.

THRES1 [15:8]	Threshold for pixel difference value of second level
-	This means the threshold value for pixel difference between the current frame and the previous frame.

THRES0 [7:0]	Threshold for pixel difference value of first level
-	This means the threshold value for pixel difference between the current frame and the previous frame.

In normal configuration, threshold value of a level should be larger than that of lower level. Or, threshold value of the first level is smallest of all, and that of the third level is largest.

PD_JUDDER (Pulldown Detector Judder Register) 0xF0252094

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
THSJDMAX								THSJDMIN							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HORLINE				DNLINE				CNTS							

THSJDMAX [31:24]	Threshold maximum boundary for Pulldown Judder Detector
-	This means the maximum boundary of hardware-generated threshold for Pulldown Judder Detector

THSJDMIN [23:16]	Threshold minimum boundary for Pulldown Judder Detector
-	This means the minimum boundary of hardware-generated threshold for Pulldown Judder Detector

HORLINE [15:12]	Horizontal Margin for Judder Elimination Processing
-	This means the left/right margin of the lines detected as having some judder pixels. If its value is larger, the negative effect for judder miss-detection is less.

DNLINE [11:8]	Downward Margin for Judder Elimination Processor
-	This means the downward margin of the lines detected as having some judder pixels. If its value is larger, the negative effect for judder miss-detection is less.

CNTS [7:0]	Threshold for counter of judder pixels
-	This value determines whether the current line is judder line or not. If the pixels detected as judder pixels are more of that value, the current line is determined as the judder line.

DI_STATUS (De-interlacer Status Register) 0xF02520A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		POS_Y											POS_X		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POS_X															BUSY

POS_Y [29:19]	Current Vertical Position
-	This indicates the vertical position in image to be processed currently. This position value is based on luminance resolution.

POS_X [29:19]	Current Horizontal Position
-	This indicates the horizontal position in image to be processed currently. This position value is based on luminance resolution.

BUSY [0]	De-interlacer Busy
-	This indicates that the De-interlacer hardware is processing currently.

All fields of this register are read-only.

PD_STATUS (Pulldown Detector Status Register) 0xF02520A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					MPOFF					CO		STATE			

MPOFF [10:8]	Minimum Position Offset
-	This indicates the information to know the minimum frame position in current operation. This field is used for hardware debugging.

CO [17:16]	Value of Pulldown Detection type
00	Not interleaving, but normal de-interlacing process
01	Reserved
10	Interleaving fields from the previous fields
11	Interleaving fields from the next fields

STATE[3:0]	Current FSM State
-	This indicates the current state of the hardware FSM. This field is used for hardware debugging.

All fields of this register are read-only.

DI_REGION0 (De-interlacer Region 0 Register) 0xF02520A8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN						XEND									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						XSTART									

EN [31]	Region Selection Enable
0	Normal Process
1	Only specific region of the whole image will be processed in de-interlacing

XEND [25:16]	Last Horizontal Position of Region Selection
-	This means the last horizontal position of region selection.

XSTART[9:0]	First Horizontal Position of Region Selection
-	This means the first horizontal position of region selection.

DI_REGION1 (De-interlacer Region 1 Register) 0xF02520AC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						YEND									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						YSTART									

YEND [25:16]	Last Vertical Position of Region Selection
-	This means the last vertical position of region selection.

YSTART[9:0]	First Vertical Position of Region Selection
-	This means the first vertical position of region selection.

DN_C_H_Y0 (Temporal De-noiser Coefficient Horizontal Luminance #0) **0xF0252100**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HY15		HY14		HY13		HY12		HY11		HY10		HY9		HY8	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HY7		HY6		HY5		HY4		HY3		HY2		HY1		HY0	

DN_C_H_Y1 (Temporal De-noiser Coefficient Horizontal Luminance #1) **0xF0252104**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HY31		HY30		HY29		HY28		HY27		HY26		HY25		HY24	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HY23		HY22		HY21		HY20		HY19		HY18		HY17		HY16	

HY#[1:0]	Horizontal filter coefficient in Luminance
0~3	Filter coefficient in temporal de-noiser # : coefficient number (0~31)

DN_C_V_Y0 (Temporal De-noiser Coefficient Vertical Luminance #0) **0xF0252108**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VY15		VY14		VY13		VY12		VY11		VY10		VY9		VY8	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VY7		VY6		VY5		VY4		VY3		HY2		VY1		VY0	

DN_C_V_Y1 (Temporal De-noiser Coefficient Vertical Luminance #1) **0xF025210C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VY31		VY30		VY29		VY28		VY27		VY26		VY25		VY24	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VY23		VY22		VY21		VY20		VY19		HY18		VY17		VY16	

VY#[1:0]	Vertical filter coefficient in Luminance
0~3	Filter coefficient in temporal de-noiser # : coefficient number (0~31)

DN_C_T_Y0 (Temporal De-noiser Coefficient Temporal Luminance #0) **0xF0252110**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TY15		TY14		TY13		TY12		TY11		TY10		TY9		TY8	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TY7		TY6		TY5		TY4		TY3		TY2		TY1		TY0	

DN_C_T_Y1 (Temporal De-noiser Coefficient Temporal Luminance #1) **0xF0252114**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TY31		TY30		TY29		TY28		TY27		TY26		TY25		TY24	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TY23		TY22		TY21		TY20		TY19		TY18		TY17		TY16	

VY#[1:0]	Temporal filter coefficient in Luminance
0~3	Filter coefficient in temporal de-noiser # : coefficient number (0~31)

DN_C_H_C0 (Temporal De-noiser Coefficient Horizontal Chrominance #0) **0xF0252118**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HY15		HY14		HY13		HY12		HY11		HY10		HY9		HY8	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HY7		HY6		HY5		HY4		HY3		HY2		HY1		HY0	

DN_C_H_Y1 (Temporal De-noiser Coefficient Horizontal Chrominance #1) **0xF025211C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HY31		HY30		HY29		HY28		HY27		HY26		HY25		HY24	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

HY23	HY22	HY21	HY20	HY19	HY18	HY17	HY16

HY#[1:0]	Horizontal filter coefficient in Chrominance
0~3	Filter coefficient in temporal de-noiser # : coefficient number (0~31)

DN_C_V_C0 (Temporal De-noiser Coefficient Vertical Chrominance #0) **0xF0252120**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VY15		VY14		VY13		VY12		VY11		VY10		VY9		VY8	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VY7		VY6		VY5		VY4		VY3		HY2		VY1		VY0	

DN_C_V_C1 (Temporal De-noiser Coefficient Vertical Chrominance #1) **0xF0252124**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VY31		VY30		VY29		VY28		VY27		VY26		VY25		VY24	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VY23		VY22		VY21		VY20		VY19		HY18		VY17		VY16	

VY#[1:0]	Vertical filter coefficient in Chrominance
0~3	Filter coefficient in temporal de-noiser # : coefficient number (0~31)

DN_C_T_C0 (Temporal De-noiser Coefficient Temporal Chrominance #0) **0xF0252128**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TY15		TY14		TY13		TY12		TY11		TY10		TY9		TY8	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TY7		TY6		TY5		TY4		TY3		TY2		TY1		TY0	

DN_C_T_C1 (Temporal De-noiser Coefficient Temporal Chrominance #1) **0xF025212C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TY31		TY30		TY29		TY28		TY27		TY26		TY25		TY24	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TY23		TY22		TY21		TY20		TY19		TY18		TY17		TY16	

VY#[1:0]	Temporal filter coefficient in Chrominance
0~3	Filter coefficient in temporal de-noiser # : coefficient number (0~31)

DN_STATES0_TEM (Temporal de-noiser Count States #0) 0xF0252130

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BY	DN	IM	0		LCNT_Y										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0					PCNT_Y										

BY[31]	Busy operation (Read only)
0	IDLE
1	BUSY

DN[30]	Operation Done
0	-
1	Operation Done. (If write '1', this field is clear.)

IM[29]	Done Interrupt mask
0	Masked
1	Unmasked

LCNT_Y[26:16]	Line Counter in Luminance (Read only)
-	Line Counter in Luminance

PCNT_Y[10:0]	Pixel Counter in Luminance (Read only)
-	Pixel Counter in Luminance

DN_STATES1_TEM (Temporal de-noiser Count States #1) 0xF0252134

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0					LCNT_C										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0					PCNT_C										

LCNT_C[26:16]	Line Counter in Chrominance (Read only)
-	Line Counter in Chrominance

PCNT_C[10:0]	Pixel Counter in Chrominance (Read only)
-	Pixel Counter in Chrominance

DN_DIV_IMG_TEM (Temporal de-noiser Divide image) **0xF0252138**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DEN	DTOG	0			POS										

DEN[15]	Image divide enable
0	Normal
1	Divide image (filtered image – original image)

DTOG[14]	Divided image swap
0	Left image : filtered image Right image : original image
1	Left image : original image Right image : filtered image

POS[10:0]	Divide Position
-	Divide Position

If DEN=1, Image is divided original and filtered image in Figure 10.11.

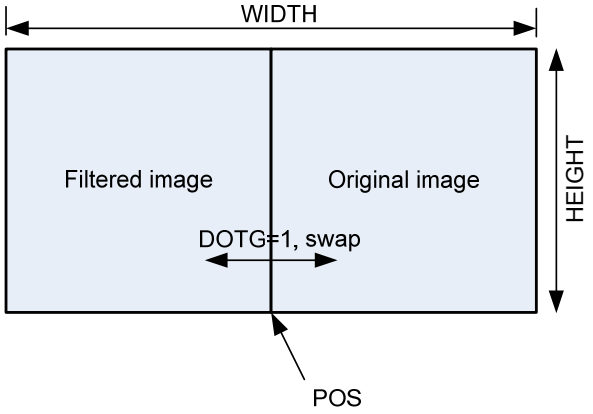


Figure 10.11 Image Divide to Original and Filtered Image

DN_C_SPA_Y0 (Spatial de-noiser Coefficient in Luminance #0) **0xF0252140**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOPE_Y								CA_Y							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA_Y								A_Y							

Field	Description
SLOPE_Y [31:24]	Parameter SLOPE in Luminance data (LPF) This value is set the converted sign of original value. The original value is always negative. (bit[7:6] is integer, the others is float) (default : **18** (-0.2857))
CA_Y [23:16]	Multiplied parameter A and C in Luminance data (LPF) (positive integer) (default : **50**)
BA_Y [15:8]	Multiplied parameter A and B in Luminance data (LPF) (positive integer) (default : **15**)
A_Y [7:0]	Parameter A in Luminance data (LPF) (positive integer) (default : **10**)

DN_C_SPA_Y1 (Spatial de-noiser Coefficient in Luminance #1) **0xF0252144**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F_Y								E_Y							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D_Y								Y_PLANE_Y							

Field	Description
F_Y [31:24]	Parameter F in Luminance data (HPF) (positive integer value) (default : **55**)
E_Y [23:16]	Parameter E in Luminance data (HPF) (positive integer value) (default : **60**)
D_Y [15:8]	Parameter D in Luminance data (HPF) (positive integer value) (default : **5**)
Y_PLANE_Y [7:0]	Parameter Y-plane in Luminance data (LPF) (positive integer) (default : **18**)

DN_C_SPA_Y2 (Spatial de-noiser Coefficient in Luminance #2) **0xF0252148**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L_Y								M2_Y							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1_Y								G_Y							

Field	Description
L_Y [31:24]	Parameter L in Luminance data (HPF) (positive integer value) (default : **74**)
M2_Y [23:16]	Parameter M2 in Luminance channel (HPF) (bit[7:6] is integer, the others is float) (positive integer value) (default : **6** (0.1))
M1_Y [15:8]	Parameter M2 in Luminance channel (HPF) (bit[7:6] is integer, the others is float) (positive integer value) (default : **64** (1.0))
G_Y [7:0]	Parameter G in Luminance data (HPF) (positive integer value) (default : **255**)

DN_C_SPA_Y3 (Spatial de-noiser Coefficient in Luminance #3) **0xF025214C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							Y_PLANE_H2_Y								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0							Y_PLANE_H1_Y								

Field	Description
Y_PLANE_H2_Y [24:16]	Parameter Y-plane in Luminance data (HPF) This value is set the converted sign of original value. (positive/negative integer) (default : **49** (-49))
Y_PLANE_H1_Y [8:0]	Parameter Y-plane in Luminance data (HPF) (refer to figure XX) This value is set the converted sign of original value. The original value is always positive. (positive integer) (default : **-5** (5))

DN_C_SPA_C0 (Spatial de-noiser Coefficient in Chrominance #0) **0xF0252150**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOPE_C								CA_C							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BA_C								A_C							

Field	Description
SLOPE_C [31:24]	Parameter SLOPE in Chrominance data (LPF) This value is set the converted sign of original value. The original value is always negative. (bit[7:6] is integer, the others is float) (default : **18** (-0.2857))
CA_C [23:16]	Multiplied parameter A and C in Chrominance data (LPF) (positive integer) (default : **25**)
BA_C [15:8]	Multiplied parameter A and B in Chrominance data (LPF) (positive integer) (default : **8**)
A_C [7:0]	Parameter A in Chrominance data (LPF) (positive integer) (default : **5**)

DN_C_SPA_C1 (Spatial de-noiser Coefficient in Chrominance #1) **0xF0252154**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F_C								E_C							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
D_C								Y_PLANE_C							

Field	Description
F_C [31:24]	Parameter F in Chrominance data (HPF) (positive integer value) (default : **55**)
E_C [23:16]	Parameter E in Chrominance data (HPF) (positive integer value) (default : **60**)
D_C [15:8]	Parameter D in Chrominance data (HPF) (positive integer value) (default : **5**)
Y_PLANE_C [7:0]	Parameter Y-plane in Chrominance data (LPF) (positive integer) (default : **7**)

DN_C_SPA_C2 (Spatial de-noiser Coefficient in Chrominance #2) **0xF0252158**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L_C								M2_C							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

M1_C	G_C

Field	Description
L_C [31:24]	Parameter L in Chrominance data (HPF) (positive integer value) (default : **74** (74))
M2_C [23:16]	Parameter M2 in Chrominance channel (HPF) (bit[7:6] is integer, the others is float) (positive integer value) (default : **6** (0.1))
M1_C [15:8]	Parameter M2 in Chrominance channel (HPF) (bit[7:6] is integer, the others is float) (positive integer value) (default : **64** (1.0))
G_C [7:0]	Parameter G in Chrominance data (HPF) (positive integer value) (default : **255**)

DN_C_SPA_C3 (Spatial de-noiser Coefficient in Chrominance #3) **0xF025215C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							Y_PLANE_H2_C								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0							Y_PLANE_H1_C								

Field	Description
Y_PLANE_H2_Y [24:16]	Parameter Y-plane in Chrominance data (HPF) This value is set the converted sign of original value. (positive/negative integer) (default : **49** (-49))
Y_PLANE_H1_Y [8:0]	Parameter Y-plane in Chrominance data (HPF) This value is set the converted sign of original value. The original value is always positive. (positive integer) (default : **-5** (5))

DN_FIFOSTATE (Denoise FIFO States) 0xF0252160

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				FIFOSTATE[23:12]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0				FIFOSTATE[11:00]											

FIFOSTATE [n]	Denoise FIFO States
[27]	FIFO overrun in denoise-3D
[26:24]	FIFO full states in denoise-3D (each FIFOs are Y,U,V)
[23]	FIFO overrun in denoise-2D (1st line buffer)
[22:20]	FIFO full states in denoise-2D (1st line buffers are Y,U,V)
[19]	FIFO overrun in denoise-2D (2nd line buffer)
[18:16]	FIFO full states in denoise-2D (2nd line buffers are Y,U,V)
[11]	FIFO underrun in denoise-3D
[10:8]	FIFO empty states in denoise-3D (rach FIFOs are Y,U,V)
[7]	FIFO underrun in denoise-2D (1st line buffer)
[6:4]	FIFO empty states in denoise-2D (1st line buffers are Y,U,V)
[3]	FIFO underrun in denoise-2D (2nd line buffer)
[2:0]	FIFO empty states in denoise-2d (2nd line buffers are Y,U,V)

DN_STATES0_SPA (Spatial de-noiser Count States #0) 0xF0252164

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BY	DN	IM	0		LCNT_Y										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0					PCNT_Y										

BY[31]	Busy operation (Read only)
0	IDLE
1	BUSY

DN[30]	Operation Done
0	-
1	Operation Done. (If write '1', this field is clear.)

IM[29]	Done Interrupt mask
0	Masked
1	Unmasked

LCNT_Y[26:16]	Line Counter in Luminance (Read only)
-	Line Counter in Luminance

PCNT_Y[10:0]	Pixel Counter in Luminance (Read only)
-	Pixel Counter in Luminance

DN_STATES1_SPA (Spatial De-noiser Count States #1) 0xF0252168

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0					LCNT_C										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0					PCNT_C										

LCNT_C[26:16]	Line Counter in Chrominance (Read only)
-	Line Counter in Chrominance

PCNT_C[10:0]	Pixel Counter in Chrominance (Read only)
-	Pixel Counter in Chrominance

DN_CTRL (Denoise Control) 0xF025216C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLUSH		0													
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0														BY_COEFF	

FLUSH[n]	FIFO Flush
[31]	FIFO flush in temporal de-noiser
[30]	FIFO flush in spatial de-noiser

BY_COEFF[n]	Bypass Coefficient
[1]	Bypass coefficient in temporal de-noiser All coefficient of temporal de-noiser is set to ‘0’
[0]	Bypass coefficient in spatial de-noiser All coefficient of spatial de-noiser is set to ‘0’

DN_DIV_IMG_SPA (Spatial de-noiser Divide image) 0xF0252170

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DEN	DTOG	0			POS										

DEN[15]	Image divide enable
0	Normal
1	Divide image (filtered image – original image)

DTOG[14]	Divided image swap
0	Left image : filtered image Right image : original image
1	Left image : original image Right image : filtered image

POS[10:0]	Divide Position
-	Divide Position

If DEN=1, Image is divided original and filtered image in Figure 10.11.

RD_IMG0_BASE0 (RDMA image #0 Base Address in Y Channel) 0xF0252180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG0_BASE0[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG0_BASE0[15:2]														0	

Field	Description
BASE0 [31:2]	RDMA image #0 base address in Y channel

RD_IMG0_BASE1 (RDMA image #0 Base Address in U Channel) 0xF0252184

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG0_BASE1[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG0_BASE1[15:2]														0	

Field	Description
BASE1 [31:2]	RDMA image #0 base address in U channel

RD_IMG0_BASE2 (RDMA image #0 Base Address in V Channel) 0xF0252188

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG0_BASE2[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG0_BASE2[15:2]														0	

Field	Description
BASE2 [31:2]	RDMA image #0 base address in V channel

RD_IMG0_OFS (RDMA image #0 Offset Address) 0xF025218C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG0_OFS1[15:0]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG0_OFS0[15:0]															

Field	Description
OFS1 [31:16]	RDMA image #0 offset address in C channel This offset means the line start point to the next line start point.
OFS0 [15:0]	RDMA image #0 offset address in Y channel This offset means the line start point to the next line start point.

RD_IMG1_BASE0 (RDMA image #1 Base Address in Y Channel) 0xF0252190

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG1_BASE0[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG1_BASE0[15:2]														0	

Field	Description
BASE0 [31:2]	RDMA image #1 base address in Y channel

RD_IMG1_BASE1 (RDMA image #1 Base Address in U Channel) 0xF0252194

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG1_BASE1[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG1_BASE1[15:2]														0	

Field	Description
BASE1 [31:2]	RDMA image #1 base address in U channel

RD_IMG1_BASE2 (RDMA image #1 Base Address in V Channel) 0xF0252198

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG1_BASE2[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG1_BASE2[15:2]														0	

Field	Description
BASE2 [31:2]	RDMA image #0 base address in V channel

RD_IMG1_OFS (RDMA image #1 Offset Address) 0xF025219C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG1_OFS1[15:0]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG1_OFS0[15:0]															

Field	Description
OFS1 [31:16]	RDMA image #1 offset address in C channel This offset means the line start point to the next line start point.
OFS0 [15:0]	RDMA image #1 offset address in Y channel This offset means the line start point to the next line start point.

RD_IMG2_BASE0_0 (RDMA decomp. data #0 Base Address in Y Channel) 0xF02521A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG2_BASE0[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG2_BASE0[15:2]														0	

Field	Description
BASE0 [31:2]	RDMA decomp.data #0 base address in Y channel

RD_IMG2_BASE1_0 (RDMA decomp. data #0 Base Address in U Channel) 0xF02521A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG2_BASE1[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG2_BASE1[15:2]														0	

Field	Description
BASE1 [31:2]	RDMA decomp. data #0 base address in U channel

RD_IMG2_BASE2_0 (RDMA decomp. data #0 Base Address in V Channel) 0xF02521A8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG2_BASE2[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG2_BASE2[15:2]														0	

Field	Description
BASE2 [31:2]	RDMA decomp. data #0 base address in V channel

RD_IMG2_BASE0_1 (RDMA decomp. data #1 Base Address in Y Channel) **0xF02521AC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG2_BASE0[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG2_BASE0[15:2]														0	

Field	Description
BASE0 [31:2]	RDMA decomp. data #1 base address in Y channel

RD_IMG2_BASE1_1 (RDMA decomp. data #1 Base Address in U Channel) **0xF02521B0**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG2_BASE1[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG2_BASE1[15:2]														0	

Field	Description
BASE1 [31:2]	RDMA decomp. data #1 base address in U channel

RD_IMG2_BASE2_1 (RDMA decomp. data #1 Base Address in V Channel) **0xF02521B4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_IMG2_BASE2[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_IMG2_BASE2[15:2]														0	

Field	Description
BASE2 [31:2]	RDMA decomp. data #1 base address in V channel

In RDMA2, two address registers are assigned to each channel in order to swap automatically the addresses to indicate two regions for compressed data by means of dual buffers. In general, the address of odd frame is assigned to RD_IMG2_BASE#_0 register, and the address of even frame is assigned to RD_IMG2_BASE#_1 register.

These registers are related to the RDMA and the COMP DMA such that the RD_IMG2_BASE#_0 register and CD_BASE#_1 register are the same and the RD_IMG2_BASE#_1 register and CD_BASE#_0 register are the same.

RD_CUR_ADDR0 (RDMA Current Address in Y Channel) 0xF02521B8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CUR_ADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUR_ADDR0[15:2]														0	

Field	Description (Read Only)
CUR_ADDR [31:2]	RDMA image #0 Current Address.

RD_CUR_ADDR1 (RDMA Current Address in U Channel) 0xF02521BC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CUR_ADDR1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUR_ADDR1[15:2]														0	

Field	Description (Read Only)
CUR_ADDR [31:2]	RDMA image #1 Current Address.

RD_CUR_ADDR2 (RDMA Current Address in V Channel) 0xF02521C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CUR_ADDR2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUR_ADDR2[15:2]														0	

Field	Description (Read Only)
CUR_ADDR [31:2]	RDMA decomp. data #1 Current Address.

RD_FIFOSTATE (RDMA FIFO States) **0xF02521C4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFOSTATE[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FIFOSTATE[15:0]															

FIFOSTATE [n]	RDMA FIFO States
[31]	FIFO underrun in the 4th FIFO (Decomp. DMA outFIFO)
[30:28]	FIFO full states in the 4th FIFO (Decomp. DMA outFIFO Y,U,V) **(Read Only)**
[27]	FIFO underrun in the 3rd FIFO (Decomp. DMA FIFO)
[26:24]	FIFO full states in the 3rd FIFO (Decomp. DMA FIFO Y,U,V) **(Read Only)**
[23]	FIFO underrun in the 2nd FIFO (RDMA #1 FIFO)
[22:20]	FIFO full states in the 2nd FIFO (RDMA #1 FIFO Y,U,V) **(Read Only)**
[19]	FIFO underrun in the 1st FIFO (RDMA #0 FIFO)
[18:16]	FIFO full states in the 1st FIFO (RDMA #0 FIFO) **(Read Only)**
[15]	FIFO overrun in the 4th FIFO (Decomp. DMA outFIFO)
[14:12]	FIFO empty states in the 4th FIFO (Decomp. DMA outFIFO Y,U,V) **(Read Only)**
[11]	FIFO overrun in the 3rd FIFO (Decomp. DMA FIFO)
[10:8]	FIFO empty states in the 3rd FIFO (Decomp. DMA FIFO Y,U,V) **(Read Only)**
[7]	FIFO overrun in the 2nd FIFO (RDMA #1 FIFO)
[6:4]	FIFO empty states in the 2nd FIFO (RDMA #1 FIFO Y,U,V) **(Read Only)**
[3]	FIFO overrun in the 1st FIFO (RDMA #0 FIFO)
[2:0]	FIFO empty states in the 1st FIFO (RDMA #0 FIFO Y,U,V) **(Read Only)**

RD_LINE_STATE0 (RDMA Count States #0) 0xF02521C8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BUSY			0		LCNT_Y										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0					PCNT_Y										

BUSY[31:29]	RDMA BUSY (Read Only)
BUSY[31]	RDMA decomp. DMA busy
BUSY[30]	RDMA image #1 busy
BUSY[29]	RDMA image #0 busy

LCNT_Y[26:16]	Line Count in Y channel (Read Only)
-	Current line counter that is processing.

PCNT_Y[10:0]	Pixel Count in Y channel (Read Only)
-	Current pixel counter that is processing.

RD_LINE_STATE1 (RDMA Count States #1) 0xF02521CC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RUD	DN	MASK		0	LCNT_C										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0					PCNT_C										

RUD[31]	Register Update Done
0	-
1	Register update

DN[30]	Frame Done
0	-
1	Frame Processing Done

MASK[29:28]	Interrupt Mask
MASK[29]	Register update interrupt mask 0 : mask 1 : unmask
MASK[28]	Frame Done interrupt mask 0 : mask 1 : unmask

LCNT_C[26:16]	Line Count in C channel (Read Only)
-	Current line counter that is processing.

PCNT_C[10:0]	Pixel Count in C channel (Read Only)
-	Current pixel counter that is processing.

RD_CTRL (RDMA Control Register) 0xF02521D0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															DC
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCM				0		DNS	CONT	BYD	TOG	0	OPT				

DC[16]	Decomp. buffer clear
0	-
1	Decomp. buffer clear

DCM[11:8]	Decomp. coefficient Mode
0	Normal Mode (No effect) {-7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7}
1	{-6, -6, -4, -4, -2, -2, 0, 0, 0, 2, 2, 4, 4, 6, 6}
2	{-6, -6, -4, -4, -2, -2, -1, -0, 1, 2, 2, 4, 4, 6, 6}
3	{-6, -6, -3, -3, -3, 0, 0, 0, 0, 0, 3, 3, 3, 6, 6}
4	{-6, -6, -3, -3, -3, -1, -1, 0, 1, 1, 3, 3, 3, 6, 6}
5	{-6, -6, -3, -3, -3, -2, -1, 0, 1, 2, 3, 3, 3, 6, 6}
6	{-6, -6, -4, -4, -3, -2, -1, 0, 1, 2, 3, 4, 4, 6, 6}
7	{-6, -6,- 6, -3, -3, -2, -1, 0, 1, 2, 3, 3, 6, 6, 6}

DNS[9]	Decompression operation Not Skip in first frame
0	Skip in first frame
1	Not skip in first frame

CONT[8]	RDMA Continuous Mode
0	Frame-by-frame mode
1	Continuous mode

BYD[7]	Bypass Decompression
0	Decoding bitstream. Input bitstream is the compression data.
1	Not decoding. Input bitstream is the non-compression data.

TOG[6]	RDMA Decomp. base address auto toggle
0	Not toggle base address. RDMA is used only RDMA_IMG2_BASE0_0, BASE0_1, BASE0_2.
1	Auto toggle base address as changing frame If Odd frame, base address is RDMA_IMG2_BASE0_0, BASE1_0, BASE2_0 If Even frame, base address is RDMA_IMG2_BASE0_1, BASE1_1, BASE2_1

OPT[4]	RDMA transfer Optimization
0	Normal. Burst 8 transfer.
1	Burst 16 transfer.

RD_COMP_PL0 (RDMA comp. data size in Y channel) 0xF02521D0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_COMP_PL0[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_COMP_PL0[15:0]															

Field	Description
PL0 [31:0]	RDMA compression data size in Y channel. In normally, this value is equal to image hsize * vsize

RD_COMP_PL1 (RDMA comp. data size in C channel) 0xF02521D4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RD_COMP_PL1[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_COMP_PL1[15:0]															

Field	Description
PL1 [31:0]	RDMA compression data size in C channel. In normally, this value is equal to image hsize * vsize / 4 in 420 Mode. If 422mode, this value is image hsize*vsize/2.

CD_BASE0_0 (Comp. DMA #0 base address in Y channel) 0xF0252200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CD_BASE0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CD_BASE0[15:2]														0	

Field	Description
BASE0 [31:2]	Compression DMA #0 base address in Y channel

CD_BASE1_0 (Comp. DMA #0 base address in U channel) 0xF0252204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CD_BASE1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CD_BASE1[15:2]														0	

Field	Description
BASE1 [31:2]	Compression DMA #0 base address in U channel

CD_BASE2_0 (Comp. DMA #0 base address in V channel) 0xF0252208

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CD_BASE2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CD_BASE2[15:2]														0	

Field	Description
BASE2 [31:2]	Comp. DMA #0 base address in V channel

CD_BASE0_1 (Comp. DMA #1 base address in Y channel) 0xF025220C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CD_BASE0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CD_BASE0[15:2]														0	

Field	Description
BASE0 [31:2]	Comp. DMA #0 base address in Y channel

CD_BASE1_1 (Comp. DMA #1 base address in U channel) 0xF0252210

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CD_BASE1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CD_BASE1[15:2]														0	

Field	Description
BASE1 [31:2]	Comp. DMA #0 base address in U channel

CD_BASE2_1 (Comp. DMA #1 base address in V channel) 0xF0252214

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CD_BASE2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CD_BASE2[15:2]														0	

Field	Description
BASE2 [31:2]	Comp. DMA #0 base address in V channel

CD_CUR_ADDR (Comp. DMA current address) 0xF0252218

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CD_CUR_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CD_CUR_ADDR[15:2]														0	

Field	Description
CUR_ADDR [31:2]	Comp. DMA current address

CD_STATE (Comp. DMA State) 0xF025221C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BY	0											FIFOSTATE[7:4]			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DN	IM	0										FIFOSTATE[3:0]			

BY[31]	Comp. DMA BUSY (Read only)
0	Idle.
1	Busy

DN[15]	Comp. DMA Frame Done
0	-
1	Frame done. If this field writes ‘1’, this field is clear.

IM[14]	Comp. DMA Frame Done interrupt mask
0	Mask.
1	Unmask.

FIFOSTATE [n]	Comp. DMA FIFO States
[7]	FIFO overrun in the FIFO.
[6:4]	FIFO full states in the FIFO (HuffDMA FIFO Y,U,V) **(Read Only)**
[3]	FIFO overrun in the the FIFO.
[2:0]	FIFO empty states in the FIFO (HuffDMA FIFO Y,U,V) **(Read Only)**

CD_CTRL (Comp. DMA Control register) **0xF0252220**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLR	0			CM				0	BY	TOG	OPT	0			

CLR[15]	Comp. DMA FIFO clear
0	-
1	FIFO clear

CM[11:8]	Compression coefficient Mode
0	Normal mode {-7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7}
1	{-6, -6, -4, -4, -2, -2, 0, 0, 0, 2, 2, 4, 4, 6, 6}
2	{-6, -6, -4, -4, -2, -2, -1, -0, 1, 2, 2, 4, 4, 6, 6}
3	{-6, -6, -3, -3, -3, 0, 0, 0, 0, 0, 3, 3, 3, 6, 6}
4	{-6, -6, -3, -3, -3, -1, -1, 0, 1, 1, 3, 3, 3, 6, 6}
5	{-6, -6, -3, -3, -3, -2, -1, 0, 1, 2, 3, 3, 3, 6, 6}
6	{-6, -6, -4, -4, -3, -2, -1, 0, 1, 2, 3, 4, 4, 6, 6}
7	{-6, -6,- 6, -3, -3, -2, -1, 0, 1, 2, 3, 3, 6, 6, 6}

BY[6]	Comp. DMA encode bypass
0	Encoding data Output bitstream is the compression data.
1	Not encoding Output bitstream is non-compression data.

TOG[5]	Comp. DMA base address auto toggle
0	Not toggle base address. Comp. DMA is used only CD_ BASE0_0, BASE1_0, BASE2_0.
1	Auto toggle base address as changing frame If Odd frame, base address is CD _BASE0_0, BASE1_0, BASE2_0 If Even frame, base address is CD_BASE0_1, BASE1_1, BASE2_1

OPT[4]	Comp. DMA transfer Optimization
0	Normal. Burst 8 transfer.
1	Burst 16 transfer.

CD_COMP_CNT0 (Comp. DMA comp. data count in Y channel) 0xF0252230

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP_CNT0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP_CNT0[15:2]														0	

Field	Description (Read only)
COMP_CNT0 [31:2]	Comp. DMA encoded data count in Y channel In normally, this field value is equal to image hsize * vsize

CD_COMP_CNT1 (Comp. DMA comp. data count in U channel) 0xF0252234

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP_CNT131:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP_CNT1[15:2]														0	

Field	Description (Read only)
COMP_CNT1 [31:2]	Comp. DMA encoded data count in U channel In normally, this field value is equal to image hsize * vsize /4 in 420 mode.

CD_HUFF_CNT2 (Comp. DMA comp. data count in V channel) 0xF0252238

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP_CNT2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP_CNT2[15:2]														0	

Field	Description (Read only)
COMP_CNT2 [31:2]	Comp. DMA encoded data count in U channel In normally, this field value is equal to image hsize * vsize /4 in 420 mode.

OD_BASE0 (ODMA base address in Y channel) 0xF0252280

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OD_BASE0[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OD_BASE0[15:2]														0	

Field	Description
BASE0 [31:2]	ODMA base address in Y channel

OD_BASE1 (ODMA base address in U channel) 0xF0252284

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OD_BASE1[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OD_BASE1[15:2]														0	

Field	Description
BASE1 [31:2]	ODMA base address in U channel

OD_BASE2 (ODMA base address in V channel) 0xF0252288

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OD_BASE2[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OD_BASE2[15:2]														0	

Field	Description
BASE2 [31:2]	ODMA base address in V channel

OD_SIZE (ODMA image size) **0xF025228C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				HEIGHT											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				WIDTH											

Field	Description
HEIGHT [27:16]	ODMA image height size
WIDTH [11:0]	ODMA image width size

OD_OFS (ODMA address offset) **0xF0252290**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				OFS1											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				OFS0											

Field	Description
OFS1 [27:16]	ODMA address offset in C channel
OFS0 [11:0]	ODMA address offset in Y channel

OD_CFG (ODMA Config) 0xF0252294

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				CP	WAITCNT			0	RDY	-	DIR	TYPE			

CP[11]	Chrominance writing mode in 420sep mode
0	Y0->U0->V0->Y1->Y2->U1->…
1	Y0->U0->Y1->V0->Y2->U1->…

WAITCT[10:8]	Wait cycle
-	Wait cycle count for RDY being ‘1’

RDY[6]	Access wait control register
-	Access wait control register. Valid for DIR being ‘1’ 0 : wait for “WAITCNT+1” cycle for direct path. 1 : wait until output FIFO is not empty

DIR[4]	DIRECT PATH
0	Normal PATH (to memory bus)
1	DIRECT PATH (to scaler or LCD path)

TYPE[1:0]	Image Type
0	4:2:0 separate mode
1	4:2:2 separate mode
2	4:2:2 sequence mode 0
3	4:2:2 sequence mode 1
10	4:2:2 interleave
11	4:2:0 interleave

OD_CTRL (ODMA control register) 0xF02522A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0							RST	0				IDONE	IBUSY	IRDY	0

RST[8]	States machine reset
0	-
1	Reset

IDONE[3]	Done Interrupt mask
0	Mask
1	Unmask

IBUSY[2]	BUSY Interrupt mask
0	Mask
1	Unmask

IRDY[1]	RDY Interrupt mask
0	Mask
1	Unmask

OD_STATE (ODMA State) 0xF02522A8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										IBY	IRY	IDN	FD	BA	RA

IBY[5]	Busy interrupt flag
0	-
1	Generate busy interrupt

IRY[4]	Ready Interrupt flag
0	-
1	Generate ready interrupt

IDN[3]	Frame done Interrupt flag
0	-
1	Generate ready interrupt

FD[2]	Frame done states
0	-
1	Frame done

BA[1]	Busy all states
0	-
1	Busy all

RA[0]	Ready all states
0	-
1	Ready all

GM_CTRL (Gamut Mapper Control Register) 0xF0252400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BYPASS															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														RND	INIT

BYPASS [31]	Bypass Enable
0	Normal Process
1	This enable bit can do bypassing between input ports and output ports in Gamut Mapper hardware. This bit can be used as another Gamut Mapper Enable bit.

RND [1]	Round
0	Normal Process
1	All the internal arithmetic logic use the round bit as ‘1’

INIT [0]	Initialization Start
0	No meaning
1	This initializes all LUTs in Gamut Mapper hardware. These procedures spend some processing time. The current status of initialization procedures can be checked by the GM_STATUS register.

GM_STATUS (Gamut Mapper Status Register) 0xF0252404

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														BUSYINIT	DNINIT

BUSYINIT [1]	Busy state in Initialization
0	Not being initialized
1	This indicates the initialization procedure is not done yet.

DNINIT [0]	Done of initialization procedure
0	No meaning
1	Done of initialization procedure

HI_CTRL (Histogram Generator Control Register) 0xF0252600

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLR	BCTRDIS											MFRM			AUTO
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRFINI						LUTDIRINI	LUTDIRUP	LUTUSE	LUTINIT	LUTUUP	CDFEN	INTP		FINI	START

CLR [31]	Clear Internal Variables
0	No meaning
1	Clear All Internal Variables in Histogram Monitor

BCTRDIS [0]	Busy Control Disable
0	Normal processing
1	If this is set, the hardware ignores the BUSY status in Histogram Generator hardware. This bit may be not useful for normal processing.

MFRM [19:17]	Multiple Frame Duration
-	This value means the number of frames to be processed at once. If the manual mode is on, after the user's triggering which is START bit, the hardware will operate during the number of frames defined by this value. Otherwise, the hardware will operate repeatedly the number of frames as a unit. If this is set as multiple frames, the hardware can make the average value from pixel data of corresponding frames. This register field is mapped to the followings. 000 : 1 frame 001 : 2 frames 010 : 4 frames 011 : 8 frames 100 : 16 frames 101 : 32 frames

AUTO [16]	Automatic Mode
0	Manual Mode. The measurement or LUT update can be controlled by the user.
1	If this bit is set, the hardware automatically processes the corresponding frame as the value of MFRM.

DIRFINI [15]	Direct Finish Signal
0	No meaning
1	This bit controls directly the hardware to be finished processing the current frame.

LUTDIRINI [9]	Direct LUT initialization
0	No meaning
1	This bit makes the hardware to start LUT initialization directly.

LUTDIRUP [8]	Direct LUT updates
0	No meaning
1	This bit makes the hardware to update LUT directly.

LUTUSE [7]	LUT Enable
0	Do not use LUT
1	Use LUT in Histogram Generator. It affects only the luminance signals.

LUTINIT [6]	LUT Initialization
0	No meaning
1	This bit makes the hardware to start LUT initialization procedure after finishing the current frame processing.

LUTUUP [5]	LUT User Update
0	No meaning
1	This bit makes the hardware to update LUT after finishing the current frame processing.

CDFEN [4]	CDF Enable
0	CDF disable
1	This bit enables the feature of calculating fully histogram equalization LUT table. If LUTUSE bit is set, the luminance signals are mapped into the new values in generated LUT table. Otherwise, the result of LUT table can be monitored by the user.

INTP [3]	Interpolation Start
0	No meaning
1	This bit makes the hardware to start the interpolation for the missing pixels between the neighbor segments. This procedure starts after the current frame processing.

FINI [1]	Measure Finish Signal
0	No meaning
1	This bit controls the hardware to be finished processing the current frame and measuring the information about histogram in current frame.

START [0]	Measure Start Signal
0	No meaning
1	This bit controls the hardware to start processing the current frame and measuring the information about histogram in current frame.

HI_STATUS (Histogram Monitor Status Register) 0xF0252604

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												INTPCSTAT			MCSTAT
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
											LUTUUPB			INTPRDY	FBUSY

INTPCSTAT [19:17]	Interpolation Controller State
-	This indicates the internal hardware state in interpolation logic. This is used for hardware debugging.

MCSTAT [16]	Measure Controller State
0	Not processing
1	This indicates that the measurement hardware logic is busy.

LUTUUPB [4]	LUT User Update BUSY
0	Not processing
1	This indicates that the LUT is being updated as the table written by the user.

INTPRDY [1]	Interpolation BUSY
0	Not processing
1	This indicates that the interpolation for missing pixels is processing.

FBUSY [1]	Frame BUSY
0	Not processing
1	This indicates that the current frame requires the Histogram Generator block to manipulate the output pixels, not just to monitor or get the information of pixels.

All fields of this register are read-only.

HI_CONFIG (Histogram Generator Configuration Register) 0xF0252608

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SIZE														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RND			VOFF					HOFF							

SIZE [30:16]	Sample Size
-	

RND [15]	Round Control Bit
0	Disable rounding
1	Enable rounding in Interpolation operation

VOFF [12:8]	Vertical Offset of Sample Pixels
-	This means the vertical offset by which the pixels are sampled vertically. This value is larger, the sample size is smaller.

HOFF [12:8]	Horizontal Offset of Sample Pixels
-	This means the horizontal offset by which the pixels are sampled horizontally. This value is larger, the sample size is smaller.

HI_SEGS0 (Histogram Generator Segments 0 Register) **0xF0252620**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEG03								SEG02							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEG01								SEG00							

HI_SEGS1 (Histogram Generator Segments 1 Register) **0xF0252624**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEG07								SEG06							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEG05								SEG04							

HI_SEGS2 (Histogram Generator Segments 2 Register) **0xF0252628**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEG11								SEG10							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEG09								SEG08							

HI_SEGS3 (Histogram Generator Segments 3 Register) **0xF025262C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								SEG14							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEG13								SEG12							

SEGx	Segment Positions
-	SEG**x** (**x** = 0, 1, 2, … 14) is the starting position of segment **x**. Total 16 segments compose 8bit pixel range(0 ~ 255). The SEG**x** is increasingly as **x** increases. The starting position of the first segment is fixed to 0, and the ending position of the last segment(SEG14, or 15th segment) is fixed to 255.

To program these registers should be done before starting the hardware, otherwise any output relative to the Histogram Generator is unreliable.

HI_CDFS0 (Histogram Generator CDF output 0 Register) **0xF0252630**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CDF03								CDF02							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDF01								CDF00							

HI_CDFS1 (Histogram Generator CDF output 1 Register) **0xF0252634**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CDF07								CDF06							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDF05								CDF04							

HI_CDFS2 (Histogram Generator CDF output 2 Register) **0xF0252638**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CDF11								CDF10							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDF09								CDF08							

HI_CDFS3 (Histogram Generator CDF output 3 Register) **0xF025263C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CDF15								CDF14							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDF13								CDF12							

CDFx	CDF outputs
-	CDF**x** (**x** = 0, 1, 2, … 15) is the normalized output for segment **x**. The CDF value can be described below. $$CDF_x = \frac{\sum_{k=0}^{x} PDF_x}{TotalSampleSize} \times 256$$ PDF**x** is the number of pixels which are in the corresponding segment(segment **x**). The CDF**x** is increasingly as **x** increases.

Above all registers are read-only.

HI_CNTS0 (Histogram Generator CNT 0 Register) **0xF0252640**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT01															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT00															

HI_CNTS1 (Histogram Generator CNT 1 Register) **0xF0252644**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT03															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT02															

HI_CNTS2 (Histogram Generator CNT 2 Register) **0xF0252648**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT05															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT04															

HI_CNTS3 (Histogram Generator CNT 3 Register) **0xF025264C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT07															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT06															

…

HI_CNTS7 (Histogram Generator CNT 7 Register) **0xF025265C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT15															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT14															

CNTx	CNT output
-	CNT**x** (**x** = 0, 1, 2, … 15) is the number of pixels which are in the range of segment **x** in the current frame. These are raw values of hardware counters. These values are used to generate the CDF outputs, which are normalized to 8 bit pixel range.

Above all registers are read-only.

HI_SCALE0 (Histogram Generator ALPHA 0 Register) 0xF0252660

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA03								ALPHA02							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALPHA01								ALPHA00							

HI_SCALE1 (Histogram Generator ALPHA 1 Register) 0xF0252664

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA07								ALPHA06							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALPHA05								ALPHA04							

HI_SCALE2 (Histogram Generator ALPHA 2 Register) 0xF0252668

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA11								ALPHA10							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALPHA09								ALPHA08							

HI_SCALE3 (Histogram Generator ALPHA 3 Register) 0xF025266C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA15								ALPHA14							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALPHA13								ALPHA12							

ALPHAx	ALPHA parameter
-	ALPHA**x** (**x** = 0, 1, 2, … 15) is the strength parameter for segment **x**. The ALPHA value is used to control the strength of each segment in the histogram equalization. The ALPHA value is larger, the effect of equalization increases. The maximum value of 255 means that the equalization result be transferred as it is.

HI_LUTS00 (Histogram Generator LUT 0 Register) 0xF0252700

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LUT003								LUT002							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUT001								LUT000							

HI_LUTS01 (Histogram Generator LUT 1 Register) 0xF0252704

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LUT007								LUT006							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUT005								LUT004							

HI_LUTS63 (Histogram Generator LUT 63 Register) 0xF02527FC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LUT255								LUT254							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUT253								LUT252							

LUTSx	LUT value
-	LUT**x** (**x** = 0, 1, 2, … 255) is the mapping value for pixel **x**. If LUTUSE field in HI_CTRL register is set, the final output of Histogram Generator hardware is the output of LUT table. These registers can be read or written. Writing this table can be possible at any time, and can be updated and effected into actual outputs when the LUTUUP field or the LUTDIRUP in HI_CTRL register is set. But, Reading this table can be possible when the LUTUSE field is off or the FBUSY field in HI_STATUS register is not active.

11 LVDS

11.1 Overview

The LVDS transmitter converts 35s of data into 5LVDS data streams. A phase-locked clock is transmitted in parallel with the data streams. At a clock frequency of 80MHz, 35ts of RGB data and 7bits of timing and control data are transmitted at a rate of 560s per LVDS data channel. Using a 80MHz clock, the throughput is 350Mbytes/sec. The transmitter provides rising/falling edge clocks for convenient interface with a variety of graphics and LCD panel controllers.

11.1.1 LVDS Specific Features

- Output clock range : 25 to 80MHz
- 35:7 data channel compression up to 560 Mbps on each LVDS channel
- Power down mode
- Up to 350 Mbytes/sec bandwidth
- Falling clock edge data strobe
- DLL requires no external component
- 6 LVDS output channels (5 data channels, 1clock channel)

11.2 Architecture

TCC8900 LVDS has following interface.
- Dedicated Video input interface
- Converted from LCD interface to LVDS interface
- DDI_Config for set the control register of each block

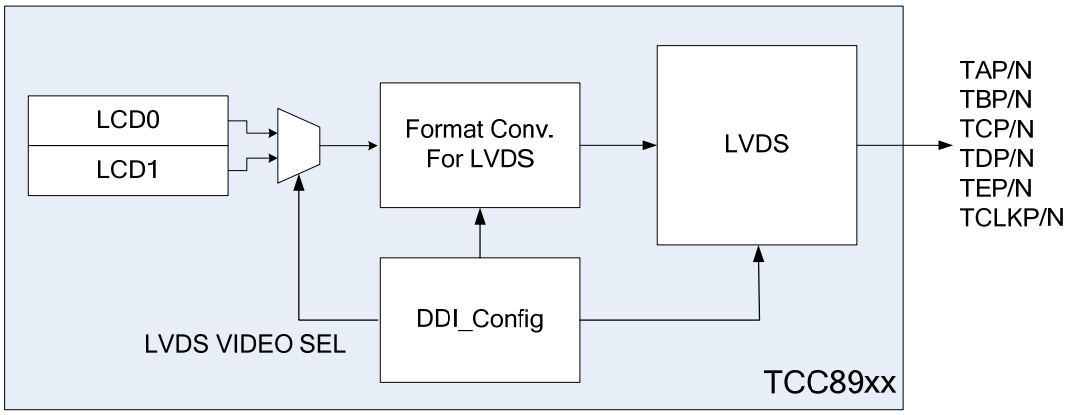


Figure 11.1 Connection of LVDS in TCC8900

11.2.1 LVDS

11.2.2 Block Diagram of LVDS

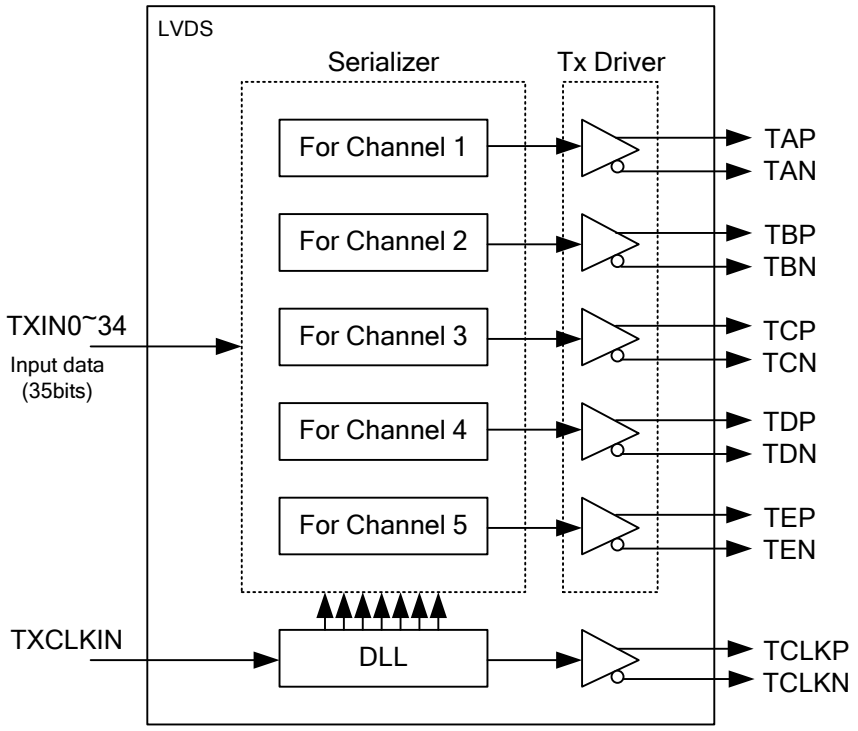


Figure 11.2 Block Diagram of LVDS

11.2.3 Timing Diagram of LVDS

The timing diagram of LVDS is shown below. The input data channels that support 35:7 data channel compression are connected to PXCLK, LHS, LVS, LDE of LCD Controller. The connection between LCD controller and LVDS can be configurable in LVDS_TXO_SEL0~8 of DDI_CONFIG register

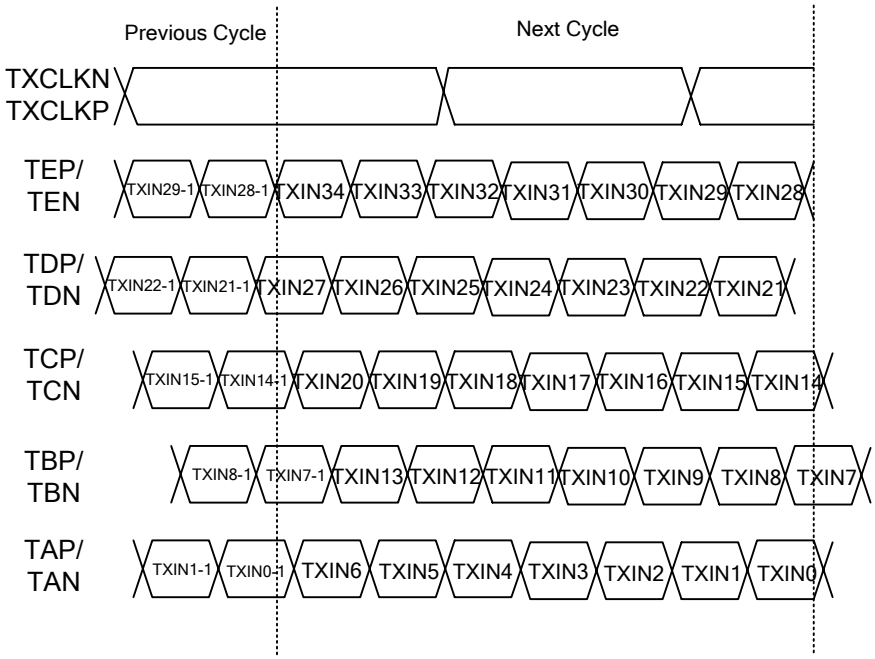


Figure 11.3 Timing Diagram of LVDS

12 DDI_CONFIG

12.1 Overview

The DDI_CONFIG controls over DDI bus device configurations, like LCD port muxing, on-the-fly mode port connection, LVDS port mapping, HDMI AES KEY setting, and so on

12.1.1 DDI_CONFIG Specific Features

- connection method setting between LVDS 35:7 data channel and LCD Controller ports
- Port connection of Scaler, LCDC, and VIQE for on-the-fly-mode
- Control the Connection path of each display devices (LCDC, LVDS, HDMI, TV-Encoder)
- LVDS on-off control
- HDMI on-off control
- Set HDMI AESKEY for encryption

12.2 Block Diagram of DDI_CONFIG

Below figure describes simple block diagram of DDI_CONFIG block. The port mux for on-the-fly mode lies between LCD Controllers and VIQE/memory to memory scaler0/1, to eliminate the duplicate memory access, which is inevitable if they are not directly connected. For example, “VIQE-MSCL-LCDC” combination enables one-stop processing of image enhancement, scaling and ouput to external LCD and therefore it can reduce memory access time.

The LCD Controller supports various output interfaces such as LVDS, HDMI, TV-OUT, LCDC. The LCDC interface can be provided upto 2 channels.

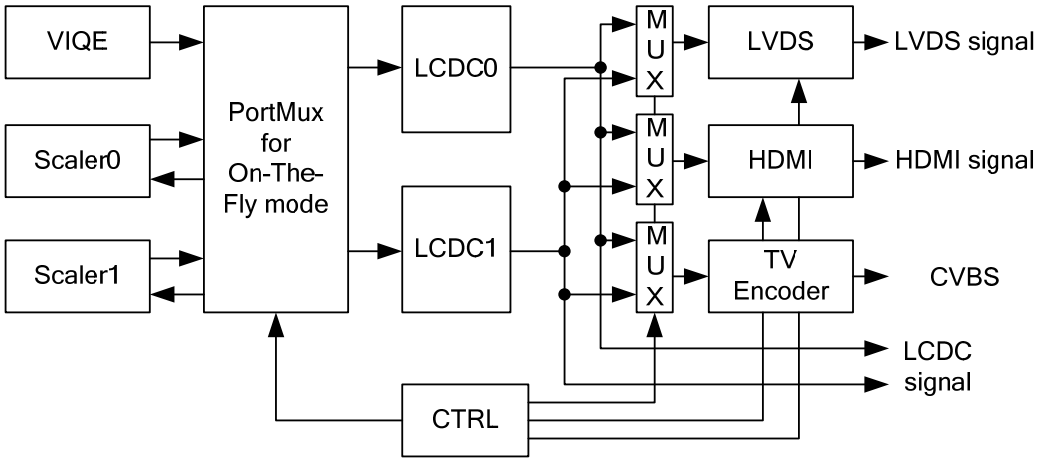


Figure 12.1 Block Diagram of DDI_CONFIG in TCC8900

12.3 Register Description

Table 12.1 DDI_CONFIG Register Map (Base Address = 0xF0251000)

Name	Address	Type	Reset	Description
NTSCPAL_SEL	0x000	R/W	0x00000001	NTSCPAL_Encoder select
LVDS_CTRL	0x004	R/W	0x04444443	LVDS Control register
LVDS_TXO_SEL0	0x008	R/W	0x03020100	LVDS TXOUT select #0
LVDS_TXO_SEL1	0x00C	R/W	0x09080504	LVDS TXOUT select #1
LVDS_TXO_SEL2	0x010	R/W	0x0D0C0B0A	LVDS TXOUT select #2
LVDS_TXO_SEL3	0x014	R/W	0x13121110	LVDS TXOUT select #3
LVDS_TXO_SEL4	0x018	R/W	0x1A191514	LVDS TXOUT select #4
LVDS_TXO_SEL5	0x01C	R/W	0x0E070618	LVDS TXOUT select #5
LVDS_TXO_SEL6	0x020	R/W	0x1B17160F	LVDS TXOUT select #6
LVDS_TXO_SEL7	0x024	R/W	0x1F1E1F1E	LVDS TXOUT select #7
LVDS_TXO_SEL8	0x028	R/W	0x001E1F1E	LVDS TXOUT select #8
HDMI_CTRL	0x02C	R/W	0x00008002	HDMI Control register
PWDN	0x030	R/W	0x00000000	Power Down
SWRESET	0x034	R/W	0x00000000	Soft Reset
ON_THE_FLY	0x038	R/W	0x00000000	On-The-Fly mode
HDMI_AES	0x044	R/W	0x00000000	HDMI AES
HDMI_AES_DATA0	0x048	RW	0x00000000	HDMI AES DATA #0

Name	Address	Type	Reset	Description
HDMI_AES_DATA1	0x04C	R/W	0x00000000	HDMI AES DATA #1
HDMI_AES_HW0	0x050	R/W	0x00000000	HDMI AES HW #0
HDMI_AES_HW1	0x054	R/W	0x00000000	HDMI AES HW #1
HDMI_AES_HW2	0x058	R/W	0x00000000	HDMI AES HW #2

NTSCPAL_SEL (NTSCPAL Select) 0xF0251000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															SEL

SEL [0]	NTSCPAL Select
0	Connect port from LCDC0 to TV-Encoder.
1	Connect port from LCDC1 to TV-Encoder

LVDS_CTRL (LVDS Control register) 0xF0251004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0					ECON			0	DCON			0	CCON		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	BCON			0	ACON			0	CLK_CON			0	EN	RST	SEL

SEL [0]	LVDS Select
0	Connect port from LCDC0 to LVDS.
1	Connect port from LCDC1 to LVDS

RST [1]	LVDS Reset
0	Normal
1	Reset

EN [2]	LVDS Enable
0	Disable
1	Enable

CLKCON [6:4]	Clock Control
-	TCLKP/TCLKN skew control. (default : 4)

ACON [10:8]	Data ch. 0 Control
-	TAP/TAN skew control (default : 4)

BCON [14:12]	Data ch.1 Control
-	TBP/TBN skew control. (default : 4)

CCON [18:16]	Data ch. 2 Control
-	TCP/TCN skew control (default : 4)

DCON [22:20]	Data ch.3 Control
-	TDP/TDN skew control. (default : 4)

ECON [26:24]	Data ch. 4 Control
-	TEP/TEN skew control (default : 4)

LVDS_TXO_SELx (LVDS TX_Data x) **0xF0251008 + (x * 0x04)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL_3+(4*x)								SEL_2+(4*x)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL_1+(4*x)								SEL_0+(4*x)							

Field	Description
SEL_0 ~ SEL_34	Connection bit of LPXDATA and TXIN of LVDS. The mean of 0~23 is the bit of TXIN. For example, the SEL_0 is 0, connection path is LPXDATA[0] and TXIN0. The other example, SEL_0 is 1, connection path is LPXDATA[0] and TXIN1.

SEL_x	Bit of LPXDATA
0	LPXDATA [0]
1	LPXDATA[1]
:	:
22	LPXDATA[22]
23	LPXDATA[23]
24	LDE
25	LHS
26	LVS
27~29, 32~34	0
30~31	1
TXCLK	LPXCLK

HDMI_CTRL (HDMI Control register) 0xF025102C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL	EN	0										RESET			

SEL [15]	Select for connection path of LCDC
0	Connection path is LCDC0 and HDMI
1	Connection path is LCDC1 and HDMI

EN[14]	HDMI Enable
0	Disable
1	Enable

RESET[3:0]	Reset for HDMI
Bit 0	HDMI Reset
Bit 1	SPDIF Reset
Bit 2	TMDS Reset
Bit 3	NOT USED

PWDN (Power down for DDIBUS) **0xF0251030**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0							PWDN								

PWDN[8:0]	Power down for DDIBUS
0	Normal
1	Power Down

BIT	Module
0	Camera Interface (CIF)
1	Video Image Quality Enhancer (VIQE)
2	LCD #0 Interface (LCDC0)
3	LCD #1 Interface (LCDC1)
4	LCDSI Interface (LCDSI)
5	Memory Scaler #0 (MSCL0)
6	Memory Scaler #1 (MSCL1)
7	DDIBUS Cache (DDIC)
8	HDMI interface (HDMI)

SWRESET (SWReset for DDIBUS) 0xF0251034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0							SWRESET								

SWRESET[8:0]	SW Reset for DDIBUS
0	Normal
1	SW Reset

BIT	Module
0	Camera Interface (CIF)
1	Video Image Quality Enhancer (VIQE)
2	LCD #0 Interface (LCDC0)
3	LCD #1 Interface (LCDC1)
4	LCDSI Interface (LCDSI)
5	Memory Scaler #0 (MSCL0)
6	Memory Scaler #1 (MSCL1)
7	DDIBUS Cache (DDIC)
8	HDMI interface (HDMI)

ON_THE_FLY (On The Fly mode) 0xF0251038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										SC1		SC0		VIQE	

VIQE[1:0]	Port Connection for the OnTheFly mode in VIQE
0	Connection path is from VIQE to LCDC0 **** In this case, the SC0 and SC1 should not be "0"***
1	Connection path is from VIQE to LCDC1 **** In this case, the SC0 and SC1 should not be "1"***.
2	Connection path is from VIQE to MSCL0 **** In this case, the SC1 should not be "2***"
3	Connection path Is from VIQE to MSCL1 **** In this case, the SC0 should not be "3"***

SC0[3:2]	Port Connection for the OnTheFly mode in MSCL0
0	Connection path is from MSCL0 to LCDC0 **** In this case, the VIQE and SC1 should not be "0"***
1	Connection path is from MSCL0 to LCDC1 **** In this case, the VIQE and SC1 should not be "1"***
2	NOT USED
3	Connection path Is from MSCL0 to MSCL1 **** In this case, the VIQE should not be "3"***

SC1[5:4]	Port Connection for the OnTheFly mode in MSCL1
0	Connection path is from MSCL1 to LCDC0
1	Connection path is from MSCL1 to LCDC1
2	Connection path Is from MSCL1 to MSCL0 **** In this case, the VIQE should not be "2"***
3	NOT USED

To use the on-the-fly mode, this registers should be set appropriately to get required path connections and formerly each module must be enabled.

On-the-fly connections are applicable from the combination of above register.

The related fields are "SRC" fields in the LI0C, LI1C, LI2C of the LCD controllers, "INPATH" of the MSCCTR register and "PATH" of the DSTCFG register of the memory-to-memory scaler, and "DIR" field of the OD_CFG of the Video Image Enhancement Block. Only the one of the "SRC" fields of the LI0C, LI1C, LI2C should be "1" to connect On-the-Fly path.

HDMI_AES (HDMI AESKEY Valid) 0xF0251044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															VLD

VLD[0]	HDMI AES KEY valid
0	Normal
1	Valid

HDMI_AES_DATA0 (HDMI AESKEY DATA0) 0xF0251048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AESKEY_DATA0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AESKEY_DATA0															

Field	Description
AESKEY DATA_0	AESKEY DATA[31:0]

HDMI_AES_DATA1 (HDMI AESKEY DATA1) 0xF025104C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															DATA1

Field	Description
AESKEY DATA_1	AESKEY DATA[32]

HDMI_AES_HW_x (HDMI AESKEY HW_x) 0xF0251050 + (x * 0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AESKEY_HW_x															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AESKEY_HW_x															

Field	Description
AESKEY HW_0	AESKEY DATA[64:33]
AESKEY HW 1	AESKEY DATA[96:65]
AESKEY HW_2[30:0]	AESKEY DATA[127:97]

13 DDI_Cache

13.1 Overview

Multiple memory access from each DDI sub-block and late response time degrades overall performance. Therefore the internal DDI Cache is required to make up for these excessive memory access time. It can fill internal buffer (64 depth * 64bit) at once when a read request has received from sub-blocks.

13.1.1 DDI_CACHE Specific Features

- 64 depths * 64 bit buffers * 6 channels
- Support the connection path of 26 channel DMAs
  VIQE : 9 channels
  LCDC0 : 5 channels
  LCDC1 : 5 channels
  MSCL0 : 3 channels
  MSCL1 : 3 channels
  CIF : 1 channel
- Maximum connection paths are 6 for operation at the same time

13.2 Block Diagram of DDI_CACHE

The block diagram of DDI_CACHE is shown in next figure. It is composed of six buffers(64depth*64bit) and is connected with read DMA path of each sub-block such as VIQE, LCDC, MSCL, CIF.

PORT_MUX can assign only 6 request channels out of 26 request channels and 6 DMA slaves are access from these channels. Other channels, not assigned to DMA blocks, will read data directly from memory.

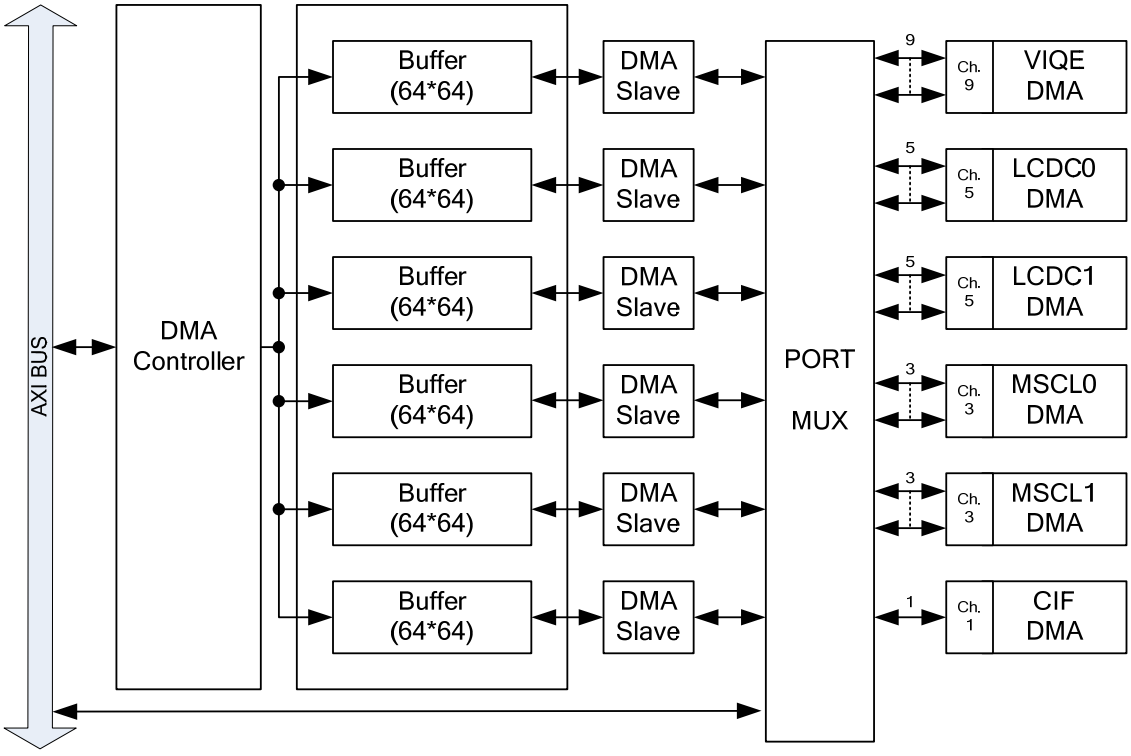


Figure 13.1 Block diagram of DDI_CACHE in TCC8900

13.3 Register Description

Table 13.1 DDI_CACHE Register Map (Base Address = 0xF0250000)

Name	Address	Type	Reset	Description
DDIC_CTRL	0x000	R/W	0x00000000	DDI_CACHE Control
DDIC_CFG0	0x004	R/W	0x00000000	DDI_CACHE Configuration #0
DDIC_CFG1	0x008	R/W	0x00000000	DDI_CACHE Configuration #1

DDIC_CTRL (DDI_CACHE Control) 0xF0250000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BW	0					ENI[25:16]									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ENI[15:0]															

BW [31]	BWRAP
0	
1	

ENI[25:0]	Enable Connection Path
ENI[25]	CIF_DMA
ENI[24]	DDIC_VIQE_DMA2_2
ENI[23]	DDIC_VIQE_DMA2_1
ENI[22]	DDIC_VIQE_DMA2_0
ENI[21]	DDIC_VIQE_DMA1_2
ENI[20]	DDIC_VIQE_DMA1_1
ENI[19]	DDIC_VIQE_DMA1_0
ENI[18]	DDIC_VIQE_DMA0_2
ENI[17]	DDIC_VIQE_DMA0_1
ENI[16]	DDIC_VIQE_DMA0_0
ENI[15]	DDIC_MSCL1_DMA_2
ENI[14]	DDIC_MSCL1_DMA_1
ENI[13]	DDIC_MSCL1_DMA_0
ENI[12]	DDIC_MSCL0_DMA_2
ENI[11]	DDIC_MSCL0_DMA_1
ENI[10]	DDIC_MSCL0_DMA_0
ENI[9]	DDIC_LCD1_DMA_2
ENI[8]	DDIC_LCD1_DMA_1
ENI[7]	DDIC_LCD1_DMA_0_2
ENI[6]	DDIC_LCD1_DMA_0_1
ENI[5]	DDIC_LCD1_DMA_0_0
ENI[4]	DDIC_LCD0_DMA_2
ENI[3]	DDIC_LCD0_DMA_1
ENI[2]	DDIC_LCD0_DMA_0_2
ENI[1]	DDIC_LCD0_DMA_0_1
ENI[0]	DDIC_LCD0_DMA_0_0

If ENI[x] == 1, connection path is enable, the other is disable.

DDIC_CFG0 (DDI_CACHE Configuration #0) 0xF0250004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			SEL27					0			SEL26				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0															

DDIC_CFG1 (DDI_CACHE Configuration #1) 0xF0250008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			SEL31					0			SEL30				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0			SEL29					0			SEL28				

SEL_26 ~SEL_31	CACHE Selection
0	DDIC_LCD0_DMA_0_0
1	DDIC_LCD0_DMA_0_1
2	DDIC_LCD0_DMA_0_2
3	DDIC_LCD0_DMA_1
4	DDIC_LCD0_DMA_2
5	DDIC_LCD1_DMA_0_0
6	DDIC_LCD1_DMA_0_1
7	DDIC_LCD1_DMA_0_2
8	DDIC_LCD1_DMA_1
9	DDIC_LCD1_DMA_2
10	DDIC_MSCL0_DMA_0
11	DDIC_MSCL0_DMA_1
12	DDIC_MSCL0_DMA_2
13	DDIC_MSCL1_DMA_0
14	DDIC_MSCL1_DMA_1
15	DDIC_MSCL1_DMA_2
16	DDIC_VIQE_DMA_0_0
17	DDIC_VIQE_DMA_0_1
18	DDIC_VIQE_DMA_0_2
19	DDIC_VIQE_DMA_1_0
20	DDIC_VIQE_DMA_1_1
21	DDIC_VIQE_DMA_1_2
22	DDIC_VIQE_DMA_2_1
23	DDIC_VIQE_DMA_2_2
24	DDIC_VIQE_DMA_2_3
25	DDIC_CIF_DMA

The SEL26 ~ SEL31 is the buffer of cache. If Cache is enable (ENI field is set), you choose buffer that connect to each DMA.

PART7 – VIDEO BUS

TCC8900

High Performance and Low-Power Processor

For Digital Media Applications

Rev. 1.00

Aug 07, 2009

TelecChips

Revision History

Date	Revision	Description
2008-12-11	0.00	Initial release
2009-01-30	0.01	- update page number
2009-02-25	0.02	- update page number
2009-05-04	0.03	- update page number
2009-08-07	1.00	- Deleting Change Log

TABLE OF CONTENTS

Contents

1 Introduction 1-1
2 Bus Architecture 2-3
3 Address and Register Map 3-5
4 Video Codec 4-7
4.1 Overview 4-7
4.2 Features 4-7
4.3 Clock and Reset 4-8
4.3.1 Clock 4-8
4.3.1.1 AXI Bus Clock, aclk 4-8
4.3.1.2 Video Decoder Clock, cclk 4-8
4.3.1.3 APB Bus Clock, pclk 4-9
4.4 Frame Buffer 4-9
4.4.1 Memory Map 4-9
4.4.2 Mapping Picture to Frame Buffers 4-9
4.4.3 Frames Buffers for Luminance Component 4-11
4.4.4 Frame Buffers for Chrominance Component 4-11
4.4.5 Frame Buffer Size in SDRAM 4-12
5 Video Cache 5-15
5.1 Resister Map 5-15
6 JPEG Codec 6-19
6.1 Overview 6-19
6.2 JPEG Encoder Register 6-20
6.3 JPEG Decoder Register 6-28
7 Video Bus Configuration 7-39

Figures

Figure 2.1 The Video Bus Architecture 2-3
Figure 4.1 Frame Buffer Configuration 4-11
Figure 4.2 Luminance Pixel Arrangement in Frame Buffer 4-11
Figure 4.3 Chrominance Pixel Arrangement in Separate Cb/Cr Buffer 4-12
Figure 4.4 Chrominance Pixel Arrangement in Interleaved Cb/Cr Buffer 4-12
Figure 6.1 JPEC Block Diagram 6-19

Tables

Table 4.1 Detailed Features of the CODEC 4-7
Table 4.2 The Example Operating Frequency for Each Standard and Bitrate in D1 4-8
Table 5.1 Video Cache Registers (Base Address = 0xF0701000) 5-15
Table 6.1 JPEG Encoder Registers (Base Address = 0xF072000) 6-20
Table 6.2 JPEG Decoder Registers (Base Address = 0xF071000) 6-28

1 Introduction

TCC8900 VIDEOBUS includes multi-standard Video Codec and JPEG Codec.

[Features]

- VIDEO INTERFACES
  - Block Diagram
  - Pipelines
  - Bus Architecture
  - Clock and Reset
  - Frame Buffer
  - The Bit Processor Program and Data Memory
  - Bit Processor
  - Video Codec
  - Register Descriptions

- JPEG INTERFACES
  - Encoder
  - Decoder

2 Bus Architecture

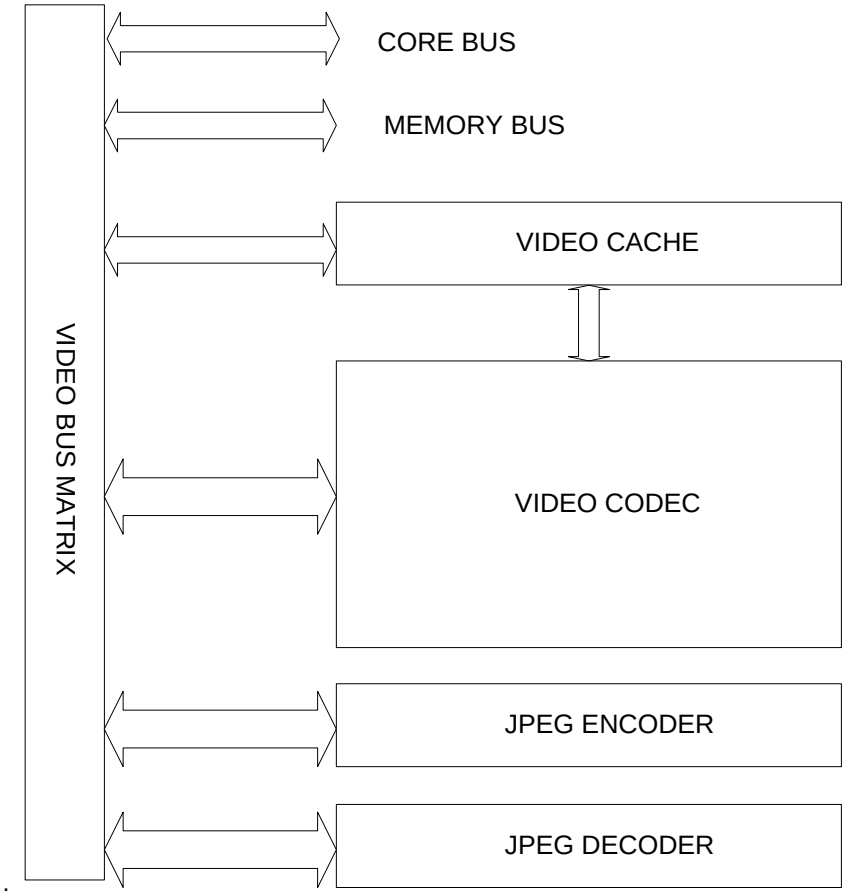


Figure 2.1 The Video Bus Architecture

3 Address and Register Map

Base Address		Peripherals
0xF0700000		VIDEO CODEC
0xF0710000		JPEG Decoder
0xF0720000		JPEG Encoder
0xF0701000		VIDEOCACHE

4 Video Codec

4.1 Overview

The TCC8900 VIDEO CODEC is a high performance multi-standard video codec IP that can perform the H.264 BP/MP/HP, VC-1 SP/MP/AP, MPEG-4 SP/ASP, Divx, MPEG-1/2, and RV-8/9/10 decoding. It can encode or decode multiple video clips with multiple standards simultaneously.

It connects with the system via the 32-bit AMBA 3 APB bus for system control and 64-bit AMBA3 AXI for data throughput, and takes advantage of on-chip memories to achieve high performance.

The VIDEO CODEC has a 16-bit telechips proprietary DSP, called as BIT processor. The BIT processor controls the internal video codec sub blocks and communicates with a host processor through the host interface. The required resource for the host CPU to control the VIDEO CODEC is very low, under 1 MIPS, because all functions such as rate control, FMO, ASO, video codec control and error resilience are implemented in the BIT processor. The most part of sub-blocks in the VIDEO CODEC is optimally shared, which enables to achieve the ultra low power and low gate count.

4.2 Features

The main features of VIDEO CODEC are fully compliant with H.264 BP/MP/HP, VC-1 SP/MP/AP, MPEG-4 SP/ASP(except GMC), Divx (Xvid), MPEG-1/2, and RV-8/9/10. The VIDEO CODEC can support various error resilience tools and also support multiple decoding and full duplex multi-party-call simultaneously.

The VIDEO CODEC is easy to integrate into the system because its interface is composed of general and simple AXI/APB. And the VIDEO CODEC provides programmability, flexibility and ease of upgrade in decoding/encoding or host interface because all the controls in decoding/encoding process and host interface are implemented as firmware in a programmable BIT processor.

The detailed features of the VIDEO CODEC are as follows.

Table 4.1 Detailed Features of the CODEC

	Standard	Profile	Level
Encoder	H.264	Baseline	3.0
	MPEG-4	SP	
	H.263	Profile 3	
Decoder	H.264	BP/MP/HP	5.1
	MPEG-4	ASP	
	H.263	Profile 3	
	VC-1	SP/MP/AP	
	MPEG-2	Main	High
	Divx(Xvid)	Home theater	
	RV	8/9/10	

- Encoding Tools
  - [±16, ±16] 1/2 and 1/4-pel accuracy motion estimation
  - 16x16, 16x8, 8x16 and 8x8 block sizes are supported.
  - Available block sizes can be configurable.
  - The encoder uses only one reference frame for the motion estimation.
  - Unrestricted motion vector
  - Prediction
    - MPEG-4 AC/DC prediction
    - H.264/AVC intra-prediction
  - H.263 Annex J, K (RS=0 and ASO=0), and T are supported.
  - Error resilience tools
    - MPEG-4 resync marker & data-partitioning with RVLC (Fixed number of bits/macroblocks between macroblocks)
    - CIR (Cyclic Intra Refresh)/AIR (Adaptive Intra Refresh)
    - Bit-rate control (CBR & VBR)
  - Minimum encoding image size is 16 pixels in horizontal and16 pixels in vertical.
- Decoding Tools
  - H.264
    - Fully compatible with the ITU-T Recommendation H.264 specification in BP/MP and HP.
    - Supports CABAC/CAVLC

- Variable block size (16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4)
- Error detection, concealment and error resilience tools

- VC1
  - Supports all VC-1 profile features – SMPTE “Proposed SMPTE Standard for Television: VC-1 Compressed Video Bitstream format and Decoding Process”
  - Supports Simple/Main/Advanced Profile
  - Multi-resolution (Dynamic resolution) is not processed inside of video decoder
- MPEG-4
  - Supports Simple/Advanced Simple profile except GMC.
  - Supports H.263 Baseline Profile
  - Support Divx from ver3.x to ver6.x
  - Supports Xvid
- MPEG-2
  - Fully compatible with ISO/IEC 13182-2 MPEG2 specification in Main Profile.
  - Support I,P and B frame
  - Support field coded picture (interlaced) and fame coded picture
- RV-8/9/10
  - Fully compatible with RV-8/9/10 except re-sampling feature.

- Value added features
  - MPEG-2 partial acceleration
  - Pre/Post rotator/mirror
  - Built-in de-blocking filter for MPEG-2/MPEG-4 and Divx
- Programmability
  - The VIDEO CODEC embeds telechips proprietary 16-bit DSP processor dedicated to processing bitstream and controlling the codec hardware.
  - General purpose registers and interrupt for communication between a host processor and the video IP
- Interrupt
  - Interrupt from/to external host processor or interrupt controller
- Power related signal
  - Supports VPU_IDLE
  - Supports VPU_UNDERRUN

4.3 Clock and Reset

4.3.1 Clock

The VIDEO CODEC VPU uses 3 clock sources for the AXI bus (ACLK), the APB bus (PCLK), and video decoder module (CCLK), and 1 extra clock for system clock controller (RCLK). Based on the GALS (Globally Asynchronous Locally Synchronous) architecture of the IP, each clock source can be asynchronous to another. The duty cycles of all clocks are “don’t care” because the VIDEO CODEC VPU uses only rising edge of them..

4.3.1.1 AXI Bus Clock, aclk

The VIDEO CODEC VPU uses the AMBA AXI bus architecture as the system bus to interface with an external memory. All signals of the interface are output from an internal AXI register slice. The required frequency of the aclk is highly depends on the bus-loading, bandwidth, of user’s system.

4.3.1.2 Video Decoder Clock, cclk

The CCLK is for the video decoder module of the VIDEO CODEC VPU. VIDEO CODEC VPU decodes D1 30fps @ 30~50MHz. Below table shows the required frequencies of the cclk for a few examples.

Table 4.2 The Example Operating Frequency for Each Standard and Bitrate in D1

Standard	Operating Frequency	Bit Rate
MPEG-2 MP	40MHz	12Mbps
	35MHz	8Mbps
MPEG-4 ASP(Divx)	40MHz	5Mbps
	35MHz	4Mbps
H.264 HP	40MHz	4Mbps
VC-1 AP	40MHz	4Mbps

Note: This value is estimated by proprietary simulation environment.

4.3.1.3 APB Bus Clock, pclk

The PCLK is used for the APB bus. A host processor can communicate with the VIDEO CODEC VPU through the bus. The clock frequency does not have any relationship with other clock sources.

4.4 Frame Buffer

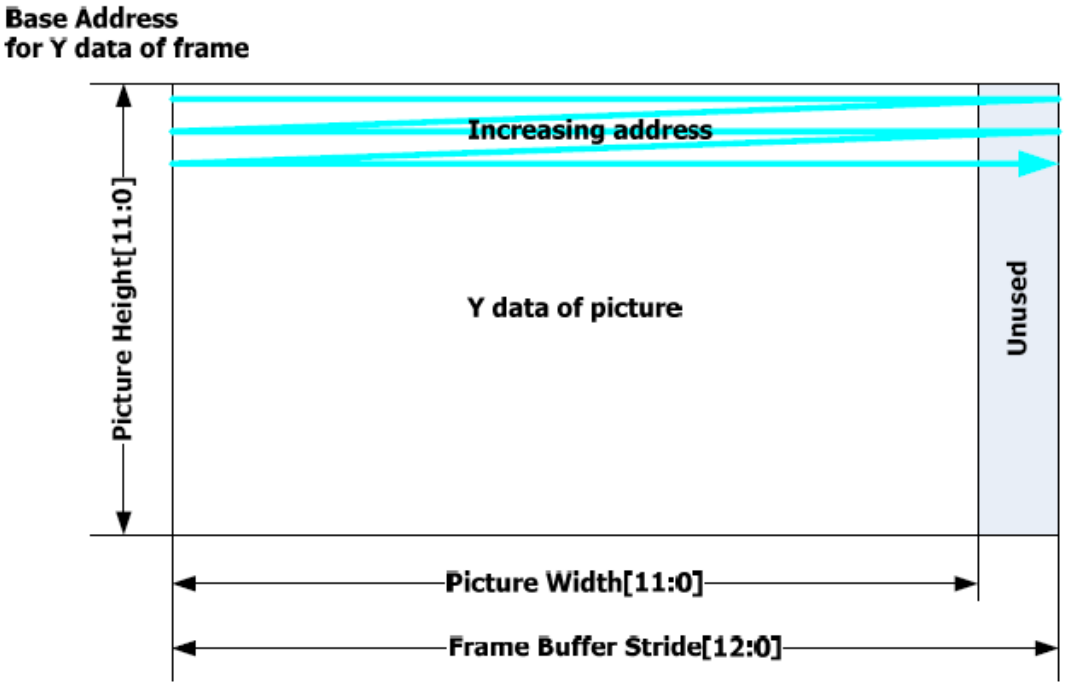
4.4.1 Memory Map

The source and decoded pictures (frames or fields) are each comprised of three sample arrays, one luma and two chroma sample arrays. In this document, the frame buffer means the sample array in an external memory. Following are the features of the frame buffer:

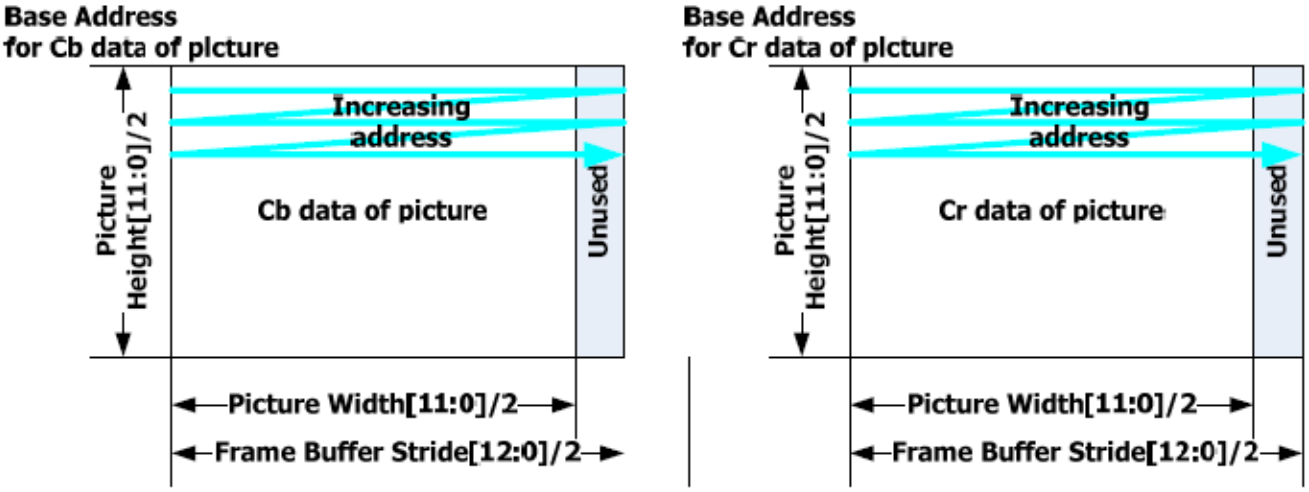
- A frame buffer is configured with following parameters:
  - Luminance frame buffer base address in 8-bye alignment
  - Cb frame buffer base address in 8-byte alignment
  - Cr fame buffer base address in 8-byte alignment: In case of the interleaved Cb/Cr map, a base address for the Cr is meaningless because a base address for the Cb is used to store or load the interleaved Cb/Cr samples.
  - Stride for the width of the luminance frame buffer: The stride should be equal or greater than the width of picture and multiple of 8-It means the least significant 3-bit of the 13-bit stride should be always 0. Stride of the chrominance frame buffers is the half of the stride for the luminance.
  - Endianess of the frame buffers
- The internal registers for specifying a frame buffer and a picture size are 12-bit width.
  - The stride for specifying the width of frame buffer in pixel unit
  - The picture size in pixel unit for both horizontal and vertical direction
- An endianess of the frame buffers is configurable as the little or the big endian.
- Both the interleaved Cb/Cr map and the separate Cb/Cr map are supported.
- A field pair is stored in a single frame buffer.

4.4.2 Mapping Picture to Frame Buffers

The Figure 4.1 shows how the frame buffer is configured. There are two kinds of configuration. The first configuration is const of (a) and (b). In the configuration, each chroma component has its own buffer. Therefore, there are two buffers for Cb and Cr components. The second configuration is consist of (a) and (c). This configuration is for reduction of bus-load using longer burst-length than the 1st configuration. As shown in the (c), the Cb and Cr samples are stored in the interleaved way.



(a) Frame buffer for luminance component



(b) Frame buffer for non-interleaved Cb & Cr components

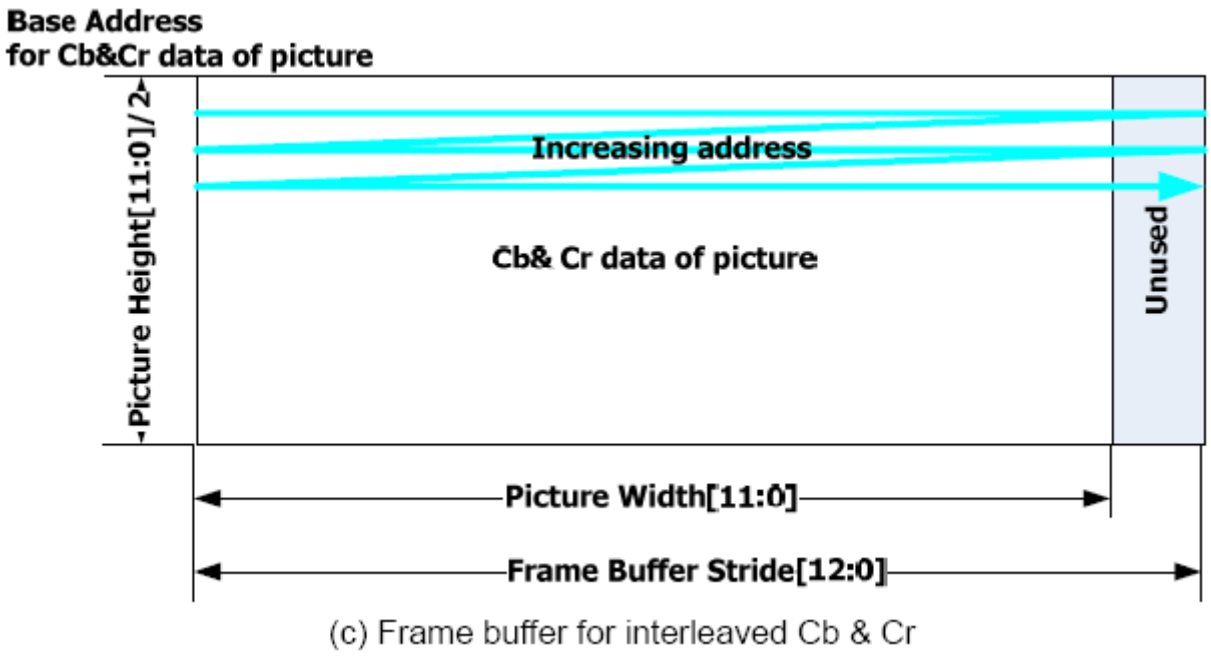


Figure 4.1 Frame Buffer Configuration

The following chapters give a detailed description for these configurations considering endianess by giving examples for QCIF(176x144) image.

4.4.3 Frames Buffers for Luminance Component

The Figure 4.2 shows how luminance samples are mapped to external memory. The base address in the figure is for luminance frame buffer.

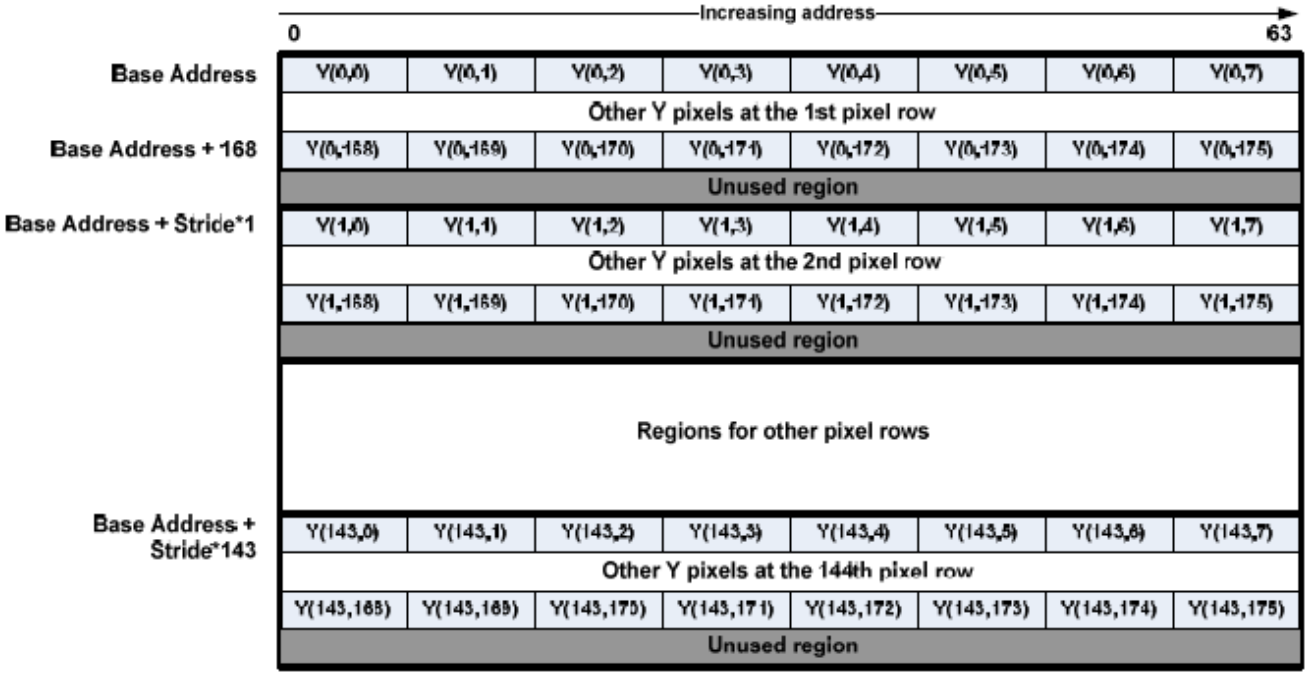


Figure 4.2 Luminance Pixel Arrangement in Frame Buffer

4.4.4 Frame Buffers for Chrominance Component

The Figure 4.3 shows how chrominance samples are mapped to separate Cb/Cr buffer of external memory. Each chrominance component has its own buffer.

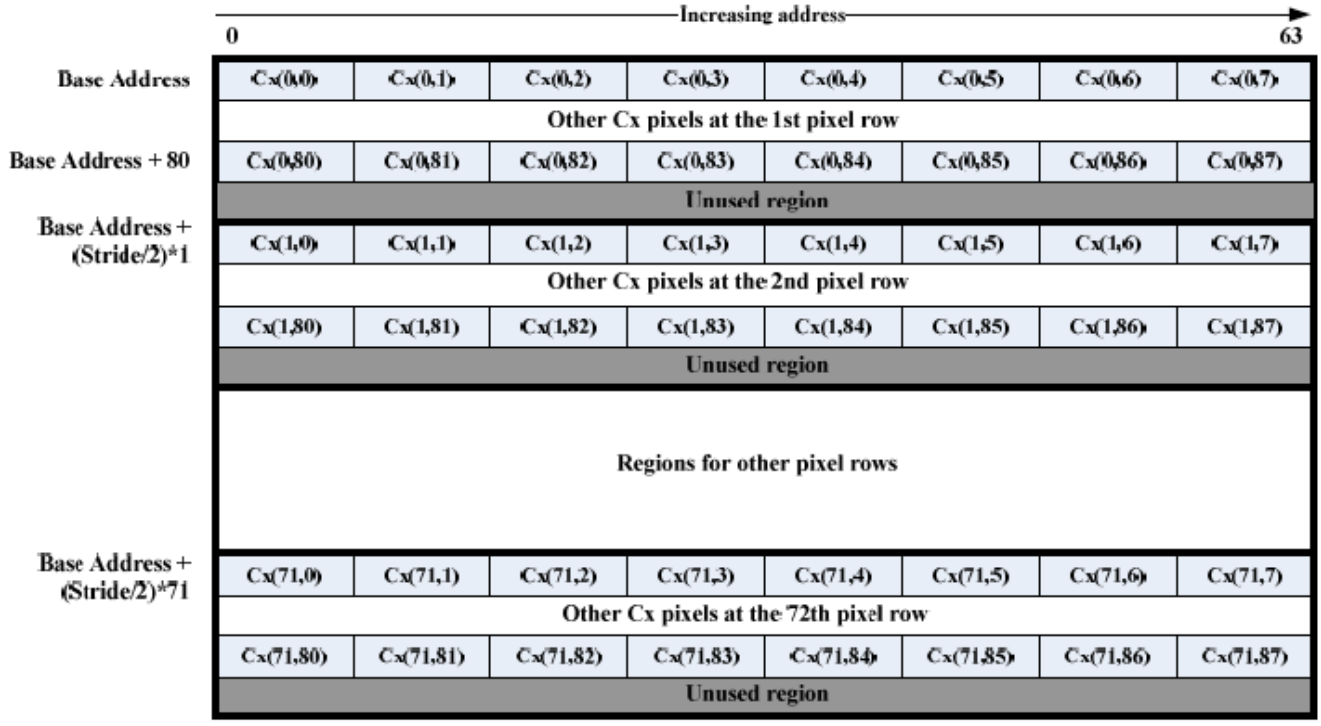


Figure 4.3 Chrominance Pixel Arrangement in Separate Cb/Cr Buffer

The Figure 4.4shows how chrominance samples are mapped to interleaved Cb/Cr buffer of external memory. In the interleaved Cb/Cr map, only one buffer is assigned for Cb and Cr of picture. Its stride is the same as the luminance's.

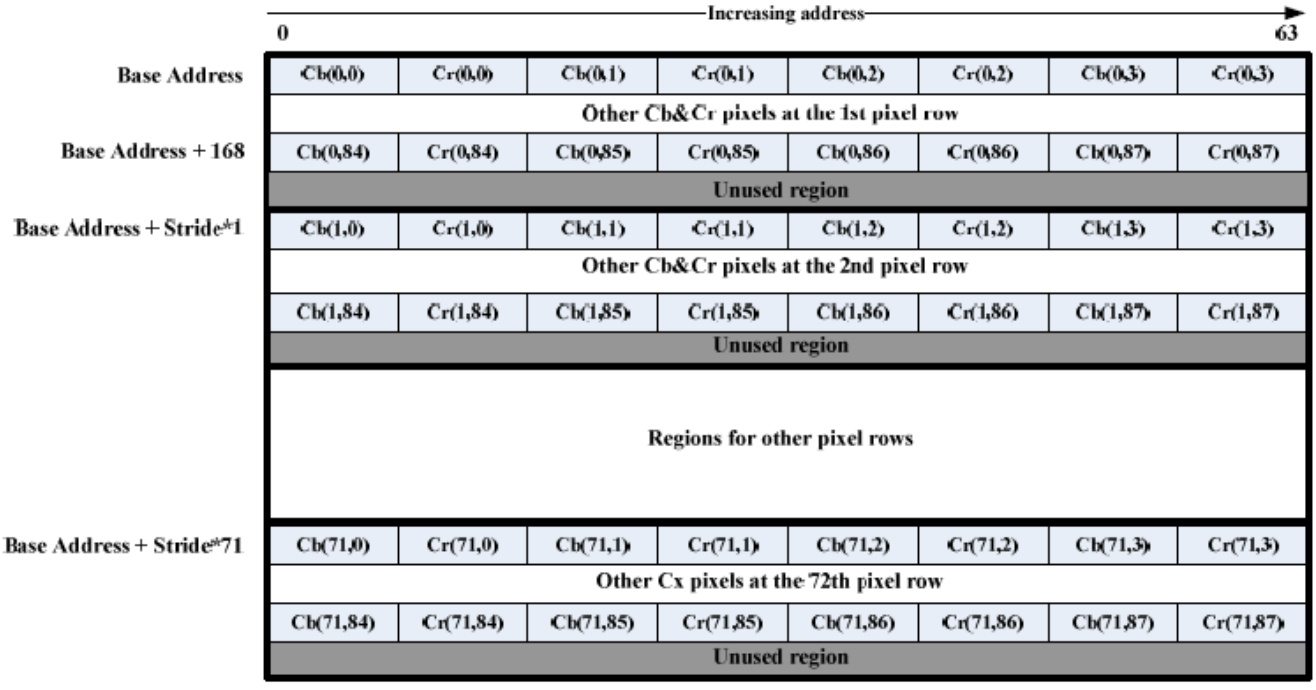


Figure 4.4 Chrominance Pixel Arrangement in Interleaved Cb/Cr Buffer

4.4.5 Frame Buffer Size in SDRAM

The required frame buffer size in SDRAM is different for each standard. Below table shows the required memory (SDRAM) size for each standard to support D1 (720x576). These values are the worst case which is defined by corresponding specification.

#		H.264	VC-1	MPEG-4	MPEG-2
1	Frame Buffer	7 frame (4,252.5 Kbyte)	7 frame (4,252.5 Kbyte)	7 frame (4,252.5 Kbyte)	7 frame (4,252.5 Kbyte)
2	Direct motion vector	708.75 Kbyte	25.4 Kbyte	25.4 Kbyte	
3	Overlap filter		7.1 Kbyte		
4	De-blocking filter	11.25 Kbyte	22.5 Kbyte		
5	Intra prediction(ACDC)	4.22 Kbyte	5.625 Kbyte	5.625 Kbyte	
6	MVP/MB information	5.625 Kbyte	2.11 Kbyte	2.11 Kbyte	
7	Slice information	131 Kbyte			
8	Bit Plane		3 Kbyte		
9	Data-partitioning			96 Kbyte	
	Total	5 Mbyte	3.6 Mbyte	2.48 Mbyte	2.38 Mbyte

The above table shows the SDRAM requirement for each standard. The SDRAM requirement for number 3, 4, 5, and 6 can be removed if user uses secondary AXI for internal SRAM.

Note : picX=720, picY=576
mbX (Number of macroblock in row) =45.
mbY (Number of macroblock in column) =36.
1 frame for D1 is 720 x 576 x 1.5 = 622,080 byte.

5 Video Cache

5.1 Resister Map

Table 5.1 Video Cache Registers (Base Address = 0xF0701000)

Name	Addr	Type	Mode	Reset	Description
VCCTRL	0x000	R/W	-	0x00000000	Video cache control register
VCREG	0x004	R/W	-	0x00000000	Video cache read/write register
VWBCTRL	0x008	R/W	-	0x00000000	Video cache write valid register
R0MIN	0x024	R/W	-	-	Video cache read reason 0 minimum register
R0MAX	0x028	R/W	-	-	Video cache read reason 0 maximum register
R1MIN	0x02c	R/W	-	-	Video cache read reason 1 minimum register
R1MAX	0x030	R/W	-	-	Video cache read reason 1 maximum register
R2MIN	0x034	R/W	-	-	Video cache read reason 2 minimum register
R2MAX	0x038	R/W	-	-	Video cache read reason 2 maximum register
R3MIN	0x03c	R/W	-	-	Video cache read reason 3 minimum register
R3MAX	0x040	R/W	-	--	Video cache read reason 3 maximum register

VCCTRL **0xF0701000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWRASVE															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
														INVAL	ON

Field	Name	RW	Reset	Description
0	ON	R/W	0x00000000	Cache On
1	INVAL	R/W	0x0	Prefetch Cache Invalidate All
31	PWRASVE	R/W	0x00000000	Not-used

VCREG **0xF0701004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		RD3	RD3			WR2	RD2			WR1	RD1			WR0	RD0

Field	Name	RW	Reset	Description
0	RD0	R/W	0x00000000	Read region 0 enable.
1	WR0	R/W		Write region 0 enable.
4	RD1	R/W		Read region 1 enable.
5	WR1	R/W		Write region 1 enable.
8	RD2	R/W		Read region 2 enable.
9	WR2	R/W		Write region 2 enable.
12	RD3	R/W		Read region 3 enable.
13	WR3	R/W		Write region 3 enable.

VWBCTRL **0xF0701008**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TVALUE															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
										WCBV		WCRST	DRAINST	TEN	DRAIN

Field	Name	RW	Reset	Description
0	DRAIN	R/W	0x00000000	Write Buffer Drain enable 0 : not-drain 1 : forced drain
1	TEN	R/W		Timeout enable. 0 : Write buffer is not drained when timeout. 1 : Write buffer will be drained when timeout.
2	DRAINST	R		Read only drain status.
3	WCRST	R/W		Write Cache Reset. 0 : Not-Reset 1 : Reset
5	WCBV	R/W		Write Access Wait Control Option Register. 0 : Wait until the data stored to the external memory such as DRAM. 1 : Wait until the data stored to the write buffer. Normally, the write time can faster than case 0.
31-16	TVALUE	R/W		Timeout value.

R0MIN **0xF0701024**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0 MIN															

Field	Name	RW	Reset	Description
31-00	R0MIN	R/W		Lower bound address for region 0

R0MAX **0xF0701028**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0 MAX															

Field	Name	RW	Reset	Description
31-00	R0MAX	R/W		Upper bound address for region 0

R1MIN **0xF070102C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R1 MIN															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1 MIN															

Field	Name	RW	Reset	Description
31-00	R1MIN	R/W		Lower bound address for region 1

R1MAX **0xF0701030**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R1 MAX															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1 MAX															

Field	Name	RW	Reset	Description
31-00	R1MAX	R/W		Upper bound address for region 1

R2MIN **0xF0701034**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R2 MIN															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R2 MIN															

Field	Name	RW	Reset	Description
31-00	R2MIN	R/W		Lower bound address for region 2

R2MAX **0xF0701038**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R2 MAX															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R2 MAX															

Field	Name	RW	Reset	Description

31-00	R2MAX	R/W		Upper bound address for region 2

R3MIN **0xF070103C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R3 MIN															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R3 MIN															

Field	Name	RW	Reset	Description
31-00	R3MIN	R/W		Lower bound address for region 3

R3MAX **0xF0701040**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R3 MAX															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R3 MAX															

Field	Name	RW	Reset	Description
31-00	R3MAX	R/W		Upper bound address for region 3

6 JPEG Codec

6.1 Overview

The JPEG block of TCC8900 is a high performance solution for image and video compression/decompression applications. It can *compress* 24Mbyte/sec data stream in real-time and compliance with the baseline ISO/IEC 10918-1 JPEG standard. Additionally, the JPEG block can be used as DCT/IDCT calculator for video encoding and decoding.

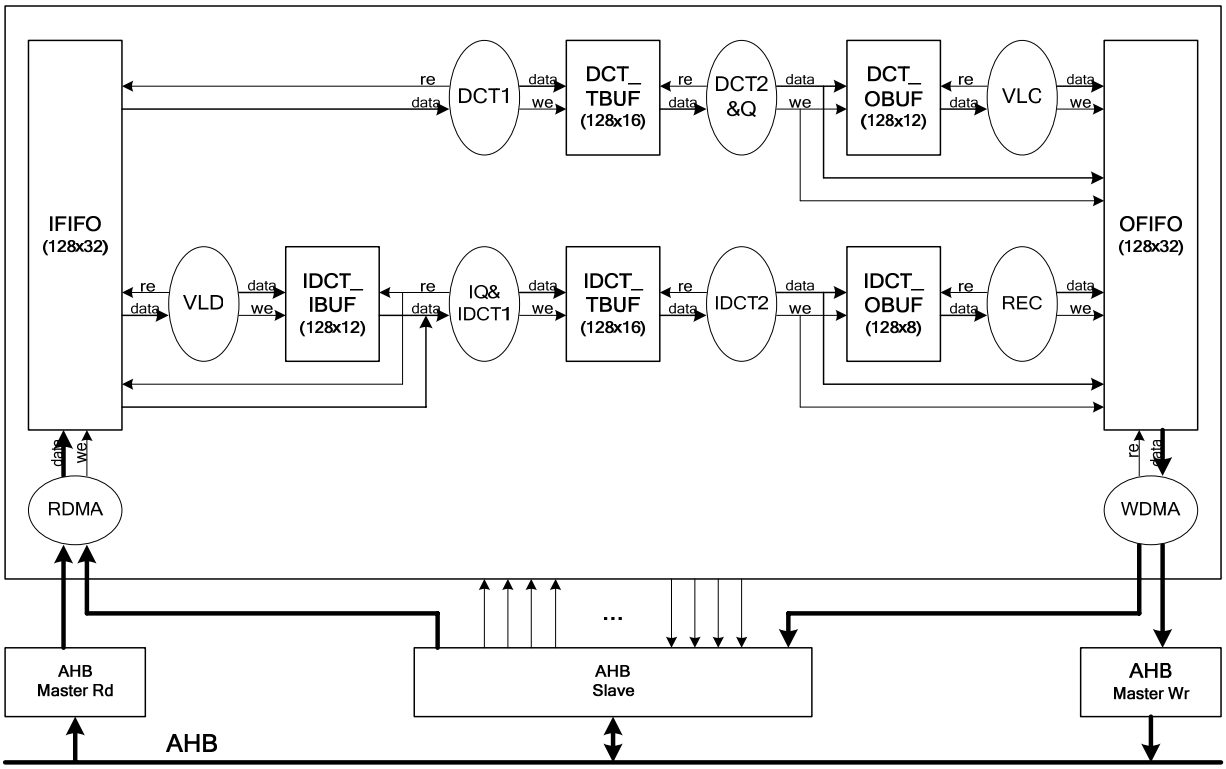


Figure 6.1 JPEC Block Diagram

The JPEG block compress any image size up to 4096x4096 and the image samples according to the user-defined quantization and Huffman tables, and it produces an ISO/IEC 10918-1 compatible data stream. Each 8x8 image block is frequency transformed by a forward DCT into the domain of 2-dimensional DCT(2D-DCT) basis images. Image samples are frequency transformed from the DCT unit using 15-bit internal accuracy. The 11-bit output DCT coefficients are then scanned int the Zig-Zag order and quantized according to programmed Quantization tables.

The quantization prepares the blocks for efficient coding. Each DCT coefficient block is divided by an 8x8 quantization matrix. The quantized DCT coefficients are then fed to the differential coding and run-length coding unit that produces the Run-Amplitude pairs for the Huffman coder. The Huffman encoder accepts symbols from RLE unit, which are further compressed by Huffman encoder, which encodes them according to one of the two possible programmed Huffman tables. Finally, Huffman compressed data stream is written to the output FIFO. The AHB bus master stores data in output FIFO to the predefined address pointer. The software header generator makes the JPEG file header and merges with the body, generated by the JPEG block.

The JPEG block is capable of decoding Baseline ISO/IEC 10918-1 JPEG streams. It outputs decoded image samples in MCU by MCU, raster scan order. The software header parser parses the JPEG file header and configs Huffman and quantization table. Then set the file body pointer and run the JPEG decoder. The AHB bus master reads the body and stores in input FIFO. These data fed into the VLD block and decoded. The Inverse Quantization block multiplies each coefficient according to the quantization tables and produces the 11 bit DCT coefficients. These are then scanned in the inverse Zig-Zag order and stored in the IDCT buffer. The IDCT is then applied on the coefficients, to finally produce the image samples. When that image is too big to stored in memory, that can be decimated by 1/4 or 1/16.

6.2 JPEG Encoder Register

Table 6.1 JPEG Encoder Registers (Base Address = 0xF072000)

Name	Addr	Type	Mode	Reset	Description
JP_MOD	0x004	R/W	ALL	0x00000000	JPEG codec mode register
JP_INT_MASK	0x008	R/W	ALL	0x0000001f	Interrupt mask register
JP_INT_LEVEL	0x00c	R/W	SLV	0x000000ff	FIFO interrupt level register
JP_TRG_MOD	0x010	R/W	ALL	0x00000000	Polling or Interrupt mode selection register
R_YBUF_ADDR	0x020	R/W	JP	0x00000000	Raw data buffer Y address register
R_UBUF_ADDR	0x024	R/W	JP	0x00000000	Raw data buffer U address register
R_VBUF_ADDR	0x028	R/W	JP	0x00000000	Raw data V address register
R_BUF_INFO	0x02c	R/W	JP	0x00000000	Raw data buffer information register
JP_SIZE	0x030	R/W	JP	0x00000000	Image size information register
JP_CHROMA	0x034	R/W	JP	0x00000000	Image format information register
JP_CBUF_ADDR	0x38	R/W	JP	0x00000000	Coded data buffer address register
JP_CBUF_SIZE	0x03c	R/W	JP	0x00000fff	Coded data buffer size register
JP_START	0x070	W	ALL	0x00000000	Codec start command register
JP_SBUF_WCNT	0x080	R/W	MST	0x00000000	Source buffer write count register
JP_SBUF_RCNT	0x084	R	MST	0x00000000	Source buffer read count register
JP_DBUF_WCNT	0x088	R	MST	0x00000000	Destination buffer write count register
JP_DBUF_RCNT	0x08c	R/W	MST	0x00000000	Destination buffer read count register
JP_IFIFO_ST	0x090	R	SLV	0x00000000	Input FIFO status register
JP_OFIFO_ST	0x094	R	SLV	0x00000000	Output FIFO status register
JP_INT_FLAG	0x0a0	R	ALL	0x00000000	Interrupt flag register
JP_INT_ACK	0x0a4	R	ALL	0x00000000	Interrupt ack register
JP_IFIFO_WD	0x0c0	W	SLV	0x00000000	Input FIFO write data register
JP_OFIFO_RD	0x0e0	R	SLV	0x00000000	Output FIFO read data register
JPC_QTAB0	0x100 -	W	JPC	0x00000000	Encoder Q table 0 (64 entries)
JPC_QTAB1	0x200 -	W	JPC	0x00000000	Encoder Q table 1 (64 entries)

JPEG Codec Mode Register(JP_MODE) 0xF7200004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														OPM	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
															IPM

Field	Name	RW	Reset	Description
17-16	OPM	RW	0	Operation Mode 0 : Encoding Mode
0	IPM	RW	0	Input Mode 0 : Master Input Mode 1 : Slave Input Mode

Interrupt Mask(JP_INT_MASK) 0xF0720008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
										IMSK					

In the interrupt mode, those bits can mask the interrupt.

IMSK[5:0]	Description
n	IMSK[5] : Raw Data Buffer IMSK[4] : Coded Buffer IMSK[3] : Input FIFO IMSK[2] : Output FIFO IMSK[1] : Error IMSK[0] : Operation End.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Field	Name	RW	Reset	Description
17-16	OPM	RW	0	Operation Mode 0 : Encoding Mode
0	IPM	RW	0	Input Mode 0 : Master Input Mode 1 : Slave Input Mode

Input FIFO Interrupt Level Register(JP_INT_LEVEL) 0xF072000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								IFIFO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								OFIFO							

IFIFO[7:0]	Description
n	Input FIFO interrupt level. (0 ~ 127)

OFIFO[7:0]	Description
n	Output FIFO interrupt level (0 ~ 127)

Trigger Mode Register(JP_TRG_MOD) 0xF0720010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
															TMD

TMD[0]	Description
n	0 : Polling mode. 1 : Interrupt mode.

Raw Data Buffer Address Register (R_Y/U/VBUF_A) 0xF07200(20/24/28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R_BUF_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R_BUF_ADDR[15:2]															

YBUF_ADDR[31:2]	Description
n	The raw buffer Y start address.

UBUF_ADDR[31:2]	Description
n	The raw buffer U start address.

VBUF_ADDR[31:2]	Description
n	The raw buffer V start address.

Raw Buffer Information Register (R_BUF_INFO) 0xF072002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				IFRM_HSIZE											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				IFRM_VSIZE											

FRM_HSIZE[11:0]	Description
n	Input buffer horizontal size (unit == pixel)

FRM_VSIZE[11:0]	Description
n	Input buffer vertical size (unit == line)

Image Size Register (IMG_SIZE) 0xF0720030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				IMG_HSIZE[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				IMG_VSIZE[11:0]											

IMG_HSIZE[11:0]	Description
n	Image horizontal size (unit == pixel)

IMG_VSIZE[11:0]	Description
n	Image vertical size (unit == line)

Image Format Register(CHROMA) 0xF0720034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													CHROMA		

CHROMA[2:0]	Description
1	YUV420 (Y,U,V separated mode)
2	YUV422 (Y,U,V separated mode)

Coded Buffer Address Register(CBUF_ADDR) 0xF0720038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CBUF_ADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CBUF_ADDR[15:2]														0	0

CBUF_ADDR[11:0]	Description
n	The coded data buffer start address.. In encoding mode : encoding data output pointer

Coded Buffer Size Register (CBUF_SIZE) 0xF072003C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				CBUF_SIZE											

CBUF_SIZE[11:0]	Description
n	Coded data buffer size (unit = 1024bytes)

Block Start Register (JP_START) 0xF0720070

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
															ST

ST	Description
1	Encoding start command.

Source Buffer Write Counter (SBUF_WC) 0xF0720080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				WC											

RC[11:0]	Description
n	Source buffer write count (unit = 1line)

Source Buffer Read Counter (SBUF_RC) 0xF0720084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				RC											

RC[11:0]	Description
n	Source buffer read count (unit = 1line)

Destination Buffer Write Counter (DBUF_WC) 0xF0720088

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														WC[17:16]	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WC[15:0]															

WC[17:0]	Description
n	Destination buffer write count (unit = 16bytes)

Destination Buffer Read Counter (DBUF_RC) 0xF072008C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				RC											

RC[11:0]	Description
n	Destination buffer read count (unit = 1024bytes)

Input FIFO Status (IFIFO_ST) 0xF0720090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								FILL							

s	Description
n	Output FIFO fill count

Output FIFO Status (OFIFO_ST) 0xF0720094

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								FILL							

FILL[7:0]	Description
n	Output FIFO fill count

Interrupt Flags (INT_FLAG) 0xF07200A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
										IFLAG[5:0]					

IFLAG[5:0]	Description
5	Raw data buffer status
4	Coded buffer status
3	Input FIFO status
2	Output FIFO status
1	Decoding error
0	Job finished

Interrupt ACK Register (INT_ACK) 0xF07200A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

When read this register, Interrupt Flags are cleared.

Input FIFO Write Data Register (IFIFO_WD) 0xF07200C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDATA[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WDATA[15:0]															

In slave mode (the JPEG decoding/encoding or DCT/IDCT mode), host feds data into this register. (maximum burst size == 8)

WDATA[31:0]	Description
n	The write data register in slave mode.

Output FIFO Read Data Register (OFIFO_RD) 0xF07200E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDATA[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RDATA[15:0]															

In slave mode (the JPEG decoding/encoding), host gets result data from this register. (maximum burst size == 8)

RDATA[31:0]	Description
n	The read data register in slave mode.

Encoder Q Table 0 (EN_QTAB0) **0xF0720100-0xF07201FC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								Qdata1[11:4]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Qdata1[3:0]				Qdata0[11:0]											

Encoder Q Table 1 (EN_QTAB1) **0xF0720200-0xF07202FC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								Qdata1[11:4]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Qdata1[3:0]				Qdata0[11:0]											

EN_QTAB0(64 entries) is the JPEG encoder quantization table for Y component. (4096/qstep) and raster scan ordered.

EN_QTAB1(64 entries) is the JPEG encoder quantization table for C component. (4096/qstep) and raster scan ordered.

Qdata1[11:0]	Description
n	Encoder quantization data (64 entries)

Qdata0[11:0]	Description
n	Encoder quantization data (64 entries)

- 24bit Q Table Setting Guide

```
For(i=0; i<32; i=i+1) {
   For(j=0; j<64; j++) {
            If( (2*i+1) == hZZ[j] )
                     Wdata = qmat[0][j] << 12;
   }
   For(j=0; j<64; j++) {
            If( (2*i) == hZZ[j] )
                     Wdata = Wdata + qmat[0][j];
   }
   Regw(QtableBase+i*4, Wdata);
}
```

6.3 JPEG Decoder Register

Table 6.2 JPEG Decoder Registers (Base Address = 0xF071000)

Name	Addr	Type	Mode	Reset	Description
JP_MOD	0x004	R/W	ALL	0x00000000	JPEG codec mode register
JP_INT_MASK	0x008	R/W	ALL	0x0000001f	Interrupt mask register
JP_INT_LEVEL	0x00c	R/W	SLV	0x000000ff	FIFO interrupt level register
JP_TRG_MOD	0x010	R/W	ALL	0x00000000	Polling or Interrupt mode selection register
R_YBUF_ADDR	0x020	R/W	JP	0x00000000	Raw data buffer Y address register
R_UBUF_ADDR	0x024	R/W	JP	0x00000000	Raw data buffer U address register
R_VBUF_ADDR	0x028	R/W	JP	0x00000000	Raw data V address register
R_BUF_INFO	0x02c	R/W	JP	0x00000000	Raw data buffer information register
JP_SIZE	0x030	R/W	JP	0x00000000	Image size information register
JP_CHROMA	0x034	R/W	JP	0x00000000	Image format information register
JP_CBUF_ADDR	0x38	R/W	JP	0x00000000	Coded data buffer address register
JP_CBUF_SIZE	0x03c	R/W	JP	0x00000fff	Coded data buffer size register
JPD_TBL_ID	0x050	R/W	JPD	0x00000000	Decoder table index register
JPD_RST_INTV	0x054	R/W	JPD	0x00000000	Decoder reset interval register
JPD_OUT_SCL	0x058	R/W	JPD	0x00000000	Decoder output scaling register
JP_START	0x070	W	ALL	0x00000000	Codec start command register
JP_SBUF_WCNT	0x080	R/W	MST	0x00000000	Source buffer write count register
JP_SBUF_RCNT	0x084	R	MST	0x00000000	Source buffer read count register
JP_DBUF_WCNT	0x088	R	MST	0x00000000	Destination buffer write count register
JP_DBUF_RCNT	0x08c	R/W	MST	0x00000000	Destination buffer read count register
JP_IFIFO_ST	0x090	R	SLV	0x00000000	Input FIFO status register
JP_OFIFO_ST	0x094	R	SLV	0x00000000	Output FIFO status register
JP_INT_FLAG	0x0a0	R	ALL	0x00000000	Interrupt flag register
JP_INT_ACK	0x0a4	R	ALL	0x00000000	Interrupt ack register
JP_IFIFO_WD	0x0c0	W	SLV	0x00000000	Input FIFO write data register
JP_OFIFO_RD	0x0e0	R	SLV	0x00000000	Output FIFO read data register
JPD_IQTAB0	0x300 -	W	JPD	0x00000000	Decoder IQ table 0 (64 entries)
JPD_IQTAB1	0x400 -	W	JPD	0x00000000	Decoder IQ table 1 (64 entries)
JPD_IQTAB2	0x500 -	W	JPD	0x00000000	Decoder IQ table 2 (64 entries)
JPD_HT_DC0_C	0x600 -	W	JPD	0x00000000	Decoder huffman table (dc0 code, 16 entries)
JPD_HT_AC0_C	0x640 -	W	JPD	0x00000000	Decoder huffman table (ac0 code, 16 entries)
JPD_HT_DC1_C	0x680 -	W	JPD	0x00000000	Decoder huffman table (dc1 code, 16 entries)
JPD_HT_AC1_C	0x6c0 -	W	JPD	0x00000000	Decoder huffman table (ac1 code, 16 entries)
JPD_HT_DC0_A	0x700 -	W	JPD	0x00000000	Decoder huffman table (dc0 addr, 16 entries)
JPD_HT_AC0_A	0x740 -	W	JPD	0x00000000	Decoder huffman table (ac0 addr, 16 entries)
JPD_HT_DC1_A	0x780 -	W	JPD	0x00000000	Decoder huffman table (dc1 addr, 16 entries)
JPD_HT_AC1_A	0x7c0 -	W	JPD	0x00000000	Decoder huffman table (ac1 addr, 16 entries)
JPD_HT_DC0_V	0x800 -	W	JPD	0x00000000	Decoder huffman table (dc0 var, 12 entreis)
JPD_HT_AC0_V	0x840 -	W	JPD	0x00000000	Decoder huffman table (ac0 var, 162 entries)
JPD_HT_DC1_V	0xc00 -	W	JPD	0x00000000	Decoder huffman table (dc1 var, 12 entreis)
JPD_HT_AC1_V	0xc40 -	W	JPD	0x00000000	Decoder huffman table (ac1 var, 162 entries)

JPEG Codec Mode Register (JP_MODE) 0xF0710004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														OPM	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
															IPM

Field	Name	RW	Reset	Description
17-16	OPM	RW	0	Operation Mode 1 : Decoding Mode
0	IPM	RW	0	Input Mode 0 : Master Input Mode 1 : Slave Input Mode

Interrupt Mask(JP_INT_MASK) 0xF0710008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
										IMSK					

In the interrupt mode, those bits can mask the interrupt.

IMSK[5:0]	Description
n	IMSK[5] : Raw Data Buffer IMSK[4] : Coded Buffer IMSK[3] : Input FIFO IMSK[2] : Output FIFO IMSK[1] : Error IMSK[0] : Operation End.

Input FIFO Interrupt Level Register(JP_INT_LEVEL) 0xF071000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								IFIFO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								OFIFO							

IFIFO[7:0]	Description
n	Input FIFO interrupt level. (0 ~ 127)

OFIFO[7:0]	Description
n	Output FIFO interrupt level (0 ~ 127)

Trigger Mode Register(JP_TRG_MOD) 0xF0710010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
															TMD

TMD[0]	Description
n	0 : Polling mode. 1 : Interrupt mode.

Raw Data Buffer Address Register (R_Y/U/VBUF_A) 0xF07100(20/24/28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R_BUF_ADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R_BUF_ADDR[15:2]															

YBUF_ADDR[31:2]	Description
n	The raw buffer Y start address.

UBUF_ADDR[31:2]	Description
n	The raw buffer U start address.

VBUF_ADDR[31:2]	Description
n	The raw buffer V start address.

Raw Buffer Information Register (R_BUF_INFO) 0xF071002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				IFRM_HSIZE											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				IFRM_VSIZE											

FRM_HSIZE[11:0]	Description
n	Input buffer horizontal size (unit == pixel)

FRM_VSIZE[11:0]	Description
n	Input buffer vertical size (unit == line)

Image Size Register (IMG_SIZE) 0xF0710030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				IMG_HSIZE[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				IMG_VSIZE[11:0]											

IMG_HSIZE[11:0]	Description
n	Image horizontal size (unit == pixel)

IMG_VSIZE[11:0]	Description
n	Image vertical size (unit == line)

JPEG File Format Register(CHROMA) 0xF0710034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													CHROMA		

CHROMA[2:0]	Description
0	Y only (output format : Y only)
1	YUV420 (output format : YUV420)
2	YUV422 (output format : YUV422)
3	YUV444 (output format : YUV422)
4	YUV422S (output t format : YUV422)

Coded Buffer Address Register(CBUF_ADDR) 0xF0710038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CBUF_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CBUF_ADDR[15:2]														0	0

CBUF_ADDR[11:0]	Description
n	The coded data buffer start address.. In decoding mode : JPEG file body but header.

Coded Buffer Size Register (CBUF_SIZE) 0xF071003C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				CBUF_SIZE											

CBUF_SIZE[11:0]	Description
n	Coded data buffer size (unit = 1024bytes)

Decoder Table ID(JPD_TBL_ID) 0xF0710050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										Y_QT		U_QT		V_QT	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
										V_HT		U_HT		V_HT	

The JPEG decoder has three quantization tables. These register values indicate table number, each of color component(Y,U,V) use which table. “00” means table 0. “01” means table 1. “10” means table 2.

Y_QT[1:0]	Description
n	Quantization Table ID for Y.

U_QT[1:0]	Description
n	Quantization Table ID for U.

V_QT[1:0]	Description
n	Quantization Table ID for V.

Y_HT[1:0]	Description
n	Huffman Table ID for Y.

U_HT[1:0]	Description
n	Huffman Table ID for U.

V_HT[1:0]	Description
n	Huffman Table ID for V.

Decoder Restart Interval Register(JPD_RST_INTV) 0xF0710054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTV															

INTV[15:0]	Description
n	Decoder restart interval. If the JPEG header has Restart Interval, this register value is set by the Restart Interval of the JPEG header. Any other case, set zero.

Decoder Output Scale Register (JPD_OUT_SCL) 0xF0710058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														SCL	

SCL[1:0]	Description
n	Decoder output scale ratio. 0 : no scale 1 : 1/4 (area ratio) 2 : 1/16 (area ratio).

Block Start Register (JP_START) 0xF0710070

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															ST

ST	Description
1	Decoding start command.

Destination Buffer Write Counter (DBUF_WC) 0xF0710080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				WC											

RC[11:0]	Description
n	Destination buffer write count (unit = 1line)

Destination Buffer Read Counter (DBUF_RC) 0xF0710084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RC											

RC[11:0]	Description
n	Destination buffer read count (unit = 1line)

Source Buffer Write Counter (SBUF_WC) 0xF0710088

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														WC[17:16]	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WC[15:0]															

WC[17:0]	Description
n	Source buffer write count (unit = 16bytes)

Source Buffer Read Counter (SBUF_RC) 0xF071008C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				RC											

RC[11:0]	Description
n	Source buffer read count (unit = 1024bytes)

Input FIFO Status (IFIFO_ST) 0xF0710090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								FILL							

FILL[7:0]	Description
n	Output FIFO fill count

Output FIFO Status (OFIFO_ST) 0xF0710094

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								FILL							

FILL[7:0]	Description
n	Output FIFO fill count

Interrupt Flags (INT_FLAG) 0xF07100A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
										IFLAG[5:0]					

IFLAG[5:0]	Description
5	Raw data buffer status
4	Coded buffer status
3	Input FIFO status
2	Output FIFO status
1	Decoding error
0	Job finished

Interrupt ACK Register (INT_ACK) 0xF07100A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

When read this register, Interrupt Flags are cleared.

Input FIFO Write Data Register (IFIFO_WD) 0xF07100C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDATA[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WDATA[15:0]															

In slave mode (the JPEG decoding/encoding or DCT/IDCT mode), host feds data into this register. (maximum burst size == 8)

WDATA[31:0]	Description
n	The write data register in slave mode.

Output FIFO Read Data Register (OFIFO_RD) 0xF07100E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDATA[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RDATA[15:0]															

In slave mode (the JPEG decoding/encoding), host gets result data from this register. (maximum burst size == 8)

RDATA[31:0]	Description
n	The read data register in slave mode.

Decoder Inverse Q Table 0(DE_IQTAB0) **0xF0710300-0xF07103FC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				DATA											

Decoder Inverse Q Table 1(DE_IQTAB1) **0xF0710400-0xF07104FC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				DATA											

Decoder Inverse Q Table 2(DE_IQTAB2) **0xF0710500-0xF07105FC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				DATA											

DATA[11:0]	Description
n	Decoder inverse quantization data (64 entries) DE_IQTAB0/1/2(64 entries) is the de-quantization matrix table 0, zigzag scan ordered.

Decoder Huffman Table DC0 Code (DE_HT_DC0_C) **0xF0710600-xF071063C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE															

Decoder Huffman Table AC0 Code (DE_HT_AC0_C) **0xF0710640-0xF071067C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE															

Decoder Huffman Table DC1 Code (DE_HT_DC1_C) **0xF0710680-0xF07106BC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE															

Decoder Huffman Table AC1 Code (DE_HT_AC1_C) **0xF07106C0-0xF07106FC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE															

DATA[11:0]	Description
n	Huffman table code length. DE_HT_DC0_C values are DC Huffman table 0 code length. (LSB aligned)

Decoder Huffman Table DC0 Address (DE_HT_DC0_A) **0xF0710700-0xF071073C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								ADDR							

Decoder Huffman Table AC0 Address (DE_HT_AC0_A) **0xF0710740-0xF071077C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								ADDR							

Decoder Huffman Table DC1 Address (DE_HT_DC1_A) **0xF0710780-0xF07107BC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								ADDR							

Decoder Huffman Table AC1 Address (DE_HT_AC1_A) **0xF07107C0-F07107FC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								ADDR							

ADDR[7:0]	Description
n	Huffman table address DE_HT_DC0_A values are DC Huffman table 0 addresses of the JPEG decoder. (12 entries) DE_HT_AC0_A values are AC Huffman table 0 addresses of the JPEG decoder. (162 entries) DE_HT_DC1_A values are DC Huffman table 1 addresses of the JPEG decoder. (12 entries) DE_HT_AC1_A values are AC Huffman table 1 addresses of the JPEG decoder. (162 entries)

Decoder Huffman Table DC0 Value (DE_HT_DC0_V) **0x90006800-0x9000683C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								VAL							

Decoder Huffman Table AC0 Value (DE_HT_AC0_V) **0x90006840-0x90006AC4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								VAL							

Decoder Huffman Table DC1 Value (DE_HT_DC1_V) **0x90006C00-0x90006C3C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								VAL							

Decoder Huffman Table AC1 Value (DE_HT_AC1_V) **0x90006C40-0x90006EC4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								VAL							

VAL[7:0]	Description
n	Huffman table value DE_HT_DC0_A values are DC Huffman table 0 values of the JPEG decoder. (16 entries) DE_HT_AC0_A values are AC Huffman table 0 values of the JPEG decoder. (162 entries) DE_HT_DC1_A values are DC Huffman table 1 values of the JPEG decoder. (16 entries) DE_HT_AC1_A values are AC Huffman table 1 values of the JPEG decoder. (162 entries)

7 Video Bus Configuration

Power Down Mode Value **0xF0702000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
												val			

VAL[3:0]	Description
3	VIDEO CACHE power down 0 : Bus Clock for the Video Cache Clock is Enabled 1 : Bus Clock for the Video Cache Clock is Disabled
2	VIDEO codec power down 0 : Bus Clock for the Video Codec Clock is Enabled 1 : Bus Clock for the Video Codec Clock is Disabled ***The core clock for the video codec can be disabled in the CKC register.***
1	JPEG Decoder power down 0 : Bus Clock for the JPEG Decoder is Enabled 1 : Bus Clock for the JPEG Decoder is Disabled
0	JPEG Encoder power down 0 : Bus Clock for the JPEG Encoder is Enabled 1 : Bus Clock for the JPEG Encoder is Disabled

Soft Reset Register **0xF0702004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
												VAL			

VAL[3:0]	Description
3	VIDEO CACHE soft reset 0 : Not-Reset, 1 : Reset
2	VIDEO codec soft reset 0 : Not-Reset, 1 : Reset
1	JPEG Decoder soft reset 0 : Not-Reset, 1 : Reset
0	JPEG Encoder soft reset 0 : Not-Reset, 1 : Reset

PART8 – GRAPHIC BUS

TCC8900

High Performance and Low-Power Processor

For Digital Media Applications

Rev. 1.00

Aug 07, 2009

TelleChips

Revision History

Date	Revision	Description
2008-12-11	0.00	Initial release
2009-01-15	0.01	- Add figure about GPU idle configuration register and clock
2009-02-25	0.02	- Minor changes
2009-05-06	0.03	- Add Register Descriptions about GPU
2009-08-07	1.00	- Deleting Change Log

TABLE OF CONTENTS

Contents

1 Introduction 1-1
2 Bus Architecture 2-3
3 Address and Register Map 3-5
4 Overlay Mixer 4-7
4.1 Overview 4-7
4.2 Features 4-9
4.3 Register Description 4-12
4.4 Overlay Mixer Operation 4-45
4.4.1 Source Rotate / Mirror and Destination Rotate / Mirror Operation 4-45
4.4.2 YUV to RGB Format Conversion 4-46
4.4.3 Arithmetic Operation 4-47
4.4.4 Alpha-Blending and ROP with Window Offset and Chroma-Key 4-48
4.4.5 RGB to YUV Format Conversion 4-51
4.4.6 Destination RGB Dithering 4-52
5 2D/3D GPU 5-53
5.1 Overview 5-53
5.1.1 Pixel Processor Features 5-53
5.1.2 Geometry Processor Features 5-53
5.2 GPU Structure 5-54
5.3 Pixel Processor Register Description 5-55
5.4 Geometry Processor Register Description 5-57
5.5 MMU configuration Register Description 5-66
6 GRPBUS Configuration 6-67

Figures

Figure 2.1 The GRP Hardware Bus Architecture 2-3
Figure 4.1 Overlay Mixer Block Diagram 4-8
Figure 4.2 Supported Data Format of Overlay Mixer 4-10
Figure 4.3 Sampling of YUV Format 4-10
Figure 4.4 Detail Information of YUV Format 4-11
Figure 4.5 Source/Destination Base Address of Data Format 4-13
Figure 4.6 Example: Input Source Image Data to RGB888/ARGB8888 Conversion 4-16
Figure 4.7 (a) 16 Level 4x4 Dither Matrix (b) Example of Dither Matrix Setting 4-42
Figure 4.8 Source & Destination Mirror / Rotate Operation 4-45
Figure 4.9 Source YUV to RGB Conversion 4-46
Figure 4.10 Arithmetic Operation 4-47
Figure 4.11 ROP / Alpha-Blending Block Diagram 4-48
Figure 4.12 ROP and Alpha-Blending Operation Example 4-48
Figure 4.13 RGB to YUV Format Converter 4-51
Figure 4.14 Destination RGB Dithering 4-52
Figure 4.15 Destination RGB Dithering Operation 4-52
Figure 5.1 2D/3D Graphics System 5-54
Figure 5.2 Pixel Processor Register Map 5-55
Figure 5.3 Geometry Processor Configuration Register Map 5-57
Figure 6.1 GPU Idle Configuration Register and Clock 6-70

Tables

Table 4.1 Overlay Mixer Register Map (Base Address = 0xF0010000) 4-12
Table 5.1 Pixel Processor Register Map (Base Address = 0xF0000000) 5-56
Table 5.2 Geometry Processor Control Register Map (Base Address = 0xF0000000) 5-58
Table 5.3 Geometry Processor PLB Configuration Register Map 5-58
Table 5.4 Geometry Processor Vertex Shader Register Map 5-58
Table 5.5 VS_CONF_REG_INP_SPEC Vertex Data Format 5-64
Table 5.6 VS_CONF_REG_OUTP_SPEC Vertex Data Format 5-65
Table 5.7 MMU Configuration Register Map (Base Address = 0xF0000000) 5-66
Table 6.1 GRPBUS Configuration Register Map (Base Address = 0xF0004000) 6-67
Table 6.2 GRPBUS BWRAP Register Map (Base Address = 0xF0005000) 6-67

1 Introduction

TCC8900 GRPBUS provides connections between the internal graphics bus masters and the other buses. Also on chip peripherals are located in it. The graphics bus masters supported is as follows : 2D/3D graphics rendering, various operations for raster graphics.

[Features]

- Overlay Mixer
  - Various Input / Output Image Formats
  - 90 / 180 / 270 Rotate
  - Vertical / Horizontal / Vertical & Horizontal Mirror
  - Arithmetic : Bypass, Fill, Inversion, Addition, Substract, Multiplication
  - BitBlt (Raster Operation)
  - Alpha-Blending
  - Overlay
  - YUV to RGB Format Converting
  - RGB to YUV Format Converting
  - Color LUT
  - Dithering
- 2D/3D GPU (Graphics Processing Unit)
  - Pixel Processor
    - Programmable fragment shader
    - Access to framebuffer from fragment shaders
    - Alpha blending
    - Arbitrary memory reads and writes
    - Complete non-power-of-2 texture support
    - Cube mapping
    - Dynamic recursion
    - Fast dynamic branching
    - Fast trigonometric functions, including arctangent
    - Full floating-point arithmetic
    - Framebuffer blend with destination Alpha
    - High Dynamic Range (HDR) textures and framebuffers
    - Indexable texture samplers
    - Line, quad, triangle and point sprites
    - Multiple render targets
    - No limit on program length
    - Perspective Anisotropic Filtering (AF)
    - Perspective correct texturing
    - Point sampling, bilinear, and trilinear filtering
    - Programmable mipmap level-of-detail biasing and replacement
    - Register indirect jumps
    - Stencil buffering, 8-bit
    - Two-sided stencil
    - Unlimited dependent texture reads
    - Virtualized texture samplers
    - 4-level hierarchical Z and stencil operations
    - 4 times and 16 times Full Scene Anti-Aliasing (FSAA)
    - 4-bit per texel texture compression
  - Geometry Processor
    - Programmable vertex shader
    - Autonomous operation tile list generation
    - Flexible input and output formats
    - Indexed and non-indexed geometry input
    - Primitive constructions with points, lines, triangles and quads.

2 Bus Architecture

Figure 2.1 shows the GRP bus overall architecture.

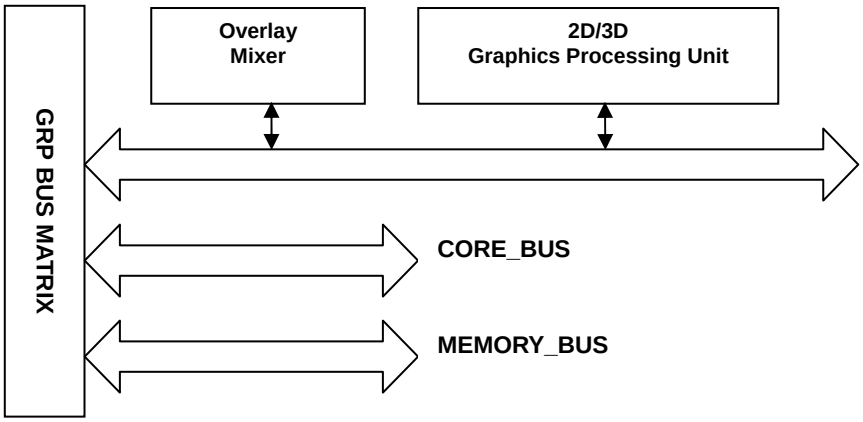


Figure 2.1 The GRP Hardware Bus Architecture

3 Address and Register Map

The TCC8900 has 2D/3D GPU and overlay mixer peripherals. These peripherals can be configured appropriately by its own registers that can be accessed through specially allocated address. These address maps are represented in the following table.

Refer to corresponding sections for detail information of each peripheral.

Base Address		Peripherals
0xF0000000		2D/3D GPU
0xF0010000		Overlay Mixer

4 Overlay Mixer

4.1 Overview

Figure 4.1 shows the whole block diagram of Overlay Mixer and its corresponding registers

As can be seen below, Overlay Mixer supports Rotating / Mirroring image, bitwise ROP, Alpha-Blending, Arithmetic or Format Converting function. These functions can simultaneously work. Overlay Mixer has two different source channels and one destination channel. In other words, one output image can be created with two Input images by using an appropriate function of Overlay Mixer. Moreover, specific local region operation is available since Overlay Mixer supports Source Image Offset, Destination Image Offset and Window Offset.

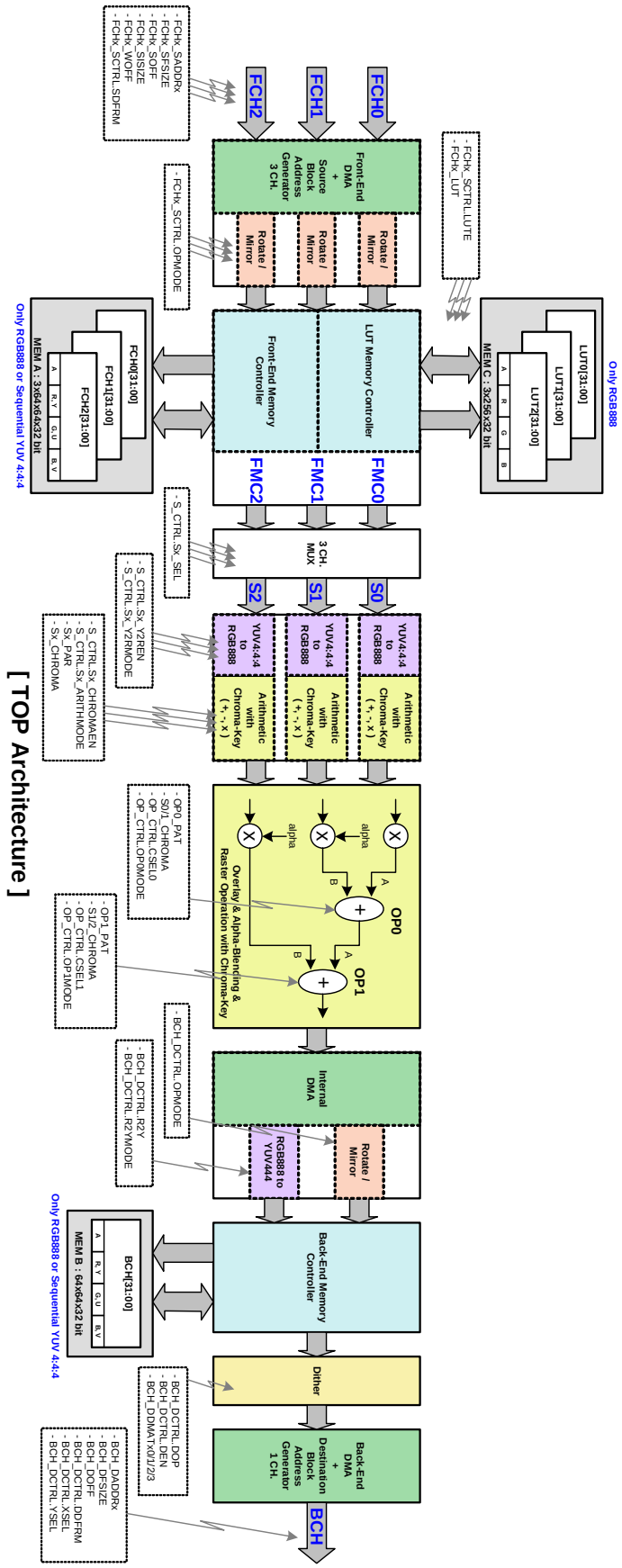


Figure 4.1 Overlay Mixer Block Diagram

4.2 Features

The features of Overlay Mixer are as follows

- Input / Output Image Format
  - Indexed Color Format : 8bpp
  - RGB Format : RGB332, RGB444, RGB454, RGB555, RGB565, RGB666, RGB888
  - Alpha-RGB(ARGB) Format : ARGB4444, ARGB3454, ARGB1555, ARGB4666, ARGB6666, ARGB4888, ARGB8888
  - Sequential YUV : YUV4:4:4, YUV4:2:2
  - Separated YUV : YUV4:4:4, YUV4:4:0, YUV4:2:2, YUV4:2:0, YUV4:1:1, YUV4:1:0
  - Interleaved YUV : YUV4:2:2, YUV4:2:0
- Rotate/Mirror Operation
  - 2 Channel Source 90/180/270 Rotate with Window Offset and Source Offset
  - 1 Channel Destination 90/180/270 Rotate with Destination Offset
  - 2 Channel Source Vertical/Horizontal/Vertical & Horizontal Mirror with Window Offset and Source Offset
  - 1 Channel Destination Vertical/Horizontal/Vertical & Horizontal Mirror with Destination Offset
- Arithmetic Operation
  - 2 Channel Arithmetic Operation with Chroma-Key Function
    - Arithmetic Function : Bypass, Fill, Inversion, Addition, Substract, Multiplication
- Raster Operation & Alpha-Blending & Overlay
  - 2 Channel to 1Channel ROP Operation with Chroma-Key & Window Offset
  - 2 Channel to 1Channel Alpha Blending with Chroma-Key & Window Offset
  - ROP Function : 16 Type
  - Alpha-Blending Operation by Fixed Alpha-Value
  - Alpha-Blending Operation by ARGB Input Image Format
- YUV to RGB Format Converting
  - 2 Independent Source YUV to RGB Format Converting
  - 4 YUV to RGB Format Converting Type
- RGB to YUV Format Converting
  - 1 Destination RGB to YUV Format Converting
  - 4 RGB to YUV Format Converting Type
- Color LUT
  - 3 Channel Source Indexed Color with LUT : 8bpp
  - 3 Channel Source Alpha with LUT : ARGB4444, ARGB3454, ARGB1555, ARGB4666, ARGB6666, ARGB4888, ARGB8888
- RGB Format Dithering
  - 1 Channel Destination Dithering : RGB332, RGB444, ARGB4444, RGB454, ARGB3454, RGB555, ARGB1555, RGB565, RGB666, ARGB4666, ARGB6666, ARGB4888

		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Indexed Color Format	8bpp	Data3[7:0]								Data2[7:0]								Data1[7:0]								Data0[7:0]							
RGB Format	RGB332(8bit)	R3[2:0]			G3[2:0]			B3[1:0]		R2[2:0]			G2[2:0]			B2[1:0]		R1[2:0]			G1[2:0]			B1[1:0]		R0[2:0]			G0[2:0]			B0[1:0]	
	RGB444(12bit)	PAD			R1[3:0]					G1[3:0]				B1[3:0]				PAD			R0[3:0]					G0[3:0]				B0[3:0]			
	ARGB4444(16bit)	A1[3:0]			R1[3:0]					G1[3:0]				B1[3:0]				A0[3:0]			R0[3:0]					G0[3:0]				B0[3:0]			
	RGB454(13bit)	PAD			R1[3:0]					G1[4:0]					B1[3:0]			PAD			R0[3:0]					G0[4:0]					B0[3:0]		
	ARGB3454(16bit)	A1[2:0]			R1[3:0]					G1[4:0]					B1[3:0]			A0[2:0]			R0[3:0]					G0[4:0]					B0[3:0]		
	RGB555(15bit)	P			R1[4:0]					G1[4:0]					B1[4:0]			P			R0[4:0]					G0[4:0]					B0[4:0]		
	ARGB1555(16bit)	A1			R1[4:0]					G1[4:0]					B1[4:0]			A0			R0[4:0]					G0[4:0]					B0[4:0]		
	RGB565(16bit)	R1[4:0]						G1[5:0]							B1[4:0]			R0[4:0]						G0[5:0]							B0[4:0]		
	RGB666(18bit)	PAD														R[5:0]							G[5:0]								B[5:0]		
	ARGB4666(22bit)	PAD											A[3:0]			R[5:0]							G[5:0]								B[5:0]		
	ARGB6666(24bit)	PAD								A[5:0]						R[5:0]							G[5:0]								B[5:0]		
	RGB888(24bit)	PAD								R[7:0]								G[7:0]								B[7:0]							
	ARGB4888(28bit)	PAD			A[3:0]					R[7:0]								G[7:0]								B[7:0]							
	ARGB8888(32bit)	A[7:0]								R[7:0]								G[7:0]								B[7:0]							
Sequential YUV Format	Sequential YUV 4:4:4	PAD								Y[7:0]								U[7:0]								V[7:0]							
	Sequential YUV 4:2:2	V0[7:0]								Y1[7:0]								U0[7:0]								Y0[7:0]							
Separated YUV Format	Separated YUV 4:4:4																																
	Separated YUV 4:4:0	Y[7:0]								Y[7:0]								Y[7:0]								Y[7:0]							
	Separated YUV 4:2:2	U[7:0]								U[7:0]								U[7:0]								U[7:0]							
	Separated YUV 4:2:0	V[7:0]								V[7:0]								V[7:0]								V[7:0]							
	Separated YUV 4:1:1																																
	Separated YUV 4:1:0																																
Interleaved YUV Format	Interleaved YUV 4:2:2	Y[7:0]								Y[7:0]								Y[7:0]								Y[7:0]							
	Interleaved YUV 4:2:0	V[7:0]								U[7:0]								V[7:0]								U[7:0]							
	Interleaved YUV 4:2:2 (Swap U and V)	Y[7:0]								Y[7:0]								Y[7:0]								Y[7:0]							
	Interleaved YUV 4:2:0 (Swap U and V)	U[7:0]								V[7:0]								U[7:0]								V[7:0]							

* Refer to Figure 4.4 for detail information of YUV Format

Figure 4.2 Supported Data Format of Overlay Mixer

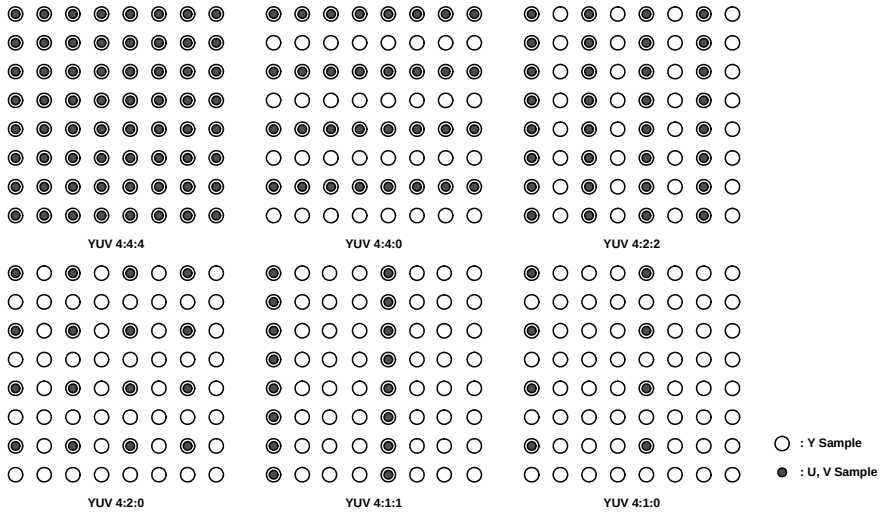


Figure 4.3 Sampling of YUV Format

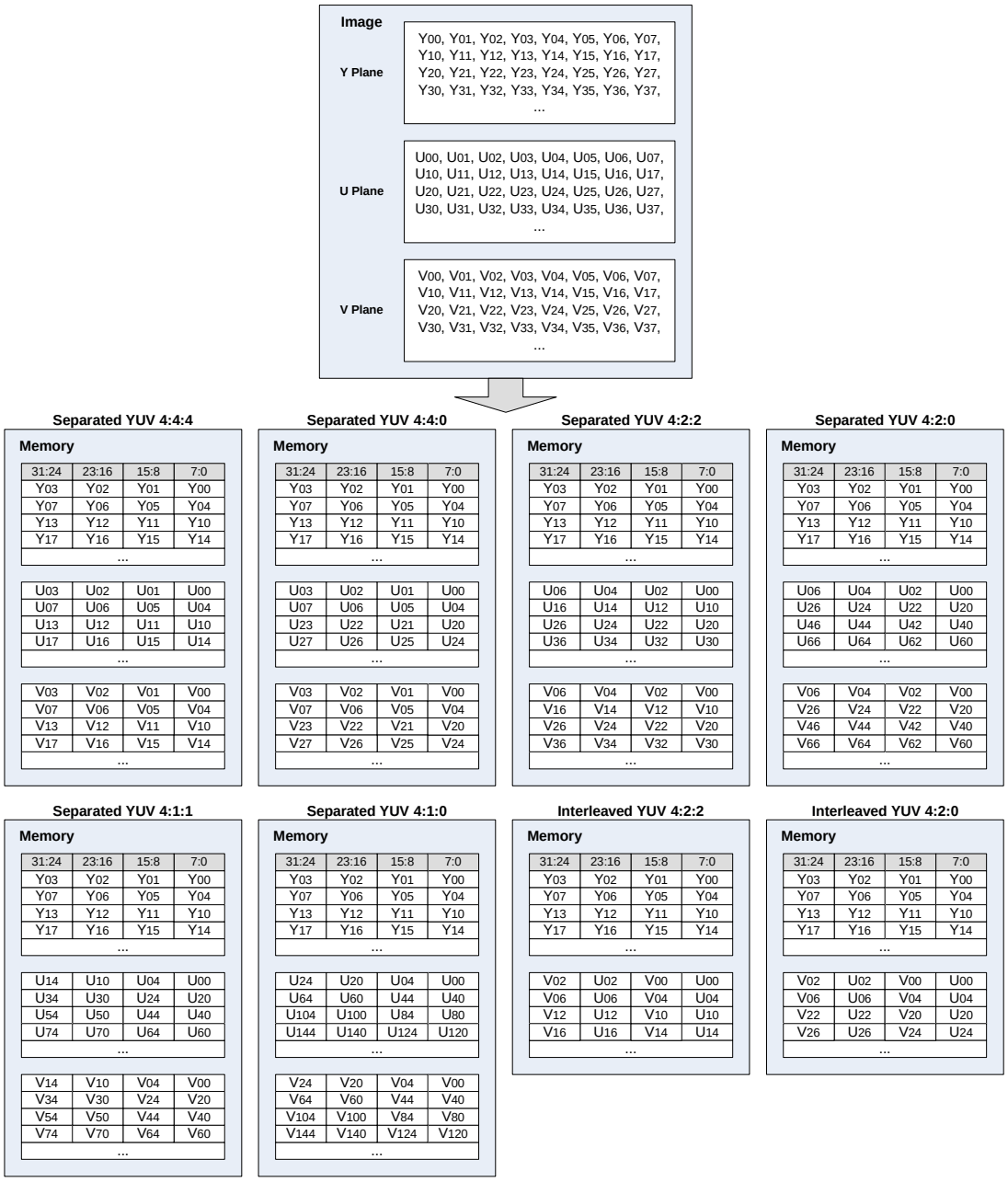


Figure 4.4 Detail Information of YUV Format

4.3 Register Description

Table 4.1 Overlay Mixer Register Map (Base Address = 0xF0010000)

Name	Address	Type	Reset	Description
FCH0_SADDR0	0x00	R/W	0x00000000	Front-End Channel 0 Source Address 0
FCH0_SADDR1	0x04	R/W	0x00000000	Front-End Channel 0 Source Address 1
FCH0_SADDR2	0x08	R/W	0x00000000	Front-End Channel 0 Source Address 2
FCH0_SFSIZE	0x0C	R/W	0x00000000	Front-End Channel 0 Source Frame Pixel Size
FCH0_SOFF	0x10	R/W	0x00000000	Front-End Channel 0 Source Pixel Offset
FCH0_SISIZE	0x14	R/W	0x00000000	Front-End Channel 0 Source Image Pixel Size
FCH0_WOFF	0x18	R/W	0x00000000	Front-End Channel 0 Window Pixel Offset
FCH0_SCTRL	0x1C	R/W	0x00000000	Front-End Channel 0 Control
FCH1_SADDR0	0x20	R/W	0x00000000	Front-End Channel 1 Source Address 0
FCH1_SADDR1	0x24	R/W	0x00000000	Front-End Channel 1 Source Address 1
FCH1_SADDR2	0x28	R/W	0x00000000	Front-End Channel 1 Source Address 2
FCH1_SFSIZE	0x2C	R/W	0x00000000	Front-End Channel 1 Source Frame Pixel Size
FCH1_SOFF	0x30	R/W	0x00000000	Front-End Channel 1 Source Pixel Offset
FCH1_SISIZE	0x34	R/W	0x00000000	Front-End Channel 1 Source Image Pixel Size
FCH1_WOFF	0x38	R/W	0x00000000	Front-End Channel 1 Window Pixel Offset
FCH1_SCTRL	0x3C	R/W	0x00000000	Front-End Channel 1 Control
FCH2_SADDR0	0x40	R/W	0x00000000	Front-End Channel 2 Source Address 0
FCH2_SADDR1	0x44	R/W	0x00000000	Front-End Channel 2 Source Address 1
FCH2_SADDR2	0x48	R/W	0x00000000	Front-End Channel 2 Source Address 2
FCH2_SFSIZE	0x4C	R/W	0x00000000	Front-End Channel 2 Source Frame Pixel Size
FCH2_SOFF	0x50	R/W	0x00000000	Front-End Channel 2 Source Pixel Offset
FCH2_SISIZE	0x54	R/W	0x00000000	Front-End Channel 2 Source Image Pixel Size
FCH2_WOFF	0x58	R/W	0x00000000	Front-End Channel 2 Window Pixel Offset
FCH2_SCTRL	0x5C	R/W	0x00000000	Front-End Channel 2 Control
S0_CHROMA	0x60	R/W	0x00000000	Source 0 Chroma-Key Parameter
S0_PAR	0x64	R/W	0x00000000	Source 0 Arithmetic Parameter
S1_CHROMA	0x68	R/W	0x00000000	Source 1 Chroma-Key Parameter
S1_PAR	0x6C	R/W	0x00000000	Source 1 Arithmetic Parameter
S2_CHROMA	0x70	R/W	0x00000000	Source 2 Chroma-Key Parameter
S2_PAR	0x74	R/W	0x00000000	Source 2 Arithmetic Parameter
S_CTRL	0x78	R/W	0x00000000	Source Control Register
-	0x7C	-	-	Reserved
OP0_PAT	0x80	R/W	0x00000000	Source Operator 0 Pattern
OP1_PAT	0x84	R/W	0x00000000	Source Operator 1 Pattern
OP_CTRL	0x88	R/W	0x00000000	Source Operation Control Register
-	0x8C	-	-	Reserved
BCH_DADDR0	0x90	R/W	0x00000000	Back-End Channel Destination Address 0
BCH_DADDR1	0x94	R/W	0x00000000	Back -End Channel Destination Address 1
BCH_DADDR2	0x98	R/W	0x00000000	Back -End Channel Destination Address 2
BCH_DFSIZE	0x9C	R/W	0x00000000	Back -End Channel Destination Frame Pixel Size
BCH_DOFF	0xA0	R/W	0x00000000	Back -End Channel Destination Pixel Offset
BCH_DCTRL	0xA4	R/W	0x00000000	Back -End Channel Control
-	0xA8 – 0xAF	-	-	Reserved
BCH_DDMAT0	0xB0	R/W	0x00000000	Back-End Channel Destination Dither Matrix 0
BCH_DDMAT1	0xB4	R/W	0x00000000	Back-End Channel Destination Dither Matrix 1
BCH_DDMAT2	0xB8	R/W	0x00000000	Back-End Channel Destination Dither Matrix 2
BCH_DDMAT3	0xBC	R/W	0x00000000	Back-End Channel Destination Dither Matrix 3
OM_CTRL	0xC0	R/W	0x00000000	Overlay Mixer Control
OM_IREQ	0xC4	R/W	0x00000000	Overlay Mixer Interrupt Request
-	0xC8 – 0x3FF	-	-	Reserved
FCH0_LUT	0x400 – 0x7FF	R/W	-	Front-End Channel 0 Lookup Table
FCH1_LUT	0x800 – 0xBFF	R/W	-	Front-End Channel 1 Lookup Table
FCH2_LUT	0xC00 – 0xFFF	R/W	-	Front-End Channel 2 Lookup Table

Front-End Channel 0 Source Address 0(FCH0_SADDR0) 0xF0010000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR0[15:2]														Reserved	

SADDR0	[31:2]	Source Address 0
N (0 ~ 4095)	It determines the Y image base address of channel 0 image when the data format of channel 0 is YUV format. But, in case the data format of channel 0 is a sequential YUV or an RGB, it determines the base address of the whole image	

The data format of FCH0 image is determined by the FCH0_SCTRL register on page 4-15.

Front-End Channel 0 Source Address 1 (FCH0_SADDR1) 0xF0010004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR1[15:2]														Reserved	

SADDR1	[31:2]	Source Address 1
N (0 ~ 4095)	It determines the U image base address of front-end channel 0 (FCH0) when the data format of channel 0 is YUV format. But, in case the data format of FCH0 image is a sequential YUV or an RGB, it is not used.	

Front-End Channel 0 Source Address 2 (FCH0_SADDR2) 0xF0010008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR2[15:2]														Reserved	

SADDR2	[31:2]	Source Address 2
N (0 ~ 4095)	It determines the V image base address of FCH0 when the data format of FCH0 is YUV format. But, in case the data format of FCH0 image is a sequential YUV or an RGB, it is not used.	

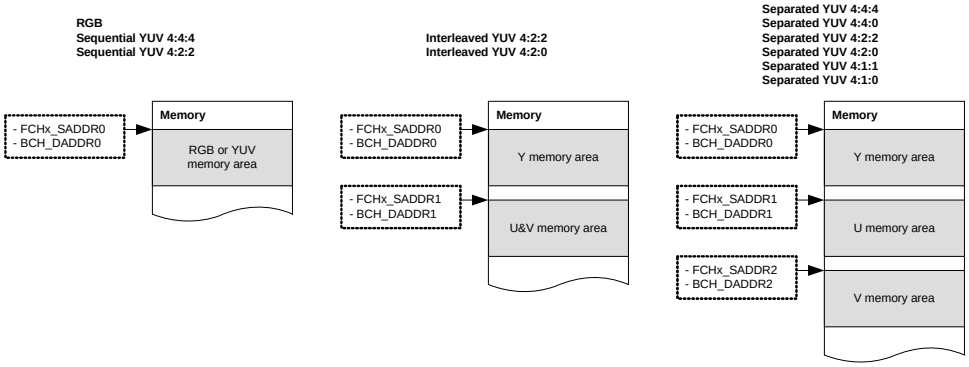


Figure 4.5 Source/Destination Base Address of Data Format

Front-End Channel 0 Source Frame Pixel Size (FCH0_SFSIZE) 0xF001000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SFSIZE_Y[11:0]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				SFSIZE_X[11:0]											

SFSIZE_Y	[27:16]	Source Frame Pixel Size Y
N (0 ~ 4095)	height of FCH0 image by pixel units	

SFSIZE_X	[11:0]	Source Frame Pixel Size X
N (0 ~ 4095)	width of FCH0 image by pixel units	

Front-End Channel 0 Source Pixel Offset (FCH0_SOFF) 0xF0010010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SOFF_Y[11:0]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				SOFF_X[11:0]											

SOFF_Y	[27:16]	Source Pixel Offset Y
N (0 ~ 4095)	N =Y Axis Source Pixel Offset	

SOFF_X	[11:0]	Source Pixel Offset X
N (0 ~ 4095)	N =X Axis Source Pixel Offset	

Front-End Channel 0 Source Image Pixel Size (FCH0_SISIZE) 0xF0010014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SISIZE_Y[11:0]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				SISIZE_X[11:0]											

SISIZE_Y	[27:16]	Source Image Pixel Size Y
N (0 ~ 4095)	N =Y Axis Source Image Pixel Size	

SISIZE_X	[11:0]	Source Image Pixel Size X
N (0 ~ 4095)	N =X Axis Source Image Pixel Size	

Front-End Channel 0 Window Pixel Offset (FCH0_WOFF) 0xF0010018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				WOFF_Y[11:0]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				WOFF_X[11:0]											

WOFF_Y	[27:16]	Window Pixel Offset Y
N (0 ~ 4095)	N =Y Axis Window Pixel Offset	

WOFF_X	[11:0]	Window Pixel Offset X
N (0 ~ 4095)	N =X Axis Window Pixel Offset	

When FCH0 Image is selected as Source 0 by SCTRL.S0SEL register, Window Offset should definitely be ‘0’.

Front-End Channel 0 Control (FCH0_SCTRL) 0xF001001C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MABC	Reserved		LUTE	SSUV	OPMODE[2:0]			0	ZF[1:0]		SDFRM[4:0]				

MABC	[15]	AXI Bus Master Address Boundary Control
0	AXI Bus Master Address Boundary = 4KByte	
1	AXI Bus Master Address Boundary = 1KByte	

LUTE	[12]	LUT Enable
0	LUT Disable	
1	LUT Enable	

SSUV	[11]	Source Swap U for V
0	Keep on U and V (LSB = U, MSB = V)	
1	Swap U for V (LSB = V, MSB = U)	

This register applies only when Data Format is Interleaved YUV Format

OPMODE	[10:8]	Operation Mode
00x	Data Copy	
010	Horizontal Mirror	
011	Vertical Mirror	
100	Vertical & Horizontal Mirror	
101	90 Degree Rotate (Clockwise)	
110	180 Degree Rotate (Clockwise)	
111	270 Degree Rotate (Clockwise)	

ZF	[6:5]	Zero Fill
00	MSB Fill Mode Enable	
01	Zero Fill Mode Enable	
1x	HOB(High-Order Bits) Fill Mode Enable	

The above applies only when Data Format is RGB Format, not RGB888 / ARGB8888.

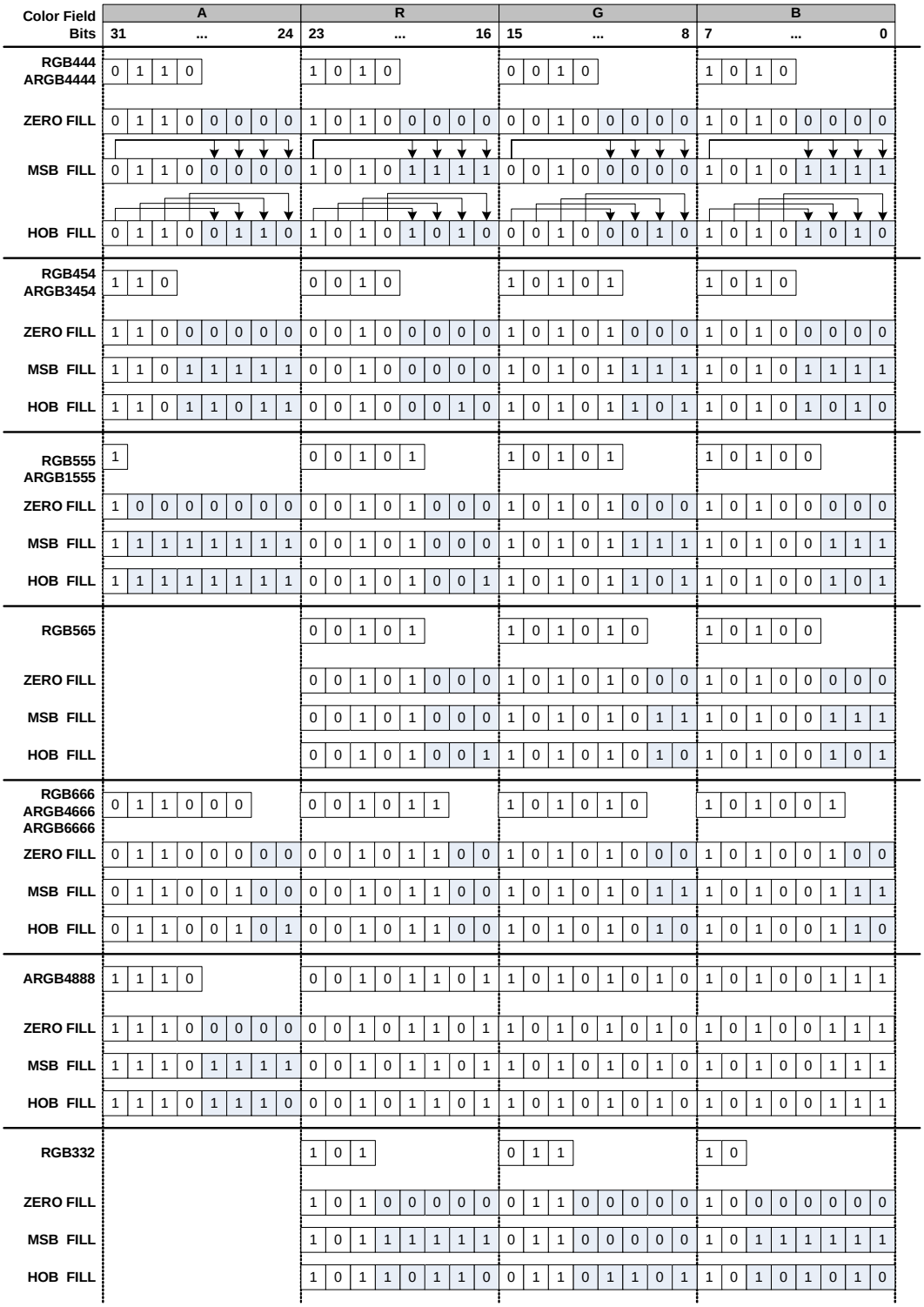


Figure 4.6 Example: Input Source Image Data to RGB888/ARGB8888 Conversion

SDFRM	[4:0]	Source Data Format
00000	Separated YUV 4:4:4	
00001	Separated YUV 4:4:0	
00010	Separated YUV 4:2:2	
00011	Separated YUV 4:2:0	
00100	Separated YUV 4:1:1	
00101	Separated YUV 4:1:0	
00110	Interleaved YUV 4:2:2	
00111	Interleaved YUV 4:2:0	
01000	Sequential YUV 4:4:4	
01001	Sequential YUV 4:2:2	
0101x	RGB332 (8bpp)	
01100	RGB444	
01101	ARGB4444	
01110	RGB454	
01111	ARGB3454	
10000	RGB555	
10001	ARGB1555	
1001x	RGB565	
10100	RGB666	
10101	ARGB4666	
10110	ARGB6666	
10111	RGB888	
11000	ARGB4888	
else	ARGB8888	

Front-End Channel 1 Source Address 0 (FCH1_SADDR0) 0xF0010020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR0[15:2]														Reserved	

Source Image Format is Separated YUV X:X:X, the value of the address is Source Address of Y region.
Source Image Format is not Separated YUV X:X:X, the value of the address is Source Address.

Front-End Channel 1 Source Address 1 (FCH1_SADDR1) 0xF0010024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR1[15:2]														Reserved	

In case Source Image Format is Separated YUV X:X:X, the value of the address is Source Address of U region.
In case Source Image Format is not Separated YUV X:X:X, the value of the address is not used.

Front-End Channel 1 Source Address 2 (FCH1_SADDR2) 0xF0010028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR2[15:2]														Reserved	

In case Source Image Format is Separated YUV X:X:X, the value of the address is Source Address of V region.
In case Source Image Format is not Separated YUV X:X:X, the value of the address is not used.

Front-End Channel 1 Source Frame Pixel Size (FCH1_SFSIZE) 0xF001002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SFSIZE_Y[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SFSIZE_X[11:0]											

SFSIZE_Y	[27:16]	Source Frame Pixel Size Y
N (0 ~ 4095)	N =Y Axis Source Frame Pixel Size	

SFSIZE_X	[11:0]	Source Frame Pixel Size X
N (0 ~ 4095)	N =X Axis Source Frame Pixel Size	

Front-End Channel 1 Source Pixel Offset (FCH1_SOFF) 0xF0010030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SOFF_Y[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SOFF_X[11:0]											

SOFF_Y	[27:16]	Source Pixel Offset Y
N (0 ~ 4095)	N =Y Axis Source Pixel Offset	

SOFF_X	[11:0]	Source Pixel Offset X
N (0 ~ 4095)	N =X Axis Source Pixel Offset	

Front-End Channel 1 Source Image Pixel Size (FCH1_SISIZE) 0xF0010034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SISIZE_Y[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SISIZE_X[11:0]											

SISIZE_Y	[27:16]	Source Image Pixel Size Y
N (0 ~ 4095)	N =Y Axis Source Image Pixel Size	

SISIZE_X	[11:0]	Source Image Pixel Size X
N (0 ~ 4095)	N =X Axis Source Image Pixel Size	

Front-End Channel 1 Window Pixel Offset (FCH1_WOFF) 0xF0010038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				WOFF_Y[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WOFF_X[11:0]											

WOFF_Y	[27:16]	Window Pixel Offset Y
N (0 ~ 4095)	N =Y Axis Window Pixel Offset	

WOFF_X	[11:0]	Window Pixel Offset X
N (0 ~ 4095)	N =X Axis Window Pixel Offset	

When FCH1 Image is selected as Source 0 by SCTRL.S0SEL register, Window Offset should definitely be ‘0’.

Front-End Channel 1 Control (FCH1_SCTRL) 0xF001003C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MABC	Reserved		LUTE	SSUV	OPMODE[2:0]			0	ZF[1:0]		SDFRM[4:0]				

MABC	[15]	AXI Bus Master Address Boundary Control
0	AXI Bus Master Address Boundary = 4KByte	
1	AXI Bus Master Address Boundary = 1KByte	

LUTE	[12]	LUT Enable
0	LUT Disable	
1	LUT Enable	

SSUV	[11]	Source Swap U for V
0	Keep on U and V (LSB = U, MSB = V)	
1	Swap U for V (LSB = V, MSB = U)	

This register applies only when Data Format is Interleaved YUV Format

OPMODE	[10:8]	Operation Mode
00x	Data Copy	
010	Horizontal Mirror	
011	Vertical Mirror	
100	Vertical & Horizontal Mirror	
101	90 Degree Rotate (Clockwise)	
110	180 Degree Rotate (Clockwise)	
111	270 Degree Rotate (Clockwise)	

ZF	[6:5]	Zero Fill
00	MSB Fill Mode Enable	
01	Zero Fill Mode Enable	
1x	HOB(High-Order Bits) Fill Mode Enable	

The above applies only when Data Format is RGB Format, not RGB888 / ARGB8888. Refer to Figure 4.6 for detail information.

SDFRM	[4:0]	Source Data Format
00000	Separated YUV 4:4:4	
00001	Separated YUV 4:4:0	
00010	Separated YUV 4:2:2	
00011	Separated YUV 4:2:0	
00100	Separated YUV 4:1:1	
00101	Separated YUV 4:1:0	
00110	Interleaved YUV 4:2:2	
00111	Interleaved YUV 4:2:0	
01000	Sequential YUV 4:4:4	
01001	Sequential YUV 4:2:2	
0101x	RGB332 (8bpp)	
01100	RGB444	
01101	ARGB4444	
01110	RGB454	
01111	ARGB3454	
10000	RGB555	
10001	ARGB1555	
1001x	RGB565	
10100	RGB666	
10101	ARGB4666	
10110	ARGB6666	
10111	RGB888	
11000	ARGB4888	
else	ARGB8888	

Front-End Channel 2 Source Address 0 (FCH2_SADDR0) 0xF0010040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR0[15:2]														Reserved	

Source Image Format is Separated YUV X:X:X, the value of the address is Source Address of Y region.
Source Image Format is not Separated YUV X:X:X, the value of the address is Source Address.

Front-End Channel 2 Source Address 1 (FCH2_SADDR1) 0xF0010044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR1[15:2]														Reserved	

In case Source Image Format is Separated YUV X:X:X, the value of the address is Source Address of U region.
In case Source Image Format is not Separated YUV X:X:X, the value of the address is not used.

Front-End Channel 2 Source Address 2 (FCH2_SADDR2) 0xF0010048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR2[15:2]														Reserved	

In case Source Image Format is Separated YUV X:X:X, the value of the address is Source Address of V region.
In case Source Image Format is not Separated YUV X:X:X, the value of the address is not used.

Front-End Channel 2 Source Frame Pixel Size (FCH2_SFSIZE) 0xF001004C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SFSIZE_Y[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SFSIZE_X[11:0]											

SFSIZE_Y	[27:16]	Source Frame Pixel Size Y
N (0 ~ 4095)	N =Y Axis Source Frame Pixel Size	

SFSIZE_X	[11:0]	Source Frame Pixel Size X
N (0 ~ 4095)	N =X Axis Source Frame Pixel Size	

Front-End Channel 2 Source Pixel Offset (FCH2_SOFF) 0xF0010050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SOFF_Y[11:0]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				SOFF_X[11:0]											

SOFF_Y	[27:16]	Source Pixel Offset Y
N (0 ~ 4095)	N =Y Axis Source Pixel Offset	

SOFF_X	[11:0]	Source Pixel Offset X
N (0 ~ 4095)	N =X Axis Source Pixel Offset	

Front-End Channel 2 Source Image Pixel Size (FCH2_SISIZE) 0xF0010054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SISIZE_Y[11:0]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				SISIZE_X[11:0]											

SISIZE_Y	[27:16]	Source Image Pixel Size Y
N (0 ~ 4095)	N =Y Axis Source Image Pixel Size	

SISIZE_X	[11:0]	Source Image Pixel Size X
N (0 ~ 4095)	N =X Axis Source Image Pixel Size	

Front-End Channel 2 Window Pixel Offset (FCH2_WOFF) 0xF0010058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				WOFF_Y[11:0]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				WOFF_X[11:0]											

WOFF_Y	[27:16]	Window Pixel Offset Y
N (0 ~ 4095)	N =Y Axis Window Pixel Offset	

WOFF_X	[11:0]	Window Pixel Offset X
N (0 ~ 4095)	N =X Axis Window Pixel Offset	

When FCH1 Image is selected as Source 0 by SCTRL.S0SEL register, Window Offset should definitely be ‘0’.

Front-End Channel 2 Control (FCH2_SCTRL) 0xF001005C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MABC	Reserved		LUTE	SSUV	OPMODE[2:0]			0	ZF[1:0]		SDFRM[4:0]				

MABC	[15]	AXI Bus Master Address Boundary Control
0	AXI Bus Master Address Boundary = 4KByte	
1	AXI Bus Master Address Boundary = 1KByte	

LUTE	[12]	LUT Enable
0	LUT Disable	
1	LUT Enable	

SSUV	[11]	Source Swap U for V
0	Keep on U and V (LSB = U, MSB = V)	
1	Swap U for V (LSB = V, MSB = U)	

This register applies only when Data Format is Interleaved YUV Format

OPMODE	[10:8]	Operation Mode
00x	Data Copy	
010	Horizontal Mirror	
011	Vertical Mirror	
100	Vertical & Horizontal Mirror	
101	90 Degree Rotate (Clockwise)	
110	180 Degree Rotate (Clockwise)	
111	270 Degree Rotate (Clockwise)	

ZF	[6:5]	Zero Fill
00	MSB Fill Mode Enable	
01	Zero Fill Mode Enable	
1x	HOB(High-Order Bits) Fill Mode Enable	

The above applies only when Data Format is RGB Format, not RGB888 / ARGB8888. Refer to Figure 4.6 for detail information.

SDFRM	[4:0]	Source Data Format
00000	Separated YUV 4:4:4	
00001	Separated YUV 4:4:0	
00010	Separated YUV 4:2:2	
00011	Separated YUV 4:2:0	
00100	Separated YUV 4:1:1	
00101	Separated YUV 4:1:0	
00110	Interleaved YUV 4:2:2	
00111	Interleaved YUV 4:2:0	
01000	Sequential YUV 4:4:4	
01001	Sequential YUV 4:2:2	
0101x	RGB332 (8bpp)	
01100	RGB444	
01101	ARGB4444	
01110	RGB454	
01111	ARGB3454	
10000	RGB555	
10001	ARGB1555	
1001x	RGB565	
10100	RGB666	
10101	ARGB4666	
10110	ARGB6666	
10111	RGB888	
11000	ARGB4888	
else	ARGB8888	

Source 0 Chroma-Key Parameter (S0_CHROMA) 0xF0010060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CHROMA_RY[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHROMA_GU[7:0]								CHROMA_BV[7:0]							

CHROMA_RY	[23:16]	Chroma-Key Value
n	R or Y Chroma-Key Value	

CHROMA_GU	[15:8]	Chroma-Key Value
n	G or U Chroma-Key Value	

CHROMA_BV	[7:0]	Chroma-Key Value
n	B or V Chroma-Key Value	

Chroma-Key Value should be set to Sequential YUV4:4:4 or RGB888 irrespective of Input Source Format.

Writing should be done in RGB888 Format when Input Format is RGBxxx. When Input Format is YUVxxx and YUVtoRGB Converter is disabled, YUV444 Format is used for Writing. If input format is YUVxxx and YUVtoRGB Converter is enabled, writing is done in RGB888 Format.

S0_CHROMA Register is effective only when S_CTRL.S0_CHROMAEN is enabled and OP_CTRL.C0/1_SEL Register is 2bit “01”.

Source 0 Arithmetic Parameter (S0_PAR) 0xF0010064

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								PAR_RY[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PAR_GU[7:0]								PAR_BV[7:0]							

PAR_RY	[23:16]	Parameter Value for Arithmetic Operation
n	R or Y Parameter Value for Arithmetic Operation.	

PAR_GU	[15:8]	Parameter Value for Arithmetic Operation
N	G or U Parameter Value for Arithmetic Operation.	

PAR_BV	[7:0]	Parameter Value for Arithmetic Operation
n	B or V Parameter Value for Arithmetic Operation.	

Parameter Value should be set to Sequential YUV4:4:4 or RGB888 irrespective of Input Source Format.

Writing should be done in RGB888 Format when Input Format is RGBxxx. When Input Format is YUVxxx and YUVtoRGB Converter is disabled, YUV444 Format is used for Writing. If input format is YUVxxx and YUVtoRGB Converter is enabled, writing is done in RGB888 Format.

S0_PAR Register is effective only when SCTRL.S0ARITHMODE is Fill, Add, Sub and Multiply.

Source 1 Chroma-Key Parameter (S1_CHROMA) 0xF0010068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CHROMA_RY[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHROMA_GU[7:0]								CHROMA_BV[7:0]							

CHROMA_RY	[23:16]	Chroma-Key Value
n	R or Y Chroma-Key Value	

CHROMA_GU	[15:8]	Chroma-Key Value
n	G or U Chroma-Key Value	

CHROMA_BV	[7:0]	Chroma-Key Value
n	B or V Chroma-Key Value	

Chroma-Key Value should be set to Sequential YUV4:4:4 or RGB888 irrespective of Input Source Format.

Writing should be done in RGB888 Format when Input Format is RGBxxx. When Input Format is YUVxxx and YUVtoRGB Converter is disabled, YUV444 Format is used for Writing. If input format is YUVxxx and YUVtoRGB Converter is enabled, writing is done in RGB888 Format.

S1_CHROMA Register is effective only when S_CTRL.S1_CHROMAEN is enabled and OP_CTRL.C0/1_SEL Register is 2bit “01”.

Source 1 Arithmetic Parameter (S1_PAR) 0xF001006C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								PAR_RY[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PAR_GU[7:0]								PAR_BV[7:0]							

PAR_RY	[23:16]	Parameter Value for Arithmetic Operation
n	R or Y Parameter Value for Arithmetic Operation.	

PAR_GU	[15:8]	Parameter Value for Arithmetic Operation
N	G or U Parameter Value for Arithmetic Operation.	

PAR_BV	[7:0]	Parameter Value for Arithmetic Operation
n	B or V Parameter Value for Arithmetic Operation.	

Parameter Value should be set to Sequential YUV4:4:4 or RGB888 irrespective of Input Source Format.

Writing should be done in RGB888 Format when Input Format is RGBxxx. When Input Format is YUVxxx and YUVtoRGB Converter is disabled, YUV444 Format is used for Writing. If input format is YUVxxx and YUVtoRGB Converter is enabled, writing is done in RGB888 Format.

S1_PAR Register is effective only when SCTRL.S1ARITHMODE is Fill, Add, Sub and Multiply.

Source 2 Chroma-Key Parameter (S2_CHROMA) 0xF0010070

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CHROMA_RY[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHROMA_GU[7:0]								CHROMA_BV[7:0]							

CHROMA_RY	[23:16]	Chroma-Key Value
n	R or Y Chroma-Key Value	

CHROMA_GU	[15:8]	Chroma-Key Value
n	G or U Chroma-Key Value	

CHROMA_BV	[7:0]	Chroma-Key Value
n	B or V Chroma-Key Value	

Chroma-Key Value should be set to Sequential YUV4:4:4 or RGB888 irrespective of Input Source Format.

Writing should be done in RGB888 Format when Input Format is RGBxxx. When Input Format is YUVxxx and YUVtoRGB Converter is disabled, YUV444 Format is used for Writing. If input format is YUVxxx and YUVtoRGB Converter is enabled, writing is done in RGB888 Format.

S2_CHROMA Register is effective only when S_CTRL.S2_CHROMAEN is enabled and OP_CTRL.C1_SEL Register is 2bit “01”.

Source 2 Arithmetic Parameter (S2_PAR) 0xF0010074

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								PAR_RY[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PAR_GU[7:0]								PAR_BV[7:0]							

PAR_RY	[23:16]	Parameter Value for Arithmetic Operation
n	R or Y Parameter Value for Arithmetic Operation.	

PAR_GU	[15:8]	Parameter Value for Arithmetic Operation
N	G or U Parameter Value for Arithmetic Operation.	

PAR_BV	[7:0]	Parameter Value for Arithmetic Operation
n	B or V Parameter Value for Arithmetic Operation.	

Parameter Value should be set to Sequential YUV4:4:4 or RGB888 irrespective of Input Source Format.

Writing should be done in RGB888 Format when Input Format is RGBxxx. When Input Format is YUVxxx and YUVtoRGB Converter is disabled, YUV444 Format is used for Writing. If input format is YUVxxx and YUVtoRGB Converter is enabled, writing is done in RGB888 Format.

S2_PAR Register is effective only when SCTRL.S2ARITHMODE is Fill, Add, Sub and Multiply.

Source Control (S_CTRL) 0xF0010078

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				S2ARITHMODE[2:0]			S1ARITHMODE[2:0]			S0ARITHMODE[2:0]			S0/1/2_Y2REN		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	S0/1/2_Y2RMODE[1:0]						S0/1/2_CHROMAEN			S2SEL[1:0]		S1SEL[1:0]		S0SEL[1:0]	

S2_ARITHMODE	[27:25]	Source 2 Arithmetic Mode
00x	Bypass Mode	
010	Fill Mode	
011	Inverter Mode	
100	Add Mode	
101	Substrate Type A Mode	
110	Substrate Type B Mode	
111	Multiplier Mode	

S1_ARITHMODE	[24:22]	Source 1 Arithmetic Mode
00x	Bypass Mode	
010	Fill Mode	
011	Inverter Mode	
100	Add Mode	
101	Substrate Type A Mode	
110	Substrate Type B Mode	
111	Multiplier Mode	

S0_ARITHMODE	[21:19]	Source 0 Arithmetic Mode
00x	Bypass Mode	
010	Fill Mode	
011	Inverter Mode	
100	Add Mode	
101	Substrate Type A Mode	
110	Substrate Type B Mode	
111	Multiplier Mode	

Arithmetic Fnc.	Equation	A	B
BYPASS	Y = A	Source Image	Do not Used
FILL	Y = B	Source Image	S0/1_PAR
INV	Y = 255 - A	Source Image	Do not Used
ADD	Y = A + B	Source Image	S0/1_PAR
SUBA	Y = A – B	Source Image	S0/1_PAR
SUBB	Y = B – A	Source Image	S0/1_PAR
MUL	Y = A x B Mantissa = B[7:6] Fraction = B[5:0]	Source Image	S0/1_PAR

Since Arithmetic Operation is done after YUVtoRGB Converting Operation, operation is working in RGB888 or YUV4:4:4 Format irrespective of Input Source Format. Therefore, writing should be done in RGB888 or YUV4:4:4 format.

If Chroma-Key Value and Selected Source Data are matched when Chroma-Key is enabled, BYPASS Operation is carried out.

S2_Y2REN	[18]	Source 2 YUV to RGB Converter Enable
0	YUV to RGB Converter Disable	
1	YUV to RGB Converter Enable	

To enable YUV to RGB Converter, Data Format of Source 2 decided by SCTL.S2_SEL[2:0] should be YUVxxx.

S1_Y2REN	[17]	Source 1 YUV to RGB Converter Enable
0	YUV to RGB Converter Disable	
1	YUV to RGB Converter Enable	

To enable YUV to RGB Converter, Data Format of Source 1 decided by SCTL.S1_SEL[2:0] should be YUVxxx.

S0_Y2REN	[16]	Source 0 YUV to RGB Converter Enable
0	YUV to RGB Converter Disable	
1	YUV to RGB Converter Enable	

To enable YUV to RGB Converter, Data Format of Source 0 decided by SCTL.S0_SEL[2:0] should be YUVxxx.

S2_Y2RMODE	[14:13]	Source 2 YUVtoRGB Convert Type
00	YUVtoRGB Converter Type0	
01	YUVtoRGB Converter Type1	
10	YUVtoRGB Converter Type2	
11	YUVtoRGB Converter Type3	

S1_Y2RMODE	[12:11]	Source 1 YUVtoRGB Convert Type
00	YUVtoRGB Converter Type0	
01	YUVtoRGB Converter Type1	
10	YUVtoRGB Converter Type2	
11	YUVtoRGB Converter Type3	

S0_Y2RMODE	[10:9]	Source 0 YUVtoRGB Convert Type
00	YUVtoRGB Converter Type0	
01	YUVtoRGB Converter Type1	
10	YUVtoRGB Converter Type2	
11	YUVtoRGB Converter Type3	

YUV4:4:4 to RGB888 Converter Type

YUVtoRGB Format Converter Type 0

$R = Y + 1.371 (V – 128)$

$G = Y – 0.336 (U – 128) – 0.698 (V – 128)$

$B = Y + 1.732 (U – 128)$

YUVtoRGB Format Converter Type 1

$R = 1.164 (Y – 16) + 1.159 (V – 128)$

$G = 1.164 (Y – 16) – 0.391 (U – 128) – 0.813 (V – 128)$

$B = 1.164 (Y – 16) + 2.018 (U – 128)$

YUVtoRGB Format Converter Type 2

$R = Y + 1.540 (V – 128)$

$G = Y – 0.183 (U – 128) – 0.459 (V – 128)$

$B = Y + 1.816 (U – 128)$

YUVtoRGB Format Converter Type 3

$R = 1.164 (Y – 16) + 1.793 (V – 128)$

$G = 1.164 (Y – 16) – 0.213 (U – 128) – 0.534 (V – 128)$

$B = 1.164 (Y – 16) + 2.115 (U – 128)$

S2_CHROMAEN	[8]	Source 2 Chroma-Key Enable for Arithmetic
0	Chroma-Key Disable for Arithmetic	
1	Chroma-Key Enable for Arithmetic	

S1_CHROMAEN	[7]	Source 1 Chroma-Key Enable for Arithmetic
0	Chroma-Key Disable for Arithmetic	
1	Chroma-Key Enable for Arithmetic	

S0_CHROMAEN	[6]	Source 0 Chroma-Key Enable for Arithmetic
0	Chroma-Key Disable for Arithmetic	
1	Chroma-Key Enable for Arithmetic	

S2_SEL	[5:4]	Source 2 Selection
00	Source 2 Disable	
01	Source 2 = Front-End Channel 0	
10	Source 2 = Front-End Channel 1	
11	Source 2 = Front-End Channel 2	

S1_SEL	[3:2]	Source 1 Selection
00	Source 1 Disable	
01	Source 1 = Front-End Channel 0	
10	Source 1 = Front-End Channel 1	
11	Source 1 = Front-End Channel 2	

S0_SEL	[1:0]	Source 0 Selection
00	Source 0 Disable	
01	Source 0 = Front-End Channel 0	
10	Source 0 = Front-End Channel 1	
11	Source 0 = Front-End Channel 2	

Source 0 should select whatever image. In other words, if there is only one Input Source Image in either FCH0 or FCH1 or FCH2, the Input Source Image should be selected to Source 0.

If there are more than 2 Input Source Images in two of FCH0, FCH1 and FCH2, the larger Input Source Image should be selected to S0 and the other should be selected to either S1 or S2.

Source Operator 0 Pattern (OP0_PAT) 0xF0010080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA[7:0]								PAT_RY[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAT_GU[7:0]								PAT_BV[7:0]							

ALPHA	[31:24]	Source Operator 0 Alpha Value
n	Alpha Value for Alpha-Blending Operation of Source Operator 0	

OP0_PAR.ALPHA Register is effective only when OPCTRL.OP0MODE is Alpha-Blending Type 0.

PAT_RY	[23:16]	Source Operator 0 Pattern Value
n	R or Y Pattern Value for Raster Operation of Source Operator 0	

PAT_GU	[15:8]	Source Operator 0 Pattern Value
n	G or U Pattern Value for Raster Operation of Source Operator 0	

PAT_BV	[7:0]	Source Operator 0 Pattern Value
n	B or V Pattern Value for Raster Operation of Source Operator 0	

Source Operator 1 Pattern (OP1_PAT) 0xF0010084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA[7:0]								PAT_RY[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAT_GU[7:0]								PAT_BV[7:0]							

ALPHA	[31:24]	Source Operator 1 Alpha Value
n	Alpha Value for Alpha-Blending Operation of Source Operator 1	

OP1_PAR.ALPHA Register is effective only when OPCTRL.OP1MODE is Alpha-Blending Type 0.

PAT_RY	[23:16]	Source Operator 1 Pattern Value
n	R or Y Pattern Value for Raster Operation of Source Operator 1	

PAT_GU	[15:8]	Source Operator 1 Pattern Value
n	G or U Pattern Value for Raster Operation of Source Operator 1	

PAT_BV	[7:0]	Source Operator 1 Pattern Value
n	B or V Pattern Value for Raster Operation of Source Operator 1	

Source Operation Control (OP_CTRL) 0xF0010088

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							ASEL1[1:0]		CSEL1[1:0]		OP1_MODE[4:0]				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							ASEL0[1:0]		CSEL0[1:0]		OP0_MODE[4:0]				

ASEL1	[24:23]	Alpha Value Selection 1 for Alpha-Blending
0x	Alpha Value is 0 ~ 255 (0% ~ 99.6%)	
10	Alpha Value is 1 ~ 256 (0.39% ~ 100%)	
11	Alpha Value is 0 ~ 127 and 129 ~ 256 (0% ~ 49.6% and 50.39% ~ 100%)	

CSEL1	[22:21]	Chroma-key Source Selection
00	Chroma-Key Operation Disable of Operator 0	
01	Chroma-Key Source = Source 0, Chroma-Key Value = S0_CHROMA	
10	Chroma-Key Source = Source 1, Chroma-Key Value = S1_CHROMA	
11	Chroma-Key Source = Source 2, Chroma-Key Value = S2_CHROMA	

OP1_MODE	[20:16]	Operation 1 Mode	
00000	Blackness	Result = 0	Raster Operation
00001	Merged Copy	Result = P & A	
00010	Merged Paint	Result = ~ A | B	
00011	Pattern Copy	Result = P	
00100	Pattern Inverter	Result = P ^ B	
00101	Pattern Paint	Result = (P | ~ A) | B	
00110	Source Copy	Result = A	
00111	Source Inverter	Result = A ^ B	
01000	Source Paint	Result = A | B	
01001	Source AND	Result = A & B	
01010	Source Erase	Result = A & ~ B	
01011	Not Source Copy	Result = ~ A	
01100	Not Source Erase	Result = ~ (A | B)	
01101	Destination Copy	Result = B	
01110	Destination Inverter	Result = ~ B	
01111	Whiteness	Result = 1	
10xxx	Alpha-Blending Type 0	Alpha-Blending with Alpha-Value	Alpha-Blending
11xxx	Alpha-Blending Type 1	Alpha-Blending with Alpha-Value of ARGB	

ASEL0	[8:7]	Alpha Value Selection 0 for Alpha-Blending
0x	Alpha Value is 0 ~ 255 (0% ~ 99.6%)	
10	Alpha Value is 1 ~ 256 (0.39% ~ 100%)	
11	Alpha Value is 0 ~ 127 and 129 ~ 256 (0% ~ 49.6% and 50.39% ~ 100%)	

CSEL0	[6:5]	Chroma-key Source Selection
00	Chroma-Key Operation Disable of Operator 0	
01	Chroma-Key Source = Source 0, Chroma-Key Value = S0_CHROMA	
10	Chroma-Key Source = Source 1, Chroma-Key Value = S1_CHROMA	
11	Reserved.	

OP0_MODE	[4:0]	Operation 0 Mode	
00000	Blackness	Result = 0	Raster Operation
00001	Merged Copy	Result = P & A	
00010	Merged Paint	Result = ~ A | B	
00011	Pattern Copy	Result = P	
00100	Pattern Inverter	Result = P ^ B	
00101	Pattern Paint	Result = (P | ~ A) | B	
00110	Source Copy	Result = A	
00111	Source Inverter	Result = A ^ B	
01000	Source Paint	Result = A | B	
01001	Source AND	Result = A & B	
01010	Source Erase	Result = A & ~ B	
01011	Not Source Copy	Result = ~ A	
01100	Not Source Erase	Result = ~ (A | B)	
01101	Destination Copy	Result = B	
01110	Destination Inverter	Result = ~ B	
01111	Whiteness	Result = 1	
10xxx	Alpha-Blending Type 0	Alpha-Blending with Alpha-Value	Alpha-Blending
11xxx	Alpha-Blending Type 1	Alpha-Blending with Alpha-Value of ARGB	

Raster Operation & Alpha-Blending of Operator 0

Mode	Equation	A	B	P	ALPHA	CHROMA
Blackness	Y = 0	Source 0	Not Used.	Not Used.	Not Used.	S1_CHROMA
Merged Copy	Y = P & A	Source 0	Not Used.	PAT0[23:00]	Not Used.	S1_CHROMA
Merged Paint	Y = ~ A | B	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Pattern Copy	Y = P	Source 0	Not Used.	PAT0[23:00]	Not Used.	S1_CHROMA
Pattern Inverter	Y = P ^ B	Source 0	Source 1	PAT0[23:00]	Not Used.	S1_CHROMA
Pattern Paint	Y = (P | ~ A) | B	Source 0	Source 1	PAT0[23:00]	Not Used.	S1_CHROMA
Source Copy	Y = A	Source 0	Not Used.	Not Used.	Not Used.	S1_CHROMA
Source Inverter	Y = A ^ B	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Source Paint	Y = A | B	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Source AND	Y = A & B	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Source Erase	Y = A & ~ B	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Not Source Copy	Y = ~ A	Source 0	Not Used.	Not Used.	Not Used.	S1_CHROMA
Not Source Erase	Y = ~ (A | B)	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Destination Copy	Y = B	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Destination Inverter	Y = ~ B	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Whiteness	Y = 1	Source 0	Not Used.	Not Used.	Not Used.	S1_CHROMA
Alpha-Blending Type 0	Y = ALPHA * B + (1 – ALPHA) *A	Source 0	Source 1	Not Used.	OP0_ALPHA	S1_CHROMA
Alpha-Blending Type 1	Y = ALPHA * B + (1 – ALPHA) *A	Source 0	Source 1	Not Used.	Alpha of Source 1	S1_CHROMA

In Alpha-Blending Type 0, OP0_ALPHA of OP_PAT0 Register is used as Alpha Value.

In Alpha-Blending Type 1, Alpha Value of Source 1 Data is used as Alpha Value. Therefore, in this case, Source Data Format should definitely be ARGB.

Raster Operation & Alpha-Blending of Operator 1

Mode	Equation	A	B	P	ALPHA	CHROMA
Blackness	Y = 0	Y of OP 0	Not Used.	Not Used.	Not Used.	S2_CHROMA
Merged Copy	Y = P & A	Y of OP 0	Not Used.	PAT1[23:00]	Not Used.	S2_CHROMA
Merged Paint	Y = ~ A | B	Y of OP 0	Source 2	Not Used.	Not Used.	S2_CHROMA
Pattern Copy	Y = P	Y of OP 0	Not Used.	PAT1[23:00]	Not Used.	S2_CHROMA
Pattern Inverter	Y = P ^ B	Y of OP 0	Source 2	PAT1[23:00]	Not Used.	S2_CHROMA
Pattern Paint	Y = (P | ~ A) | B	Y of OP 0	Source 2	PAT1[23:00]	Not Used.	S2_CHROMA
Source Copy	Y = A	Y of OP 0	Not Used.	Not Used.	Not Used.	S2_CHROMA
Source Inverter	Y = A ^ B	Y of OP 0	Source 2	Not Used.	Not Used.	S2_CHROMA
Source Paint	Y = A | B	Y of OP 0	Source 2	Not Used.	Not Used.	S2_CHROMA
Source AND	Y = A & B	Y of OP 0	Source 2	Not Used.	Not Used.	S2_CHROMA
Source Erase	Y = A & ~ B	Y of OP 0	Source 2	Not Used.	Not Used.	S2_CHROMA
Not Source Copy	Y = ~ A	Y of OP 0	Not Used.	Not Used.	Not Used.	S2_CHROMA
Not Source Erase	Y = ~ (A | B)	Y of OP 0	Source 2	Not Used.	Not Used.	S2_CHROMA
Destination Copy	Y = B	Y of OP 0	Source 2	Not Used.	Not Used.	S2_CHROMA
Destination Inverter	Y = ~ B	Y of OP 0	Source 2	Not Used.	Not Used.	S2_CHROMA
Whiteness	Y = 1	Y of OP 0	Not Used.	Not Used.	Not Used.	S2_CHROMA
Alpha-Blending Type 0	Y = ALPHA * B + (1 – ALPHA) *A	Y of OP 0	Source 2	Not Used.	OP1_ALPHA	S2_CHROMA
Alpha-Blending Type 1	Y = ALPHA * B + (1 – ALPHA) *A	Y of OP 0	Source 2	Not Used.	Alpha of Source 2	S2_CHROMA

In Alpha-Blending Type 0, OP0_ALPHA of OP_PAT0 Register is used as Alpha Value.

In Alpha-Blending Type 1, Alpha Value of Source 1 Data is used as Alpha Value. Therefore, in this case, Source Data Format should definitely be ARGB.

Back-End Destination Address 0 (BCH_DADDR0) 0xF0010090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCH_DADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_DADDR0[15:2]														Reserved	

Destination Image Format is Separated YUV X:X:X, the value of the address is Source Address of Y region.

Destination Image Format is not Separated YUV X:X:X, the value of the address is Source Address.

Back-End Destination Address 1 (BCH_DADDR1) 0xF0010094

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCH_DADDR1[31:16]															
9	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_DADDR1[15:2]														Reserved	

In case Destination Image Format is Separated YUV X:X:X, the value of the address is Source Address of U region.

In case Destination Image Format is not Separated YUV X:X:X, the value of the address is not used.

Back-End Destination Address 2 (BCH_DADDR2) 0xF0010098

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCH_DADDR2[31:16]															
9	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_DADDR2[15:2]														Reserved	

In case Destination Image Format is Separated YUV X:X:X, the value of the address is Source Address of V region.

In case Destination Image Format is not Separated YUV X:X:X, the value of the address is not used.

Back-End Channel Destination Frame Pixel Size (BCH_DFSIZE) 0xF001009C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				DFSIZE_Y[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DFSIZE_X[11:0]											

DFSIZE_Y	[27:16]	Destination Frame Pixel Size Y
N (0 ~ 4095)	N =Y Axis Destination Frame Pixel Size	

DFSIZE_X	[11:0]	Destination Frame Pixel Size X
N (0 ~ 4095)	N =X Axis Destination Frame Pixel Size	

Back-End Channel Destination Pixel Offset (BCH_DOFF) 0xF00100A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				DOFF_Y[11:0]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				DOFF_X[11:0]											

DOFF_Y	[27:16]	Destination Pixel Offset Y
N (0 ~ 4095)	N =Y Axis Destination Pixel Offset	

DOFF_X	[11:0]	Destination Pixel Offset X
N (0 ~ 4095)	N =X Axis Destination Pixel Offset	

Destination Image Size & Frame Size & Offset Limitation

BCH_DCTRL.DDFRM	Destination Frame Size / Image Size / Offset Limitation
RGB332(8bpp)	No Limitation
RGB454	No Limitation
RGB565	No Limitation
RGB666	No Limitation
RGB888	No Limitation
Separated YUV 4:4:4	No Limitation
Separated YUV 4:4:0	X axis = No Limitation, Y axis = multiplies of 2
Separated YUV 4:2:2	X axis = multiplies of 2, Y axis = No Limitation
Separated YUV 4:2:0	X axis = multiplies of 2, Y axis = multiplies of 2
Separated YUV 4:1:1	X axis = multiplies of 4, Y axis = No Limitation
Separated YUV 4:1:0	X axis = multiplies of 4, Y axis = multiplies of 2
Interleaved YUV 4:2:2	X axis = multiplies of 2, Y axis = No Limitation
Interleaved YUV 4:2:0	X axis = multiplies of 2, Y axis = multiplies of 2
Sequential YUV 4:4:4	No Limitation
Sequential YUV 4:2:2	X axis = multiplies of 2, Y axis = No Limitation

Destination Image Size is decided by both Source 0 size controlled by S0_CTRL and Back-End Destination Operation Mode controlled by BCH_CTRL.

Back-End Channel Control (BCH_DCTRL) 0xF00100A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										MABC	Reserved		YSEL	XSEL[1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R2Y	R2YMODE		Res.	DSUV	OPMODE[2:0]			0	DOP	DEN	DDFRM[4:0]				

MABC	[21]	AXI Bus Master Address Boundary Control
0	AXI Bus Master Address Boundary = 4KByte	
1	AXI Bus Master Address Boundary = 1KByte	

YSEL	[18]	YUV4:4:4 to YUVx:x:x Y Control
n	N = 0 or 1	

XSEL	[17:16]	YUV4:4:4 to YUVx:x:x X Control
n	N = 0 ~ 3	

Since inside Overlay Mixer, only RGB888 and Sequential YUV4:4:4: exist as Data Format, if Destination Data Format is RGBxxx, RGB888 is converted to RGBxxx. If Destination Data Format is Sequential / Separated YUVxxx, Sequential YUV4:4:4 is converted to Sequential / Separated YUVxxx.

In this case, when Sequential YUV4:4:4 is converted to Sequential / Separated YUVxxx, BCH_CTRL.X/YSEL Register is reflected.

R2Y	[15]	Destination Format Converter Control
0	RGB to YUV Converter Disable	
1	RGB to YUV Converter Enable	

R2YMODE	[14:13]	RGBtoYUV Converter Type
0	RGBtoYUV Converter Type0	
1	RGBtoYUV Converter Type1	
2	RGBtoYUV Converter Type2	
3	RGBtoYUV Converter Type3	

RGB to YUV Converter Type

RGBtoYUV Format Converter Type 0

Y = 0.299 R + 0.587 G + 0.114 B
U = -0.172 R – 0.339 G + 0.511 B + 128
V = 0.511 R – 0.428 G – 0.083 B + 128

RGBtoYUV Format Converter Type 1

Y = 0.257 R + 0.504 G + 0.098 B + 16
U = -0.148 R – 0.291 G + 0.439 B + 128
V = 0.439 R – 0.368 G – 0.071 B + 128

RGBtoYUV Format Converter Type 2

Y = 0.213 R + 0.715 G + 0.072 B
U = -0.117 R – 0.394 G + 0.511 B + 128
V = 0.511 R – 0.464 G – 0.047 B + 128

RGBtoYUV Format Converter Type 3

Y = 0.183 R + 0.614 G + 0.062 B + 16
U = -0.101 R – 0.338 G + 0.439 B + 128
V = 0.439 R – 0.399 G – 0.040 B + 128

DSUV	[11]	Destination Swap U for V
0	Keep on U and V (LSB = U, MSB = V)	
1	Swap U for V (LSB = V, MSB = U)	

The above applies only when Data Format is Interleaved YUV Format

OPMODE	[10:8]	Operation Mode of Back-End DMA
00x	Data Copy	
010	Horizontal Mirror	
011	Vertical Mirror	
100	Vertical & Horizontal Mirror	
101	90 Degree Rotate (Clockwise)	
110	180 Degree Rotate (Clockwise)	
111	270 Degree Rotate (Clockwise)	

DOP	[6]	Dithering Operation Type
0	1-Bit Toggle Operation	
1	Add 1 Operation	

The above applies only when Data Format is RGB Format, not RGB888 / ARGB8888 Format

DEN	[5]	Dithering Enable
0	Dithering Disable	
1	Dithering Enable	

The above applies only when Data Format is RGB Format, not RGB888 / ARGB8888 Format

DDFRM	[4:0]	Destination Data Format of Back-End DMA
00000	Separated YUV 4:4:4	
00001	Separated YUV 4:4:0	
00010	Separated YUV 4:2:2	
00011	Separated YUV 4:2:0	
00100	Separated YUV 4:1:1	
00101	Separated YUV 4:1:0	
00110	Interleaved YUV 4:2:2	
00111	Interleaved YUV 4:2:0	
01000	Sequential YUV 4:4:4	
01001	Sequential YUV 4:2:2	
0101x	RGB332 (8bpp)	
01100	RGB444	
01101	ARGB4444	
01110	RGB454	
01111	ARGB3454	
10000	RGB555	
10001	ARGB1555	
1001x	RGB565	
10100	RGB666	
10101	ARGB4666	
10110	ARGB6666	
10111	RGB888	
11000	ARGB4888	
else	ARGB8888	

Back-End Channel Destination Dither Matrix 0 (BCH_DDMAT0) **0xF00100B0**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			DDMAT03[4:0]					Reserved			DDMAT02[4:0]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DDMAT01[4:0]					Reserved			DDMAT00[4:0]				

Back-End Channel Destination Dither Matrix 1 (BCH_DDMAT1) **0xF00100B4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			DDMAT07[4:0]					Reserved			DDMAT06[4:0]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DDMAT05[4:0]					Reserved			DDMAT04[4:0]				

Back-End Channel Destination Dither Matrix 2 (BCH_DDMAT2) **0xF00100B8**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			DDMAT11[4:0]					Reserved			DDMAT10[4:0]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DDMAT09[4:0]					Reserved			DDMAT08[4:0]				

Back-End Channel Destination Dither Matrix 3 (BCH_DDMAT3) **0xF00100BC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			DDMAT15[4:0]					Reserved			DDMAT14[4:0]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DDMAT13[4:0]					Reserved			DDMAT12[4:0]				

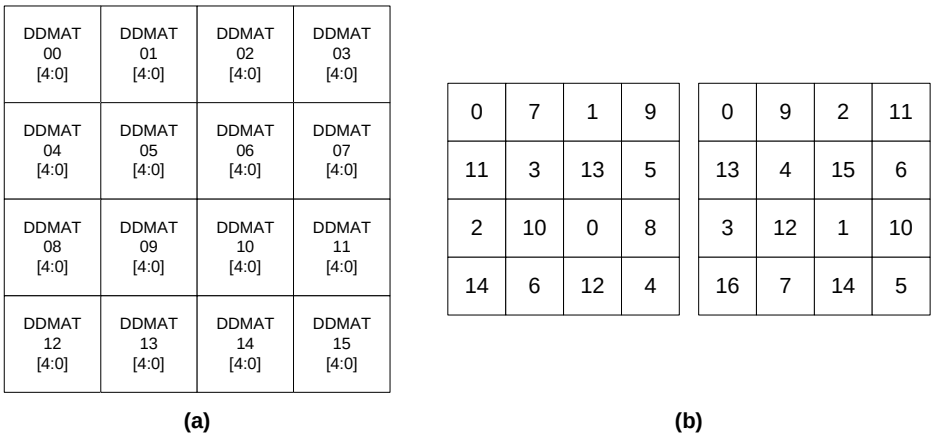


Figure 4.7 (a) 16 Level 4x4 Dither Matrix (b) Example of Dither Matrix Setting

Overlay Mixer Control (OM_CTRL) 0xF00100C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															IEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved													EN[2:0]		

IEN	[16]	Overlay Mixer Interrupt Enable
0	Interrupt Disable	
1	Interrupt Enable	

EN	[2:0]	Overlay Mixer Enable
000	Overlay Mixer Disable	
001	Overlay Mixer Enable With Front-End Channel 0	
010	Overlay Mixer Enable With Front-End Channel 1	
011	Overlay Mixer Enable With Front-End Channel 2	
100	Overlay Mixer Enable With Front-End Channel 0,1	
101	Overlay Mixer Enable With Front-End Channel 0,2	
110	Overlay Mixer Enable With Front-End Channel 1,2	
111	Overlay Mixer Enable With Front-End Channel 0,1,2	

After all transfers complete, EN[2:0] Register is automatically cleared to 3bit “000”.

EN Register should be set last after all Control Registers are set.

Overlay Mixer Interrupt Request Control (OM_IREQ) 0xF00100C4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															FLG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved															IRQ

FLG	[16]	Overlay Mixer Flag Bit
0 (Read)	Transfer of Overlay Mixer is Not Completed.	
0 (Write)	Flag Bit Clear	
1	Transfer of Overlay Mixer is Completed.	

IRQ	[0]	Overlay Mixer Interrupt Request
0 (Read)	Interrupt Request is not occurred.	
0 (Write)	Interrupt Request is cleared.	
1	Interrupt Request is occurred.	

Front-End Channel 0 Lookup Table (FCH0_LUT) **0xF0010400 ~ 0xF00107FF**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCH0_LUT[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCH0_LUT[15:0]															

When Overlay Mixer is operating, CPU can't write and read this register.

Front-End Channel 1 Lookup Table (FCH1_LUT) **0xF0010800 ~ 0xF0010BFF**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCH0_LUT[31:16]															
9	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCH0_LUT[15:0]															

When Overlay Mixer is operating, CPU can't write and read this register.

Front-End Channel 2 Lookup Table (FCH2_LUT) **0xF0010C00 ~ 0xF0010FFF**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCH0_LUT[31:16]															
9	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCH0_LUT[15:0]															

When Overlay Mixer is operating, CPU can't write and read this register.

Use of LUT

FCH_SCTRL.SDFRM	Supported LUT Field (ARGB8888)			
	A	R	G	B
RGB332(8bpp)	Yes	Yes	Yes	Yes
ARGB4444	Yes	No	No	No
ARGB3454	Yes			
ARGB1555	Yes			
ARGB4666	Yes			
ARGB6666	Yes			
ARGB4888	Yes			
ARGB8888	Yes			
RGB444	No	No	No	No
RGB454				
RGB555				
RGB565				
RGB666				
RGB888				
Separated YUV x:x:x				
Interleaved YUV x:x:x				

4.4 Overlay Mixer Operation

4.4.1 Source Rotate / Mirror and Destination Rotate / Mirror Operation

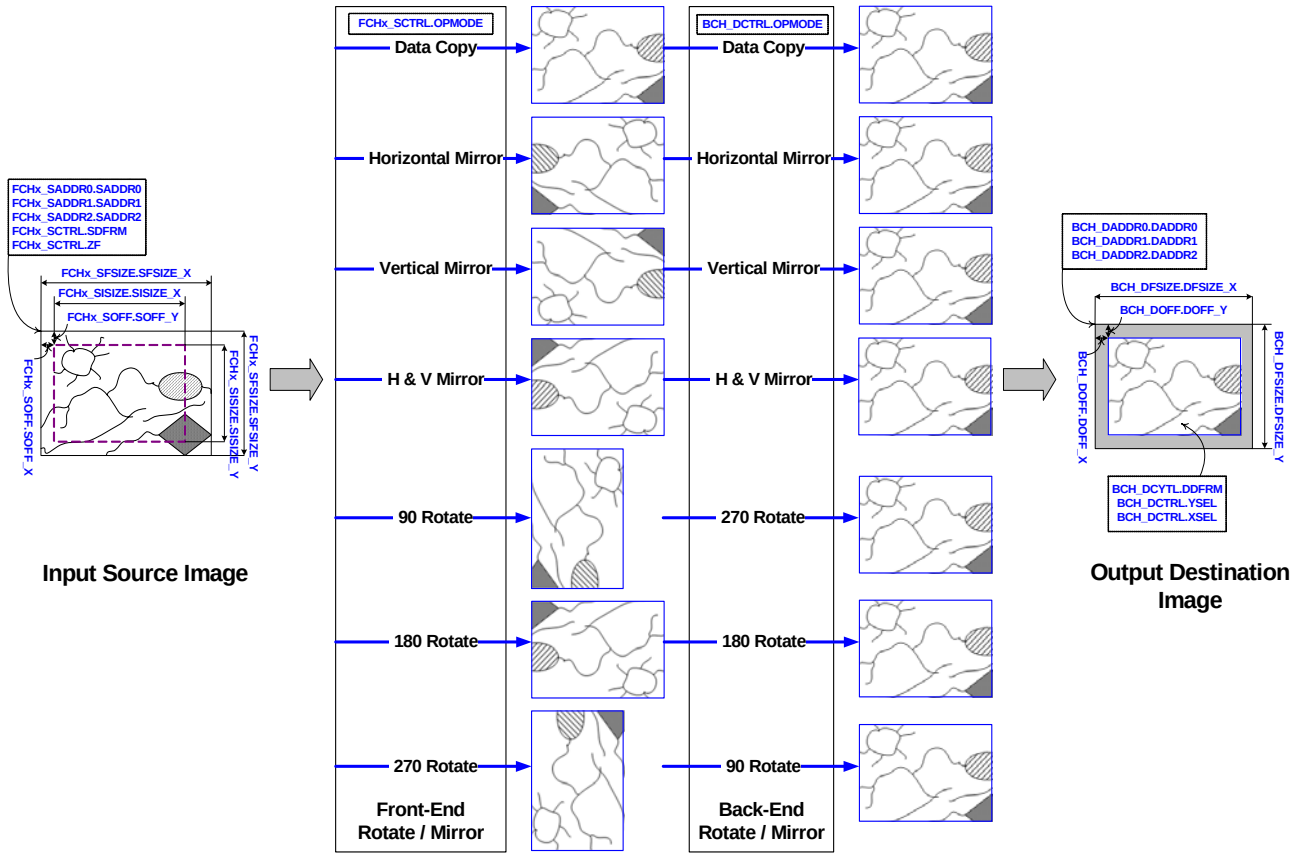


Figure 4.8 Source & Destination Mirror / Rotate Operation

Figure 4.8 illustrates one of functions supported by Overlay Mixer, Rotate / Mirror function. The figure shows Internal Register Setting when there is one Input Source Image and Input Image Source is FCH1.

As explained before, if the number of an image is One, the image should be selected to Source 0. Window Offset of the image which is selected to Source 0 should definitely be '0'. In addition to this, OM_CTRL.EN Register should be set last.

(1) FCH1_SADDR0 / 1 / 2 : The Input Source Image Start Address.
(2) FCH1_SCTRL.SDFRM, FCH1_SCTRL.ZF : The Input Source Image Data Format
(3) FCH1_SFSIZE : The Input Source Image Frame Size
(4) FCH1_SISIZE : The Input Source Image Size
(5) FCH1_SOFF : The Input Source Image Offset
(6) FCH1_SCTRL.OPMODE : FCH1 Operation Mod
(7) FCH1_WOFF : Should be Zero.
(8) S_CTRL.S0SEL : Should be Front-End Channel 1
(9) S_CTRL.S0_ARITHMOD : Should be Bypass Mode.
(10) S_CTRL.S0_Y2REN, S0_CHROMAEN : Should be Disable.
(11) OP_CTRL.C0SEL, C1SEL : Should be Disable.
(12) OP_CTRL.OP0MODE, OP1MODE: Should be Source Copy
(13) BCH_DCTRL.OPMODE : BCH Operation Mode.
(14) BCH_DCTRL.Y2R : Should be Disable.
(15) BCH_DADDR0 / 1 / 2 : The Output Destination Image Start Address.
(16) BCH_DCTRL.DDFRM, DCH_DCTRL.X / YSEL : The Output Destination Image Data Format
(17) BCH_DFSIZE : The Output Destination Image Frame Size

(18) BCH_DOFF : The Output Destination Image Offset
(19) OM_CTRL : OM_CTRL.EN = b'010

4.4.2 YUV to RGB Format Conversion

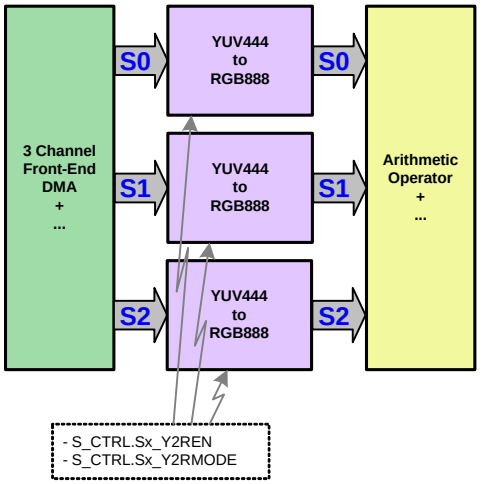


Figure 4.9 Source YUV to RGB Conversion

Figure 4.9 explains one of functions supported by Overlay Mixer, 3 Channel YUV to RGB Format Conversion function. As can be seen, 3 Channel YUV Source Image of Overlay Mixer can be converted to RGB888. S0 and S1 in Figure 4.9 are the images selected by S_CTRL.SxSEL Register.

Below shows 4 equations in YUV to RGB Conversion per each S_CTRL.Sx_Y2RMODE Register.

- YUVtoRGB Format Converter Type 0
  - R = Y + 1.371 (V – 128)
  - G = Y – 0.336 (U – 128) – 0.698 (V – 128)
  - B = Y + 1.732 (U – 128)
- YUVtoRGB Format Converter Type 1
  - R = 1.164 (Y – 16) + 1.159 (V – 128)
  - G = 1.164 (Y – 16) – 0.391 (U – 128) – 0.813 (V – 128)
  - B = 1.164 (Y – 16) + 2.018 (U – 128)
- YUVtoRGB Format Converter Type 2
  - R = Y + 1.540 (V – 128)
  - G = Y – 0.183 (U – 128) – 0.459 (V – 128)
  - B = Y + 1.816 (U – 128)
- YUVtoRGB Format Converter Type 3
  - R = 1.164 (Y – 16) + 1.793 (V – 128)
  - G = 1.164 (Y – 16) – 0.213 (U – 128) – 0.534 (V – 128)
  - B = 1.164 (Y – 16) + 2.115 (U – 128)

4.4.3 Arithmetic Operation

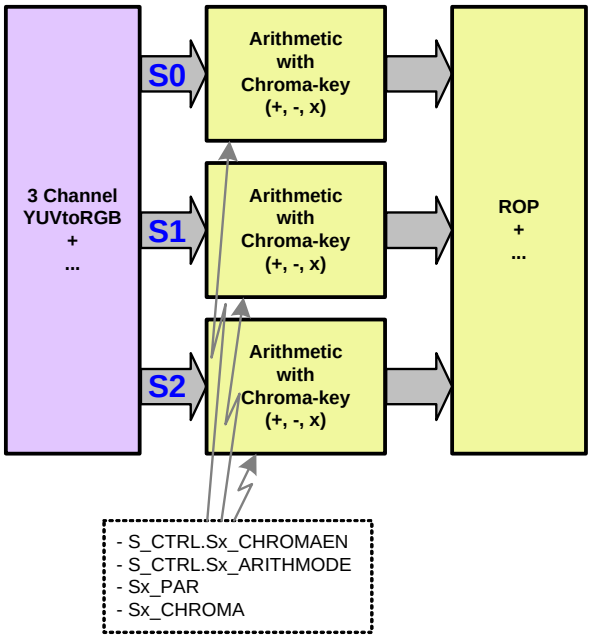


Figure 4.10 Arithmetic Operation

Figure 4.10 explains 3 Channel Arithmetic Operation, one of functions which Overlay Mixer supports. As can be seen above, arithmetic operation for 3 Channel Image Source is available in Overlay Mixer. In the figure, S0, S1 and S2 are the results of performing YUV to RGB Conversion in the images selected by S_CTRL.SxSEL Register.

The following is about Arithmetic Operation per each S_CTRL.Sx_ARITHMODE Register.

	A	B	Result	
			A = Sx_CHROMA	A != Sx_CHROMA
BYPASS	Source 0/1/2 Image	Do not Used	A	Y = A
FILL	Do not Used	Sx_PAR	A	Y = B
INV	Source 0/1/2 Image	Do not Used	A	Y = 255 - A
ADD	Source 0/1/2 Image	Sx_PAR	A	Y = A + B
SUBA	Source 0/1/2 Image	Sx_PAR	A	Y = A – B
SUBB	Source 0/1/2 Image	Sx_PAR	A	Y = B – A
MUL	Source 0/1/2 Image	Sx_PAR	A	Y = A x B Mantissa = B[7:6] Fraction = B[5:0]

As can be seen above, Arithmetic Operation is performed after Source YUV to RGB Conversion.

Therefore, for Arithmetic Operation with Chroma-Key, the value after Source YUV to RGB Conversion should be set as Chroma-Key Vaule (Sx_CHROMA).

4.4.4 Alpha-Blending and ROP with Window Offset and Chroma-Key

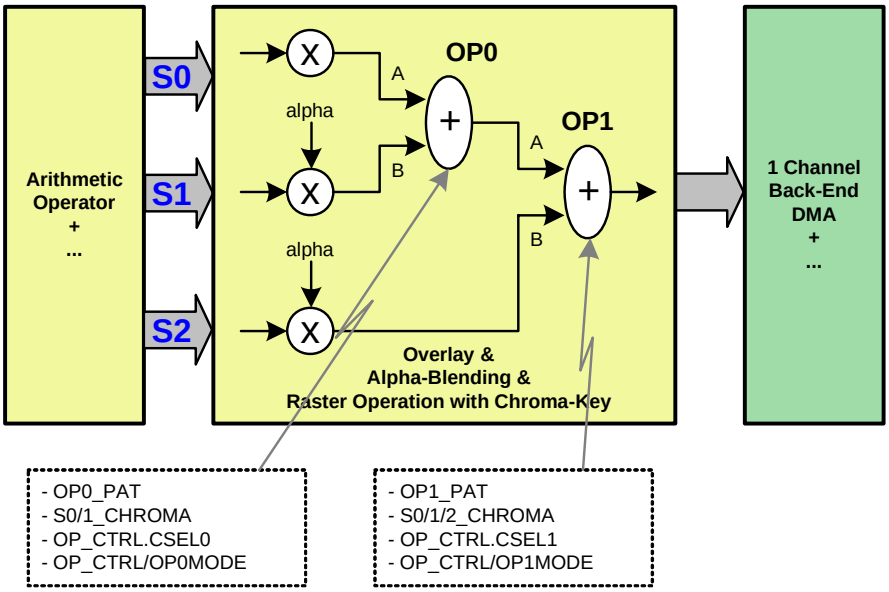


Figure 4.11 ROP / Alpha-Blending Block Diagram

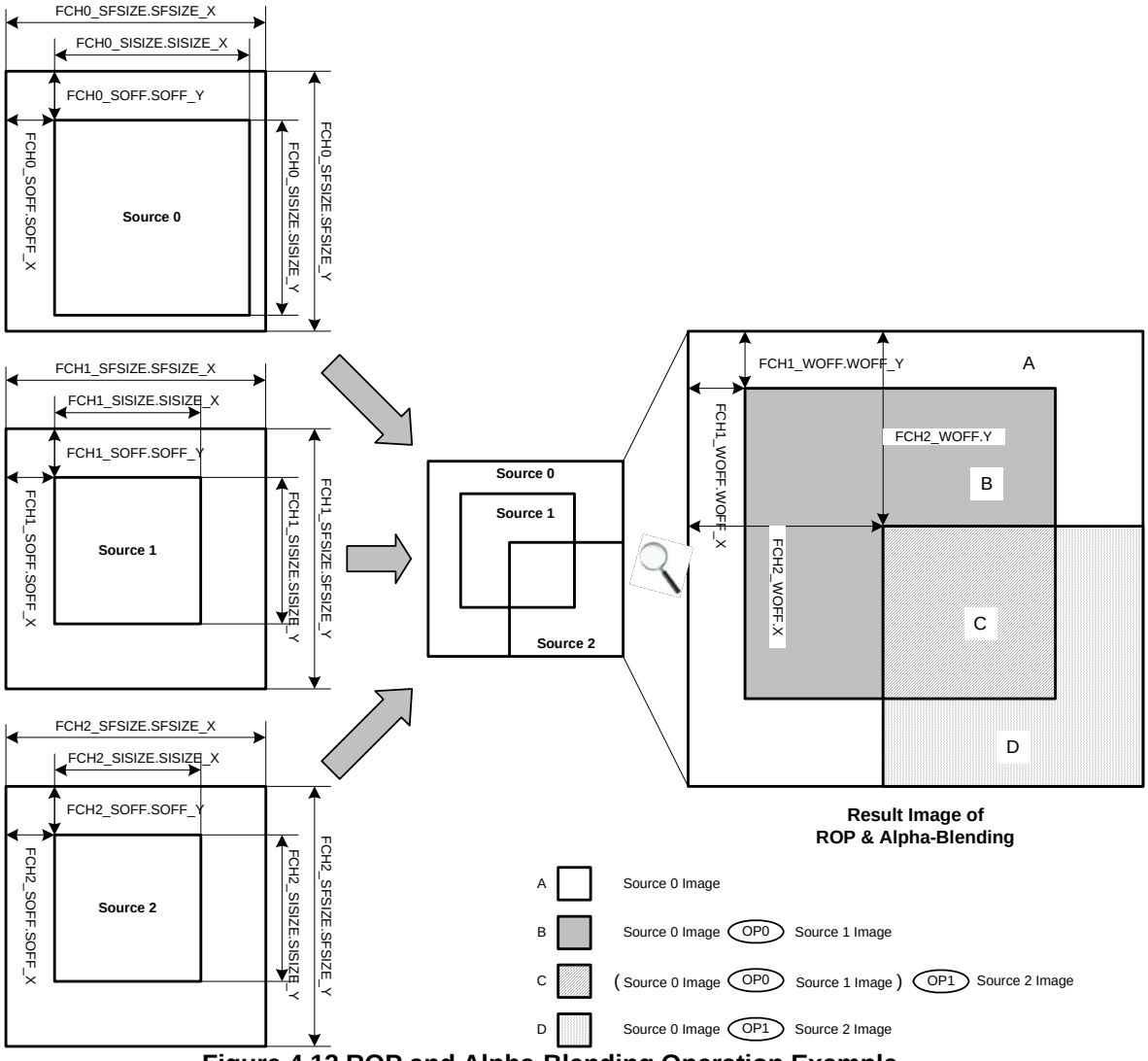


Figure 4.12 ROP and Alpha-Blending Operation Example

This figure illustrates ROP, Alpha-Blending Operation. It assumes that FCH0 is Source 0 and FCH1, FCH2 are Source 1 and Source 2 respectively. As the figure explains, Source 0 Image should definitely include other Source Images.

In addition, in case of Source 0 Image, Window Offset should definitely be '0'. Other than these, there are no requirements for ROP and Alpha-Blending Operation.

Below shows the function of Operator 0 / 1.

Mode	Result of Operator 0		
	A = Source 0 selected by S_CTRL.S0SEL		
	B = Source 1 selected by S_CTRL.S1SEL		
	P = OP0_PAT[23:0]		
	ALPHA0 = OP0_PAT[31:24] / 256		
	ALPHA1 = Alpha-Value of ARGB Format Source 1 Data		
	OPCTRL_CSEL0	01	10
		A = S0_CHROMA	B = S1_CHROMA
Blackness	Y = 0	B	A
Merged Copy	Y = P & A	B	A
Merged Paint	Y = ~ A | B	B	A
Pattern Copy	Y = P	B	A
Pattern Invert	Y = P ^ B	B	A
Pattern Paint	Y = (P | ~ A) | B	B	A
Src Copy	Y = A	B	A
Src Inverter	Y = A ^ B	B	A
Srd Paint	Y = A | B	B	A
Src AND	Y = A & B	B	A
Src Erase	Y = A & ~ B	B	A
Not Src Copy	Y = ~ A	B	A
Not Src Erase	Y = ~ (A | B)	B	A
Dst Copy	Y = B	B	A
Dst Inverter	Y = ~ B	B	A
Whiteness	Y = 1	B	A
Alpha-Blending 0	Y = ALPHA0 * B + (1 – ALPHA0) *A	B	A
Alpha-Blending 1	Y = ALPHA1 * B + (1 – ALPHA1) *A	B	A

Mode	Result of Operator 1		
	A = Result Image of Operator 0		
	B = Source 2 selected by S_CTRL.S2SEL		
	P = OP1_PAT[23:0]		
	ALPHA0 = OP1_PAT[31:24] / 256		
	ALPHA1 = Alpha-Value of ARGB Format Source 1 Data		
	OPCTRL_CSEL1	01	10
		A = S0_CHROMA	B = S1_CHROMA
Blackness	Y = 0	B	A
Merged Copy	Y = P & A	B	A
Merged Paint	Y = ~ A | B	B	A
Pattern Copy	Y = P	B	A
Pattern Invert	Y = P ^ B	B	A
Pattern Paint	Y = (P | ~ A) | B	B	A
Src Copy	Y = A	B	A
Src Inverter	Y = A ^ B	B	A
Srd Paint	Y = A | B	B	A
Src AND	Y = A & B	B	A
Src Erase	Y = A & ~ B	B	A
Not Src Copy	Y = ~ A	B	A
Not Src Erase	Y = ~ (A | B)	B	A
Dst Copy	Y = B	B	A
Dst Inverter	Y = ~ B	B	A
Whiteness	Y = 1	B	A
Alpha-Blending 0	Y = ALPHA0 * B + (1 – ALPHA0) *A	B	A
Alpha-Blending 1	Y = ALPHA1 * B + (1 – ALPHA1) *A	B	A

4.4.5 RGB to YUV Format Conversion

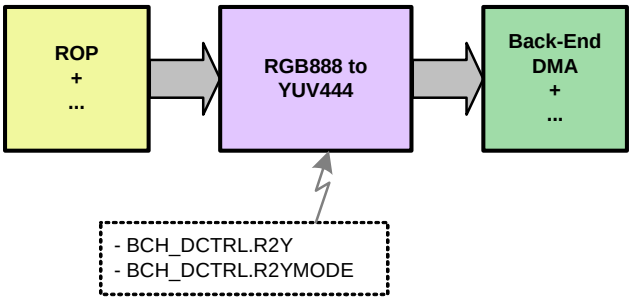


Figure 4.13 RGB to YUV Format Converter

The figure is about 1 Channel RGB to YUV Format Conversion function provided by Overlay Mixer.

As can be seen above, RGB to YUV Format Conversion is performed after ROP and Alpha-Blending Operation.

The following explains the equation of RGB to YUV Conversion per BCH_DCTRL.R2YMODE Register.

- RGBtoYUV Format Converter Type 0
  - Y = 0.299 R + 0.587 G + 0.114 B
  - U = -0.172 R – 0.339 G + 0.511 B + 128
  - V = 0.511 R – 0.428 G – 0.083 B + 128
- RGBtoYUV Format Converter Type 1
  - Y = 0.257 R + 0.504 G + 0.098 B + 16
  - U = -0.148 R – 0.291 G + 0.439 B + 128
  - V = 0.439 R – 0.368 G – 0.071 B + 128
- RGBtoYUV Format Converter Type 2
  - Y = 0.213 R + 0.715 G + 0.072 B
  - U = -0.117 R – 0.394 G + 0.511 B + 128
  - V = 0.511 R – 0.464 G – 0.047 B + 128
- RGBtoYUV Format Converter Type 3
  - Y = 0.183 R + 0.614 G + 0.062 B + 16
  - U = -0.101 R – 0.338 G + 0.439 B + 128
  - V = 0.439 R – 0.399 G – 0.040 B + 128

4.4.6 Destination RGB Dithering

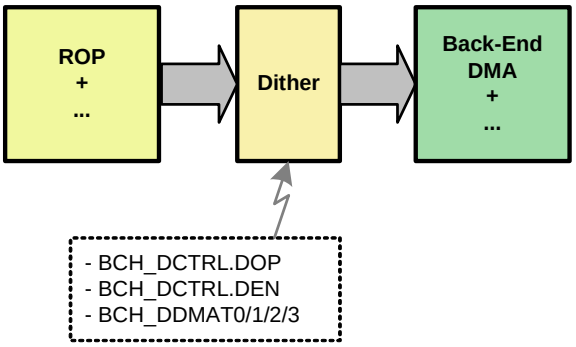


Figure 4.14 Destination RGB Dithering

The figure is about 1 Channel RGB dithering function provided by Overlay Mixer. Patterned Dithering Algorithm is used in dithering algorithm of Overlay Mixer. Below figure illustrates detail operation of Patterned Dithering.

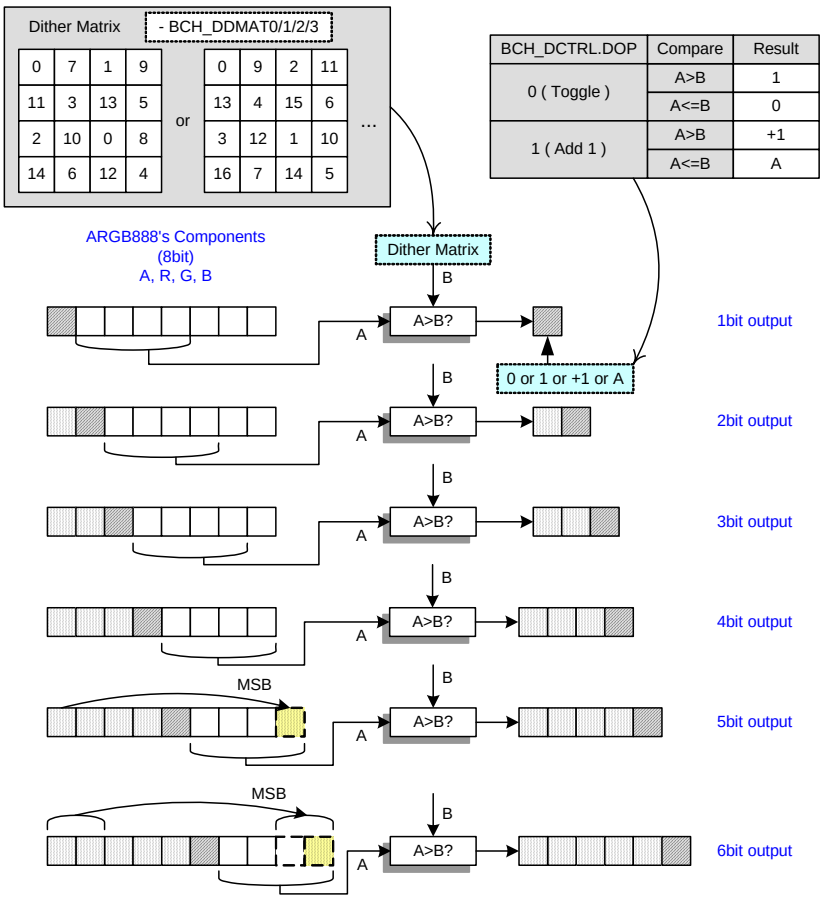


Figure 4.15 Destination RGB Dithering Operation

5 2D/3D GPU

5.1 Overview

The 2D/3D GPU is a hardware accelerator for 2D and 3D graphics systems.
The GPU consists of:

- A pixel processor
- A geometry processor
- A Memory Management Unit (MMU)
- Associated software

The GPU and its associated software are compatible with the following graphics standards:

- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenVG 1.0

5.1.1 Pixel Processor Features

The pixel processor features are:

- Programmable fragment shader
- Access to framebuffer from fragment shaders
- Alpha blending
- Arbitrary memory reads and writes
- Complete non-power-of-2 texture support
- Cube mapping
- Dynamic recursion
- Fast dynamic branching
- Fast trigonometric functions, including arctangent
- Full floating-point arithmetic
- Framebuffer blend with destination Alpha
- High Dynamic Range (HDR) textures and framebuffers
- Indexable texture samplers
- Line, quad, triangle and point sprites
- Multiple render targets
- No limit on program length
- Perspective Anisotropic Filtering (AF)
- Perspective correct texturing
- Point sampling, bilinear, and trilinear filtering
- Programmable mipmap level-of-detail biasing and replacement
- Register indirect jumps
- Stencil buffering, 8-bit
- Two-sided stencil
- Unlimited dependent texture reads
- Virtualized texture samplers
- 4-level hierarchical Z and stencil operations
- 4 times and 16 times Full Scene Anti-Aliasing (FSAA)
- 4-bit per texel texture compression

5.1.2 Geometry Processor Features

The geometry processor features are:

- Programmable vertex shader
- Autonomous operation tile list generation
- Flexible input and output formats
- Indexed and non-indexed geometry input
- Primitive constructions with points, lines, triangles and quads.

5.2 GPU Structure

This section gives a brief description of the structure of the GPU.

- The pixel processor. It uses a list of primitives generated by the geometry processor to produce a final image that is displayed on the screen. The pixel processor uses a total of 23 SRAM macros. There are 33 instances that yield a total size of 32.76KB.
- A programmable geometry processor that generates lists of primitives for the pixel processor to draw. The geometry processor uses a total of nine SRAM macros. There are 15 instances that yield a total size of 16.81KB
- A full-featured Memory Management Unit (MMU). All memory accesses from the pixel and geometry processor use the MMU for access checking and translation.
- AXI interconnect system bus protocol targeted at high performance, high clock frequency system designs.

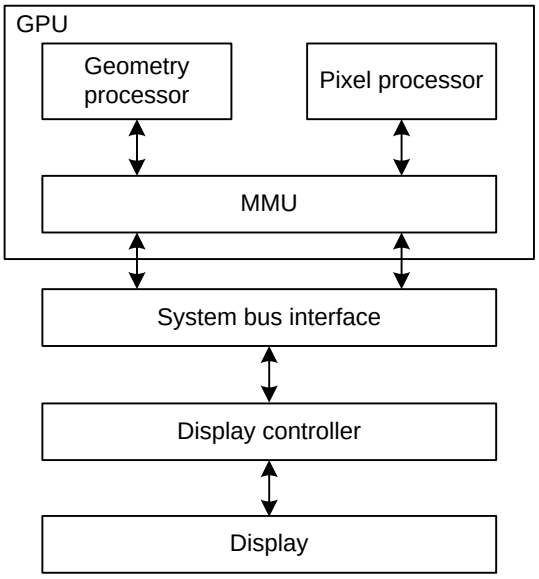


Figure 5.1 2D/3D Graphics System

5.3 Pixel Processor Register Description

The pixel renderer configuration registers consist of an 8KB address space divided by a set of register blocks. The memory extends from a bse address 0x0000 to a maximum address of 0x1F40. Figure 5.2 shows the pixel register map split into resions.

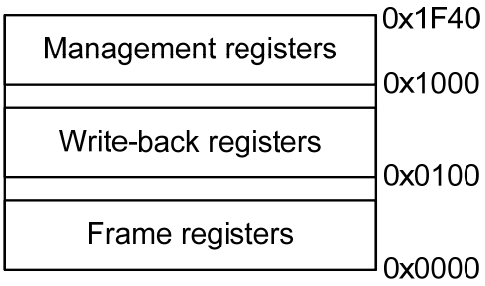


Figure 5.2 Pixel Processor Register Map

Frame registers

The Frame Registers contain frame data that is static during the rendering of a frame, and not required in the memory data structures. Because this data is typically written to in bursts from the driver, the registers are laid out as 18 contiguous registers.

Write-back registers

The processor is equipped with three write-back units WB0(0x0100), WB1(0x0200), and WB2(0x0300), that can be configured independently by three similar blocks. Each register block consists of a unique set of the following registers. In the description, WBx means either WB0, WB1, or WB2 depending on the base address. In the addresses, the underscore means the differentiating nibble between the WBs, for example, 0x0_1C is 0x011C for WB0 0x021C for WB1 and 0x031C for WB2.

Management registers

The Management Registers covering 0x1000-0x10F0 are all control and configuration registers that are not directly connected to rendering of a frame.

Table 5.1 Pixel Processor Register Map (Base Address = 0xF0000000)

Name	Address	Type	Reset	Description
REND_LIST_ADDR	0x0000	R/W	0x00000000	Renderer List Address
REND_RSW_BASE	0x0004	R/W	0x00000000	Renderer State Word Base Address
REND_VERTEX_BASE	0x0008	R/W	0x00000000	Renderer Vertex Base Address

Renderer List Address Register (REND_LIST_ADDR) 0xF0000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REND_LIST_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REND_LIST_ADDR[15:5]											Reserved				

REND_LIST_ADDR	[31:5]	Rend List Address
N	Start address of the polygon list to use for the frame	

Holds the start address of the polygon list for the current frame

Renderer State Word Base Address Register (REND_RSW_BASE) 0xF0000004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REND_RSW_BASE[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REND_RSW_BASE[15:6]										Reserved					

REND_RSW_BASE	[31:6]	Renderer State Word Base Address
N	Default renderer state word base address	

Holds the default renderer state word base address

Renderer Vertex Base Register (REND_VERTEX_BASE) 0xF0000008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REND_VERTEX_BASE[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REND_VERTEX_BASE[15:6]										Reserved					

REND_VERTEX_BASE	[31:6]	Renderer Vertex Base
N	Default vertex bundles base address	

Holds the default base address for vertex bundles

5.4 Geometry Processor Register Description

Figure 5.3 shows that the geometry configuration register has three memory location:

- The Polygon List Builder Configuration Registers are 64Bytes.
- The Control Registers are 168Bytes.
- The Vertex Shader Registers are 304Bytes.

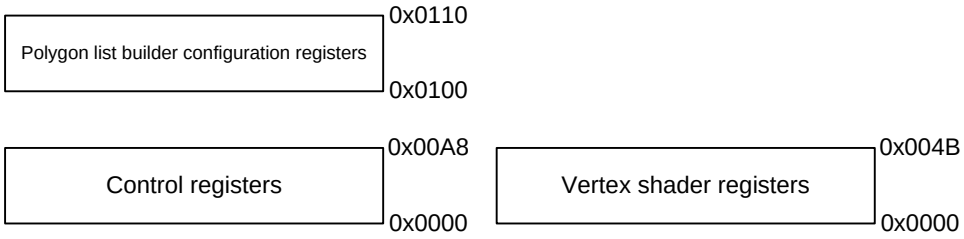


Figure 5.3 Geometry Processor Configuration Register Map

Control registers

The control registers are a set of registers that monitor and control the operation of the geometry processor. The APB bus accesses and controls these registers.

Polygon list builder configuration registers

The PLB configuration registers are a set of registers that control the operation of the polygon list builder, but can only be written from the PLB command list.

Vertex shader registers

The vertex shader configuration registers are s set of registers that control the operation of vertex shader, but can only be written from the vertex shader command list.

To use Polygon list builder configuration registers and Vertex shader registers:

1. Create a command list.
2. Put the command list into memory.
3. Enable end of command list interrupt.
4. Use the APB bus to set the start and end of the command list.
5. Start the command list processor by writing to the Command Register in the APB bus.
6. Wait for interrupt.

Table 5.2 Geometry Processor Control Register Map (Base Address = 0xF0000000)

Name	Address	Type	Reset	Description
CONTR_REG_VSCL_START_ADDR	0x2000	R/W	0x00000000	Control Register VSCL Start Address
CONTR_REG_VSCL_END_ADDR	0x2004	R/W	0x00000000	Control Register VSCL End Address
CONTR_REG_PLBCL_START_ADDR	0x2008	R/W	0x00000000	Control Register PLBCL Start Address
CONTR_REG_PLBCL_END_ADDR	0x200C	R/W	0x00000000	Control Register PLBCL End Address
CONTR_REG_PLB_ALLOC_START_ADDR	0x2010	R/W	0x00000000	Control Register PLB Allocate Start Address
CONTR_REG_PLB_ALLOC_END_ADDR	0x2014	R/W	0x00000000	Control Register PLB Allocate End Address

Table 5.3 Geometry Processor PLB Configuration Register Map

Name	Address	Type	Reset	Description
PLB_CONF_REG_VERTEX_ARRAY_ADDR	0x0100	W	0x00000000	PLB Configuration Register Vertex Array Address
PLB_CONF_REG_INDEX_ARRAY_ADDR	0x0101	W	0x00000000	PLB Configuration Register Index Array Address
PLB_CONF_REG_POINT_SIZE_ADDR	0x0102	W	0x00000000	PLB Configuration Register Point Size Address
PLB_CONF_REG_HEAP_START_ADDR	0x0103	W	0x00000000	PLB Configuration Register Heap Start Address
PLB_CONF_REG_HEAP_END_ADDR	0x0104	W	0x00000000	PLB Configuration Register Heap End Address

Table 5.4 Geometry Processor Vertex Shader Register Map

Name	Address	Type	Reset	Description
VS_CONF_REG_INP_ADDR	0x0000-0x001E	W	0x00000000	VS Configuration Register Input Address
VS_CONF_REG_INP_SPEC	0x0001-0x001F	W	0x0000003F	VS Configuration Register Input Specifier
VS_CONF_REG_OUTP_ADDR	0x0020-0x003E	W	0x00000000	VS Configuration Register Output Address
VS_CONF_REG_OUTP_SPEC	0x0021-0x003F	W	0x0000003F	VS Configuration Register Output Specifier

Control Register VSCL Start Address Register (CONRT_REG_VSCL_START_ADDR) 0xF0002000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VSCL_START_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VSCL_START_ADDR[15:3]													Reserved		

VSCL_START_ADDR	[31:3]	VSCL Start Address
N	Start address of the VSCL	

Reading from the CONTR_REG_VSCL_START_ADDR Register at 0x0000 returns the current command list item being processed. You can read the actual register value from register 0x0080.

Writing to the CONTR_REG_VSCL_START_ADDR Register sets the start address of the vertex shader command list. The value is not visible until the vertex shader is started.

Control Register VSCL End Address Register (CONTR_REG_VSCL_END_ADDR) 0xF0002004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VSCL_END_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VSCL_END_ADDR[15:3]													Reserved		

VSCL_END_ADDR	[31:3]	VSCL End Address
N	End Address of the VSCL	

The CONTR_REG_VSCL_END_ADDR Register provides the end address of the VSCL.

Control Register PLBCL Start Address Register (CONTR_REG_PLBCL_START_ADDR) 0xF0002008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLBCL_START_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PLBCL_START_ADDR[15:3]													Reserved		

PLBCL_START_ADDR	[31:3]	PLBCL Start Address
N	Start address of the PLBCL	

Reading from CONTR_REG_PLBCL_START_ADDR at 0x0008 returns the current command list item being processed. You can read the actual register value from register 0x0084.

Writing to the CONTR_REG_PLBCL_START_ADDR Register sets the start address of the PLB command list. The value is not visible until the vertex shader is started.

Control Register PLBCL End Address Register (CONTR_REG_PLBCL_END_ADDR) 0xF000200C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLBCL_END_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PLBCL_END_ADDR[15:3]													Reserved		

PLBCL_END_ADDR	[31:3]	PLBCL End Address
N	Start address of the VSCL	

The CONTR_REG_PLBCL_END_ADDR Register provides the end address of the PLBCL. When the PLBCL execution reaches the end address, the command list processing terminates, and the IRQ_PLB_END_CMD_LIST interrupt is asserted. The command at this address is not executed.

Control Register PLB Allocation Start Address Register (CONTR_REG_PLB_ALLOC_START_ADDR) 0xF0002010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLB_ALLOC_START_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PLB_ALLOC_START_ADDR[15:7]									Reserved						

PLB_ALLOC_START_ADDR	[31:7]	PLB Allocate Start Address
N	Start/current address for the polygon list allocation	

Writing to the CONTR_REG_PLB_ALLOC_START_ADDR Register sets the start address for polygon list allocation. The register does not show the new value until the CMD_UPDATE_PLB_ALLOC command is given through the CONTR_REG_CMD Register.

Reading from this register returns the current address for polygon list allocation.

Control Register PLB Allocation End Address Register (CONTR_REG_PLB_ALLOC_END_ADDR) 0xF0002014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLB_ALLOC_END_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PLB_ALLOC_END_ADDR[15:7]									Reserved						

PLB_ALLOC_END_ADDR	[31:7]	PLB Allocate End Address
N	End address for the polygon list allocation	

The CONTR_REG_PLB_ALLOC_END_ADDR Register provides the end address for polygon list allocation. If the current address for polygon list allocation reaches this address, the PLB is stalled and the IRQ_PLB_OUT_OF_MEM interrupt is asserted. The register does not show the new value until the CMD_UPDATE_PLB_ALLOC command is given through the CONTR_REG_CMD Register.

PLB Configuration Register Vertex Array Address Register (PLB_CONF_REG_VERTEX_ARRAY_ADDR) 0x0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VERTEX_ARRAY_ADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERTEX_ARRAY_ADDR[15:0]															

VERTEX_ARRAY_ADDR	[31:0]	Vertex Array Address
N	Vertex data array base address	

The PLB_CONF_REG_VERTEX_ARRAY_ADDR Register is write-only. The vertex data array base address, used for glDrawElements mode, is written to this register.

PLB Configuration Register Index Array Address Register (PLB_CONF_REG_INDEX_ARRAY_ADDR) 0x0101

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INDEX_ARRAY_ADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDEX_ARRAY_ADDR[15:0]															

INDEX_ARRAY_ADDR	[31:0]	Index Array Address
N	Vertex index array base address	

The vertex index array base address, used for glDrawElements mode, is written to the PLB_CONF_REG_INDEX_ARRAY_ADDR

PLB Configuration Register Point Size Address Register (PLB_CONF_REG_POINT_SIZE_ADDR) 0x0102

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POINT_SIZE_ADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POINT_SIZE_ADDR[15:0]															

POINT_SIZE_ADDR	[31:0]	Point Size Address
N	Point size address	

The point size address, used for glDrawElements mode, is written to the PLB_CONF_REG_POINT_SIZE_ADDR Register.

PLB Configuration Register Heap Start Address Register (PLB_CONF_REG_HEAP_START_ADDR) 0x0103

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HEAP_START_ADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HEAP_START_ADDR[15:7]									Reserved						

HEAP_START_ADDR	[31:7]	Heap Start Address
N	Polygon list heap start address	

The PLB_CONF_REG_VERTEX_ARRAY_ADDR Register is write-only. The vertex data array base address, used for glDrawElements mode, is written to this register.

PLB Configuration Register Heap End Address Register (PLB_CONF_REG_HEAP_END_ADDR) 0x0104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HEAP_END_ADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HEAP_END_ADDR[15:7]									Reserved						

HEAP_END_ADDR	[31:7]	Heap End Address
N	Polygon list heap end address	

The polygon list heap end address is written to the PLB_CONF_HEAP_END_ADDR Register.

VS Configuration Register Input Address(X) Register (VS_CONF_REG_INP_ADDR) 0x0000+(2*X)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INP_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INP_ADDR[15:0]															

INP_ADDR	[31:0]	Input Address
N	Base address of vertex array	

The VS_CONF_REG_INP_ADDR(X) Register contains the base data address for input data stream X.

VS Configuration Register Input Specifier(X) Register (VS_CONF_REG_INP_SPEC) 0x0001+(2*X)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BE	STRIDE[30:16]														
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STRIDE[15:11]					COMMA[10:6]					VERTEX_DATA_FORMAT[5:0]					

BE	[31]	Big Endian
0	The value is stored in little-endian byte order, and is converted on loading.	
1	The value is stored in big-endian byte order, and is converted on loading.	

STRIDE	[30:11]	Stride
N	Value in 1-byte unit. The stride is defined as the number of bytes separating two consecutive vertices.	

COMMA	[10:6]	Comma
N	Comma location, for fixed point formats only.	

VERTEX_DATA_FORMAT	[5:0]	Vertex Data Format
N	See Table 5.5 for vertex data format value. The reset value is 0x3F	

The PLB_CONF_REG_Z_FAR Register holds the far Z-plane position in FP32 format.

Table 5.5 VS_CONF_REG_INP_SPEC Vertex Data Format

Value	Name	Format
[63]	VS_VSTREAM_NO_DATA	No data
[62:60]	Unused	Reserved
[59:56]	VS_VSTREAM_FORMAT_1_NORM_U32 VS_VSTREAM_FORMAT_4_NORM_U32	1-4 32-bit unsigned normalized
[55:52]	VS_VSTREAM_FORMAT_1_NORM_S32 VS_VSTREAM_FORMAT_4_NORM_S32	1-4 32-bit signed normalized
[51:48]	VS_VSTREAM_FORMAT_1_FP24 VS_VSTREAM_FORMAT_4_FP24	1-4 floats, FP24
[47:44]	VS_VSTREAM_FORMAT_1_NORM_U16 VS_VSTREAM_FORMAT_4_NORM_U16	1-4 16-bit unsigned normalized
[43:40]	VS_VSTREAM_FORMAT_1_NORM_S16 VS_VSTREAM_FORMAT_4_NORM_S16	1-4 16-bit signed normalized
[39:36]	VS_VSTREAM_FORMAT_1_NORM_U8 VS_VSTREAM_FORMAT_4_NORM_U8	1-4 8-bit unsigned normalized
[35:32]	VS_VSTREAM_FORMAT_1_NORM_S8 VS_VSTREAM_FORMAT_4_NORM_S8	1-4 8-bit signed normalized
[31:28]	VS_VSTREAM_FORMAT_1_FIX_U8 VS_VSTREAM_FORMAT_4_FIX_U8	1-4 8-bit fixed point, unsigned
[27:24]	VS_VSTREAM_FORMAT_1_FIX_S8 VS_VSTREAM_FORMAT_4_FIX_S8	1-4 8-bit fixed point, unsigned
[23:20]	VS_VSTREAM_FORMAT_1_FIX_U16 VS_VSTREAM_FORMAT_4_FIX_U16	1-4 16-bit fixed point, unsigned
[19:16]	VS_VSTREAM_FORMAT_1_FIX_S16 VS_VSTREAM_FORMAT_4_FIX_S16	1-4 16-bit fixed point, signed
[15:12]	VS_VSTREAM_FORMAT_1_FP16 VS_VSTREAM_FORMAT_4_FP16	1-4 floats, FP16
[11:8]	VS_VSTREAM_FORMAT_1_FIX_U32 VS_VSTREAM_FORMAT_4_FIX_U32	1-4 32-bit fixed point, unsigned
[7:4]	VS_VSTREAM_FORMAT_1_FIX_S32 VS_VSTREAM_FORMAT_4_FIX_S32	1-4 32-bit fixed point, signed
[3:0]	VS_VSTREAM_FORMAT_1_FP32 VS_VSTREAM_FORMAT_4_FP32	1-4 floats, FP32

VS Configuration Register Output Specifier(X) Register (VS_CONF_REG_OUTP_SPEC) 0x0020+(2*X)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUTP_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OUTP_ADDR[15:0]															

OUTP_ADDR	[31:0]	Output Address
N	Base address of the output vertex stream specifier	

The VS_CONF_REG_OUTP_ADDR(X) Register contains the base data address for output data stream X.

VS Configuration Register Output Specifier(X) Register (VS_CONF_REG_OUTP_SPEC) 0x0021+(2*X)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BE	STRIDE[30:16]														
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STRIDE[15:11]					COMMA[10:6]					VERTEX_DATA_FORMAT[5:0]					

BE	[31]	Big Endian
0	The value is stored in little-endian byte order, and is converted on loading.	
1	The value is stored in big-endian byte order, and is converted on loading.	

STRIDE	[30:11]	Stride
N	Value in 1-byte unit. The stride is defined as the number of bytes separating two consecutive vertices.	

COMMA	[10:6]	Comma
N	Comma location, for fixed point formats only.	

VERTEX_DATA_FORMAT	[5:0]	Vertex Data Format
N	See Table 5.6 for vertex data format value. The reset value is 0x3F	

The PLB_CONF_REG_Z_FAR Register holds the far Z-plane position in FP32 format.

Table 5.6 VS_CONF_REG_OUTP_SPEC Vertex Data Format

Value	Name	Format
[63]	VS_VSTREAM_NO_DATA	No data
[62:53]	Unused	Reserved
[51:48]	VS_VSTREAM_FORMAT_FP24 VS_VSTREAM_FORMAT_FP24	1-4 floats, FP24
[47]	Unused	Reserved
[46]	VS_VSTREAM_FORMAT_RGB565	RGB 565, output only
[45:34]	Unused	Reserved
[33]	VS_VSTREAM_FORMAT_POINT_SIZE	Point size in FP32 format, output only
[32]	VS_VSTREAM_FORMAT_VDATA_BLOCK(XYZW)	Vertex co-ordinates in FP32 format, output only
[31:28]	VS_VSTREAM_FORMAT_1_FIX_U8 VS_VSTREAM_FORMAT_4_FIX_U8	1-4 8-bit fixed point, unsigned
[27:24]	VS_VSTREAM_FORMAT_1_FIX_S8 VS_VSTREAM_FORMAT_4_FIX_S8	1-4 8-bit fixed point, unsigned
[23:20]	VS_VSTREAM_FORMAT_1_FIX_U16 VS_VSTREAM_FORMAT_4_FIX_U16	1-4 16-bit fixed point, unsigned
[19:16]	VS_VSTREAM_FORMAT_1_FIX_S16 VS_VSTREAM_FORMAT_4_FIX_S16	1-4 16-bit fixed point, signed
[15:12]	VS_VSTREAM_FORMAT_1_FP16 VS_VSTREAM_FORMAT_4_FP16	1-4 floats, FP16
[11:8]	VS_VSTREAM_FORMAT_1_FIX_U32 VS_VSTREAM_FORMAT_4_FIX_U32	1-4 32-bit fixed point, unsigned
[7:4]	VS_VSTREAM_FORMAT_1_FIX_S32 VS_VSTREAM_FORMAT_4_FIX_S32	1-4 32-bit fixed point, signed
[3:0]	VS_VSTREAM_FORMAT_1_FP32 VS_VSTREAM_FORMAT_4_FP32	1-4 floats, FP32

5.5 MMU configuration Register Description

Table 5.7 MMU Configuration Register Map (Base Address = 0xF0000000)

Name	Address	Type	Reset	Description
MMU_DTE_ADDR	0x3000	R/W	0x00000000	MMU Current Page Table Address

MMU Current Page Table Address Register (MMU_DTE_ADDR) 0xF0003000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTE_ADDR[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DTE_ADDR[15:0]															

DTE_ADDR	[31:0]	DTE Address
N	Value	

The MMU_DTE_ADDR Register contains the base address of the current page tables. The address must be 4KB aligned. This register can only be written when the MMU is idle or stalled.

6 GRPBUS Configuration

The GRPBUS Configuration block has several registers named as PWRDOWN, SWRESET, GPU_IDLE, BWRAPCTRL.

Table 6.1 GRPBUS Configuration Register Map (Base Address = 0xF0004000)

Name	Address	Type	Reset	Description
GRPBUS_PWRDOWN	0x0000	R/W	0x00000000	Graphics bus power down
GRPBUS_SWRESET	0x0004	R/W	0x00000000	Graphics bus software reset
GRPBUS_GPU_IDLE	0x0008	R/W	0x00000003	GPU idle configuration

Table 6.2 GRPBUS BWRAP Register Map (Base Address = 0xF0005000)

Name	Address	Type	Reset	Description
GRPBUS_BWRAPCTRL	0x0000	R/W	0x00000000	Graphics bus bwrap control

Graphic Bus Power Down Register (GRPBUS_PWRDOWN) 0xF0004000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														OM	GPU

OM	[1]	Overlay Mixer
0	Overlay mixer normal operation	
1	Overlay mixer power down	

GPU	[0]	3D GPU
0	3D GPU normal operation	
1	3D GPU power down	

Graphic Bus Software Reset Register (GRPBUS_SWRESET) 0xF0004004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														OM	GPU

OM	[1]	Overlay Mixer
0	Overlay mixer keep working	
1	Overlay mixer software reset active state	

GPU	[0]	3D GPU
0	3D GPU keep working	
1	3D GPU software reset active state	

GPU Idle Configuration Register (GRPBUS_GPU_IDLE) 0xF0004008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													BOTH	GP	PP

BOTH	[2]	3D GPU Idle Both
0	The clock enable of Geometry Processor and Pixel Processor is determined from each bit of GP and PP.	
1	If IDLE signals of Geometry Processor and Pixel Processor are high, each clock signal will be disabled.	

GP	[1]	3D Geometry Processor Idle
0	The clock signal is enabled regardless of IDLE signal of Geometry Processor. If BOTH is high, this field will be ignored.	
1	The clock signal is disabled regardless of IDLE signal of Geometry Processor. If BOTH is high, this field will be ignored.	

PP	[0]	3D Pixel Processor Idle
0	The clock signal is enabled regardless of IDLE signal of Pixel Processor. If BOTH is high, this field will be ignored.	
1	The clock signal is disabled regardless of IDLE signal of Pixel Processor. If BOTH is high, this field will be ignored.	

This register is for IDLE configuration of 3D GPU. Geometry Processor and Pixel Processor has it's own IDLE bit, which indicates whether the status is in IDLE or WORKING. And each bit can enable or disable the internal clock signal.

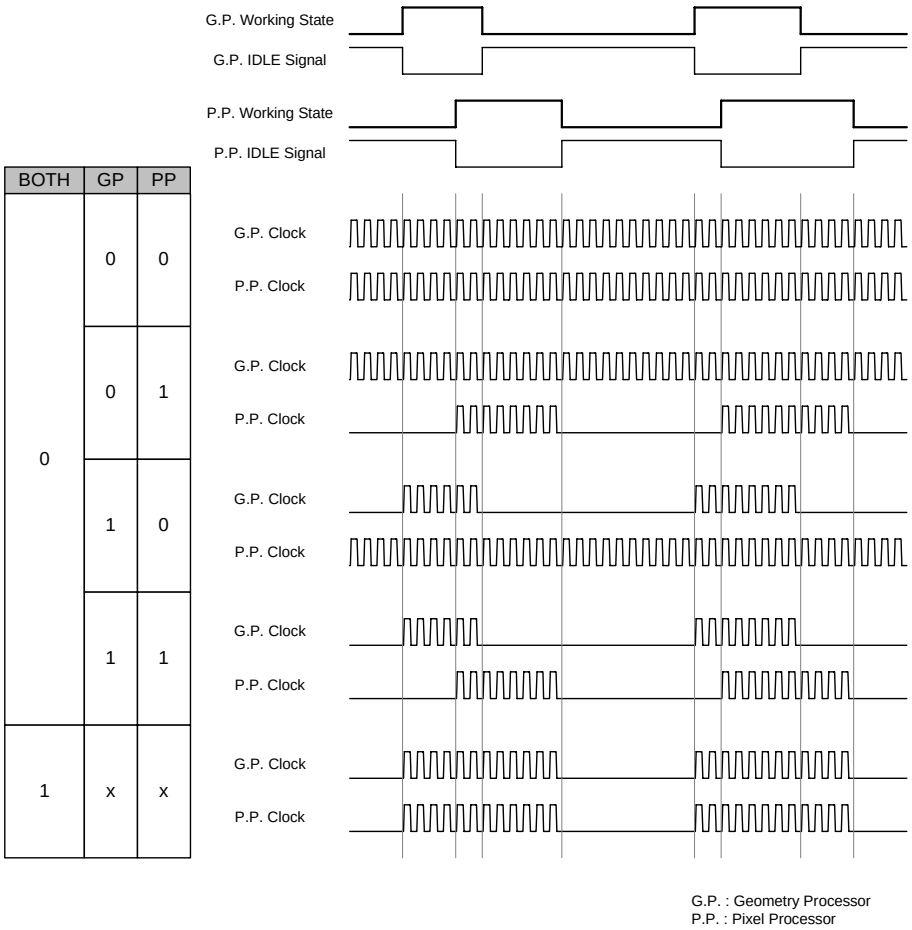


Figure 6.1 GPU Idle Configuration Register and Clock

Graphic Bus BWRAP Control Register (GRPBUS_BWRAPCTRL) **0xF0005000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved														GPU	OM

GPU	[1]	3D GPU BWRAP Control
0	3D GPU BWRAP off	
1	3D GPU BWRAP on	

OM	[0]	Overlay Mixer BWRAP Control
0	Overlay mixer BWRAP off	
1	Overlay mixer BWRAP on	