

OUTROS LIVROS DA PETIT

ELETRÔNICA DE VIDEOGAMES
Teoria e Manutenção

MANUTENÇÃO DE MICROCOMPUTADORES

ELETRÔNICA DIGITAL
Teoria e Aplicação

ELETRÔNICA BÁSICA
Teoria e Prática

SILK SCREEN

RÁDIO
Teoria e Técnicas de Consertos

CONSTRUA SEU COMPUTADOR POR MEIO
SALÁRIO MÍNIMO

TELEVISÃO
Teoria e Consertos — Cores/PB

AUTOMÓVEIS
Guia de pequenos consertos

TV — CORES E PB — CONSERTOS

VÍDEO-CASSETTE
Teoria e Assistência Técnica

CIRCUITOS DE MICROS
Apple — TK — CP — IBM/PC

petit®

Petit Editora e Marketing Direto Ltda.

Av. Brig. Luiz Antonio, 383-2 - and. - cj. 208 - fone: (011) 36-7597 - São Paulo - SP

Petit

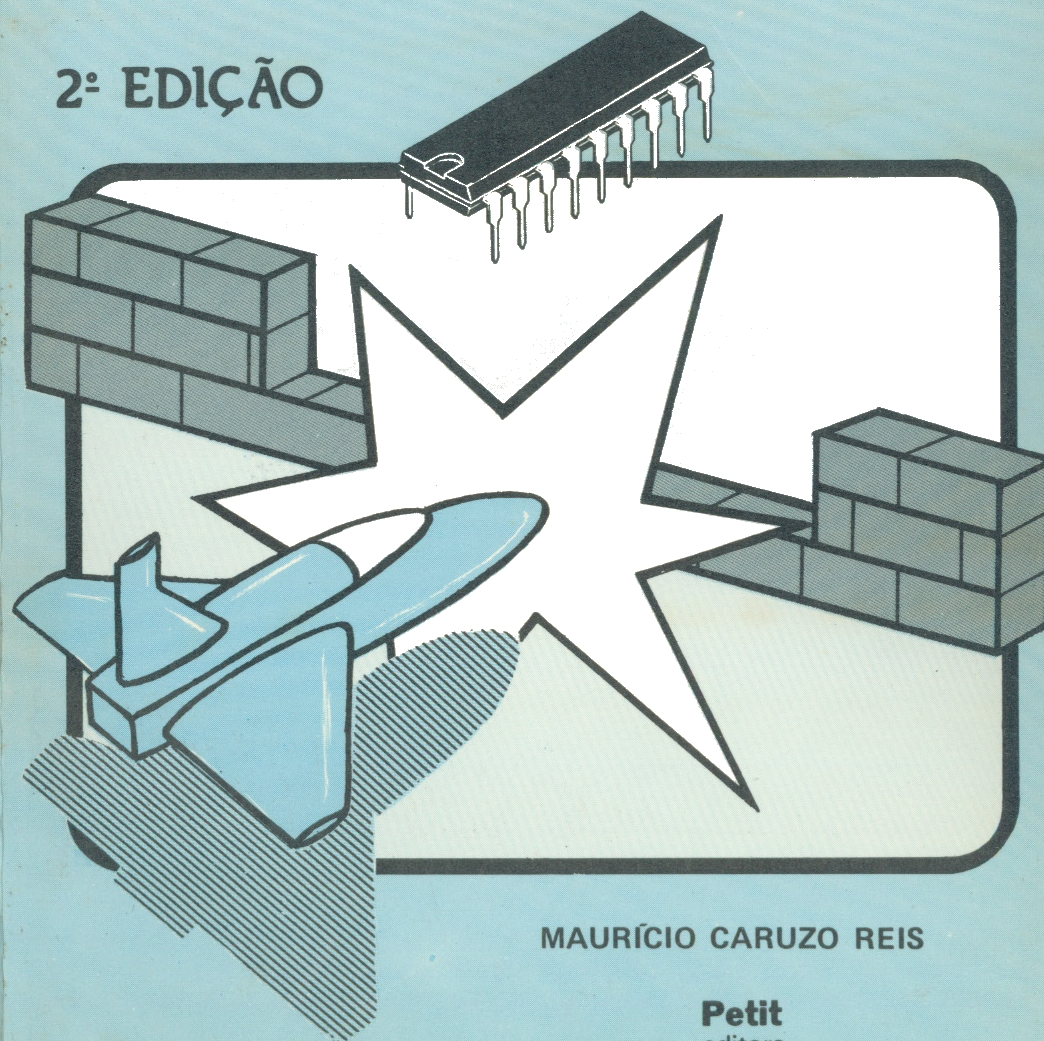
ELETRÔNICA DE VIDEOGAMES

Maurício Caruzo Reis

ELETRÔNICA DE VÍDEOGAMES

TEORIA, PROGRAMAÇÃO E
MANUTENÇÃO

2ª EDIÇÃO



MAURÍCIO CARUZO REIS

Petit
editora

ELETRÔNICA DE VÍDEOGAMES

TEORIA, PROGRAMAÇÃO E MANUTENÇÃO

MAURÍCIO CARUZO REIS

Dados de Catalogação na Publicação (CIP) internacional
(Câmara Brasileira do Livro, SP, Brasil)

Reis, Mauricio Caruso, 1950-
R311e Eletrônica de videogames : teoria, programação
e manutenção / Mauricio Caruso Reis. -- São Pau-
lo : Petit, 1986.

1. Jogos eletrônicos 2. Programas de computa-
dor I. Título.

86-1346

CDD-621.381
-001.6425

Índices para catálogo sistemático:

1. Jogos : Computadores : Programas : Processamento
de dados 001.6425
2. Jogos de vídeo : Computadores : Programas : Proces-
samento de dados 001.6425
3. Jogos eletrônicos : Engenharia 621.381
4. Programas : Computadores : Processamento de dados
001.6425

PETIT

Editora e Marketing Direto Ltda.
Av. Brig. Luiz Antonio, 383 - 2º
São Paulo - Capital
Cep 01317

TODOS OS DIREITOS RESERVADOS

É expressamente proibida a reprodução parcial ou integral sem a autorização
do editor

ÍNDICE

INTRODUÇÃO	9
CAPÍTULO 1 – TÓPICOS DE INTERESSE EM ELETRÔNICA DIGITAL	12
– SINAL DIGITAL	
– FUNÇÕES LÓGICAS	
– NUMERAÇÃO DIGITAL	
– TECNOLOGIA DE CIs	
– MEMÓRIAS SEMICONDUTORAS	
– MULTIPLEXAGEM	
– LATCH	
– BUFFER	
– FLIP-FLOP	
CAPÍTULO 2 – ARQUITETURA DE VÍDEO-GAMES	31
CAPÍTULO 3 – MICROPROCESSADORES	35
– EVOLUÇÃO DOS MICROPROCESSADORES	
– ARQUITETURA DO MICROPROCESSADORES	
– TEMPORIZAÇÃO DO MICROPROCESSADOR	
– BARRAS DE ENDEREÇO, DADOS E CONTROLES	
CAPÍTULO 4 – CARTUCHO	48
– PROGRAMA	
– NÍVEIS DE LINGUAGEM	
– PROGRAMAÇÃO PARA VÍDEO-GAMES	
– RETIRADA DA ROM DO CARTUCHO	
CAPÍTULO 5 – ATUADORES; JOYSTICK	58
– INTERFACEAMENTO DE ATUADORES	
– JOYSTICK	
– TECHADO (KEYBOARD)	
CAPÍTULO 6 – TELEVISÃO PRETO E BRANCO	68
– A FORMAÇÃO DA IMAGEM NA TELA	
– SINAL COMPOSTO DE VÍDEO	
– MODULAÇÃO DO SINAL COMPOSTO DE VÍDEO	
– RECEPTOR PRETO E BRANCO	

CAPÍTULO 7 – TELEVISÃO A CORES 84

- PM – MODULAÇÃO DE FASE
- TRANSMISSÃO A CORES
- RECEPTOR TVC
- SISTEMA PAL

CAPÍTULO 8 – GERAÇÃO DE IMAGEM E SOM 103

- MAPEAMENTO DA TELA
- CÉLULAS E FIGURAS
- PIXEL
- ROM GERADOR GRÁFICO
- ESTÁGIO DE GERAÇÃO DE IMAGEM
- PROGRAMAÇÃO DE CENAS
- CODIFICAÇÃO DA COR
- CONVERSOR PAL
- EFEITOS SONOROS

CAPÍTULO 9 – ATARI 131

- ESQUEMA GERAL
- MICROPROCESSADOR 6507/10745
- CIRCUITO DE RESET
- FONTE REGULADA
- CARTUCHO
- PIA (INTERFACE ADAPTADORA DE PERIFÉRICOS)
- CHAVES DE CONTROLE
- JOYSTICK
- TIA (INTERFACE ADAPTADORA DE TELEVISÃO)
- CIRCUITO DE CLOCK
- CONVERSOR PAL
- OSCILADOR DE ÁUDIO
- MODULADOR DE RF

CAPÍTULO 10 – ODISSEY 155

- CONSOLE
- ESQUEMA GERAL
- FONTE REGULADA
- CLOCK
- MICROPROCESSADOR 8048
- RAM 6810
- ROM – CARTUCHO
- TECLADO (KEYBOARD)
- JOYSTICK
- DISPLAY PLÁSTICO
- CODIFICADOR DE COR
- CHAVE PAL
- MODULADOR DE RF

CAPÍTULO 11 – TÉCNICAS E INSTRUMENTOS PARA MANUTENÇÃO DE VÍDEO-GAMES 191

- ABORDAGEM DO DEFEITO
- INSTRUMENTAÇÃO PARA CONserto
- MANUTENÇÃO ATARI
- MANUTENÇÃO ODISSEY

APÊNDICE A – ASSEMBLER/CÓDIGO DE MÁQUINA DO 6507 209

APÊNDICE B – ASSEMBLER/CÓDIGO DE MÁQUINA DO 8048 221

Introdução

Durante umas três décadas o receptor de televisão ficou na sala de estar unicamente para os programas transmitidos por umas poucas emissoras. Educadores, psicólogos, sociólogos, políticos, pessoal que habitualmente diverge em todos os assuntos, concordavam num ponto: a televisão era um grande mal social. Uma das bases deste julgamento consistia na passividade propiciada por ela, milhões de pessoas a ceitando passivamente o que lhes era "empurrado".

A invenção dos microcomputadores no começo da década de 1970 trouxe consigo a primeira alteração nesse estado de coisas, já que punha à disposição dos usuários os meios concretos, facilmente acessíveis (financeira e tecnicamente), para interagir com o televisor. Além do mais levou à criação e aperfeiçoamento de grande número de dispositivos eletrônicos dedicados ao vídeo.

Surge então o vídeo-game, filho mestiço da televisão com o microcomputador. Na verdade, um microcomputador especializado em jogos para vídeo (do nome inglês "game", jogo). Para muitos, inclusive técnicos, esta é a primeira surpresa: constatar que no console daquele aparelho usado como brincadeira até por crianças existe um computador completo, com todos seus estágios.

A diferença entre microcomputadores e

vídeos-games pode ser resumida em 3 pontos:

- 1 - vídeo-games tem memórias reduzidas (geralmente 2.000 posições, contra umas 64.000 dos microcomputadores).
- 2 - em microcomputadores o usuário pode entrar com programas através do teclado, utilizando linguagens de alto nível (BASIC, PASCAL, COBOL, etc) - que são fáceis e acessíveis até para pessoas não especializadas; em vídeo-games os programas são introduzidos através de memórias gravadas em fâbricas ("cartuchos"), usando código de máquina - uma linguagem complexa que exige conhecimentos técnicos de eletrônica e informática.
- 3 - todo o circuito de vídeo-game é centralizado para entrada de joystick e saída de vídeo, enquanto microcomputadores podem interagir com um universo de periféricos além desses (impressora, cassette, etc.).

Apesar das diferenças citadas, a semelhança entre micros e vídeo-games é patente. Quem já conhece os princípios de funcionamento de um terá enorme facilidade de familiarizar-se com o outro. Isso vale inclusive para técnicas e instrumentos de conserto. De resto, já existem em São Paulo oficinas especializadas na conversão de vídeo-game em microcomputador.

O ATARI é o mais famoso dos vídeo-games, sendo modelo para um sem número de cópias, imitações e versões. Foi lançado nos Estados Unidos em 1977 e é fabricado no Brasil pela POLYVOX (GRADIEN-TE) sob licença. O ODISSEY leva o nome PHILIPS, cuja marca é garantia de sucesso no mundo inteiro. Esses dois vídeo-games dominam o mercado, e certamente

quem os conhecer bem terá amplas condições de entender, manipular e consertar outros.

CAPÍTULO 1 – TÓPICOS DE INTERESSE EM ELETRÔNICA DIGITAL

1 - SINAL DIGITAL

Dispositivos digitais reconhecem apenas dois sinais: um com voltagem em nível alto, geralmente acima de 5 Volts (representado por H de "high" - alto, ou por 1); outro com voltagem em nível baixo sendo comum o aterramento 0 Volt (representado por L de "low" - baixo, ou por 0).

O sinal H é visto em gráficos como parte superior da onda quadrada, e o L como parte inferior. Na figura 1-1 são representados os dois níveis.

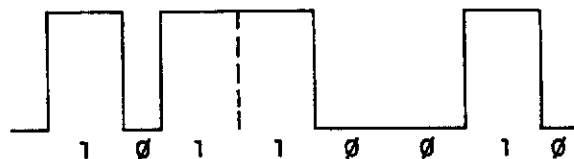


figura 1-1 - Sinal digital

Existe ainda um terceiro estado - TRISTATE - que é exatamente a ausência de níveis lógicos. Na condição de tristate o terminal se comporta como se estivesse fora do circuito, não participando das

operações.

Chamamos BIT cada sinal lógico; uma sequência de 8 bits é denominado BYTE (pronuncia-se BÁITE). Portanto, na figura 1-1 temos 1 byte, que pode ser escrito 10110010.

Quando um dispositivo digital necessita da aplicação de sinal em nível H para funcionar dizemos que é LÓGICA POSITIVA ou ATIVO ALTO. A LÓGICA NEGATIVA ou ATIVO BAIXO, ocorre na aplicação de sinal nível baixo L para por em atividade o dispositivo. Este último caso é muito frequente e convencionou-se que uma entrada de lógica negativa tenha uma barra acima de seu símbolo. Assim, o microprocessador 6507 (utilizado na ATARI) tem seu pino 1 representado por $\overline{\text{RES}}$ (de RESET). Quando é aplicado ali um sinal de nível L ele é ativado e reseta, isto é, recomeça o programa. Observe que para se executar normalmente o programa é necessário que esse pino 1 seja mantido em nível H durante todo o tempo.

Graficamente, coloca-se no símbolo do componente uma "bolinha" nos terminais que empregam lógica negativa. Na figura 1-2 temos um caso típico, um CI que precisa do sinal $\overline{\text{CE}}$ em nível L para começar a funcionar. A figura 1-2B é equivalente a 1-2A, bastando colocar a barra em $\overline{\text{CE}}$. Um terminal sem a bolinha ou sem barra na sua indicação usa lógica positiva (figura 1-2C).

2 - FUNÇÕES LÓGICAS

Sinais digitais podem ser combinados como entradas em PORTAS LÓGICAS, e o resultado será uma saída cujo nível obedecerá a regras de funções lógicas. TABELA VERDADE é uma representação gráfica mostrando entradas e saídas para cada uma das funções.

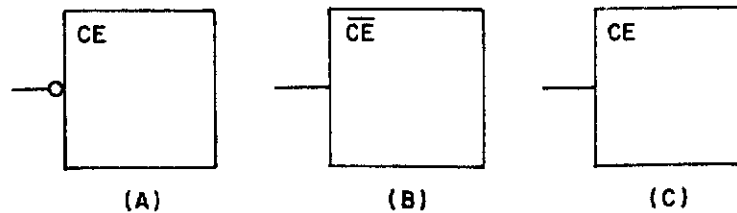


figura 1-2 - Lógica de ativação de terminais (A e B) lógica negativa (C) lógica positiva.

Na função lógica OR (OU) a saída será 1 se qualquer um dos sinais na entrada tiver nível 1. A saída será 0 somente se todas as entradas estiverem em nível 0 (figura 1-3). A tabela verdade da função OR é vista na figura 1-4. Observe que também podemos representá-la por '+'.
 clusivo), representado por $\oplus$, a saída será 1 se as duas entradas estiverem em nível diferente, e será 0 se elas estiverem com o mesmo nível lógico (figura 1-7). Tabela verdade na figura 1-8.

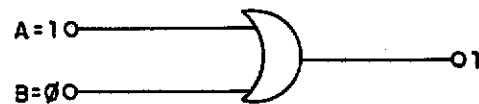


figura 1-3 - Símbolo da função OR.

Na função AND (E), também representada por '.', a saída será 1 se todas as entradas estiverem em nível 1. Se qualquer uma delas estiver em nível 0 a saída será 0 (figura 1-5). A tabela verdade da função AND consta na figura 1-6.

Na função EOR ("exclusive OR", ou ex-

A	B	A + B A OR B
1	1	1
0	1	1
1	0	1
0	0	0

figura 1-4 - Tabela verdade OR.

Na função lógica NOT (NÃO), chamada INVERSORA e indicada por uma barra na entrada, o sinal de entrada aparecerá na saída com o nível lógico trocado (figura 1-9, tabela verdade na figura 1-10).

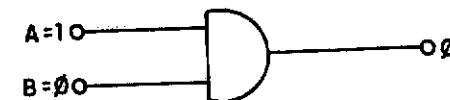


figura 1-5 - Símbolo da função AND

A	B	$A \cdot B$ A AND B
1	1	1
0	1	0
1	0	0
0	0	0

figura 1-6 - Tabela verdade AND



figura 1-7 - Símbolo da função EOR

A	B	$A \oplus B$ A EOR B
1	1	0
0	1	1
1	0	1
0	0	0

figura 1-8 - Tabela verdade EOR.

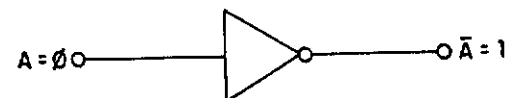


figura 1-9 - Símbolo da função NOT.

A	$\bar{A}$ NOT A
1	0
0	1

figura 1-10 - Tabela verdade NOT.

As funções AND e OR podem ser seguidas de um NOT, formando NAND (NOT AND) e NOR (NOT OR). Neste caso a saída é o nível contrário do que haveria numa função AND ou OR. Nos símbolos acrescenta-se uma bolinha antes da saída (figura 1-11A e 1-11B).

3 - NUMERAÇÃO DIGITAL

A aritmética digital é baseada em numeração binária (1- ou H, e 0-ou L), o que obriga a uma familiaridade muito grande com potências de 2. Lembre-se que 2^3 significa o número 2 multiplicado por si mesmo 3 vezes ($2^3 = 2 \times 2 \times 2 = 8$) e 2^5 é 2 multiplicado por si mesmo 5 vezes ($2^5 = 2 \times 2 \times 2 \times 2 \times 2 = 32$). Na página seguinte apresentamos tabela de potências de 2:

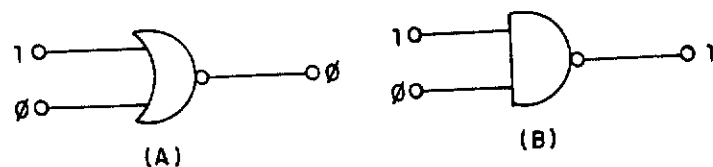


figura 1-11 - (A) Símbolo da função NOR (B) Símbolo da função NAND.

$2^0 = 1$	$2^6 = 64$	$2^{12} = 4.096$
$2^1 = 2$	$2^7 = 128$	$2^{13} = 8.192$
$2^2 = 4$	$2^8 = 256$	$2^{14} = 16.384$
$2^3 = 8$	$2^9 = 512$	$2^{15} = 32.768$
$2^4 = 16$	$2^{10} = 1.024$	$2^{16} = 65.536$
$2^5 = 32$	$2^{11} = 2.048$	

O número $2^{10} = 1024$ é chamado K-KILO, "equivalendo" a mil. Assim, 1 Kilobit = 1.024 bits e 64 Kilobytes = 65.536 bytes. É comum abreviar Kilobyte por KB e Kilobit por Kb (não confunda os dois: 1 Kb = 1.024 bits, 1 KB = 1.024 bytes = 1.024x8 bits).

Podemos expressar qualquer número decimal por soma de potências de 2. Por exemplo:

$$6 = 2^2 + 2^1 \quad 12 = 2^3 + 2^2$$

De uma maneira mais geral escrevemos '1x' as potências que entram na soma e '0x' as potências que não entram. Alguns exemplos:

$$\begin{aligned}
 0 &= (0x2^5) + (0x2^4) + (0x2^3) + (0x2^2) + (0x2^1) + (0x2^0) \\
 1 &= (0x2^5) + (0x2^4) + (0x2^3) + (0x2^2) + (0x2^1) + (1x2^0) \\
 2 &= (0x2^5) + (0x2^4) + (0x2^3) + (0x2^2) + (1x2^1) + (0x2^0) \\
 3 &= (0x2^5) + (0x2^4) + (0x2^3) + (0x2^2) + (1x2^1) + (1x2^0) \\
 4 &= (0x2^5) + (0x2^4) + (0x2^3) + (1x2^2) + (0x2^1) + (0x2^0) \\
 5 &= (0x2^5) + (0x2^4) + (0x2^3) + (1x2^2) + (0x2^1) + (1x2^0) \\
 6 &= (0x2^5) + (0x2^4) + (0x2^3) + (1x2^2) + (1x2^1) + (0x2^0) \\
 7 &= (0x2^5) + (0x2^4) + (0x2^3) + (1x2^2) + (1x2^1) + (1x2^0) \\
 8 &= (0x2^5) + (0x2^4) + (1x2^3) + (0x2^2) + (0x2^1) + (0x2^0) \\
 9 &= (0x2^5) + (0x2^4) + (1x2^3) + (0x2^2) + (0x2^1) + (1x2^0) \\
 10 &= (0x2^5) + (0x2^4) + (1x2^3) + (0x2^2) + (1x2^1) + (0x2^0) \\
 20 &= (0x2^5) + (1x2^4) + (0x2^3) + (1x2^2) + (0x2^1) + (0x2^0) \\
 35 &= (1x2^5) + (0x2^4) + (0x2^3) + (0x2^2) + (1x2^1) + (1x2^0)
 \end{aligned}$$

Convencionando escrever as potências da direita para a esquerda em ordem crescente, como acima, podemos registrar sô os 1s e 0s e subentender o restante. Os exemplos acima ficam:

DECIMAL		BINÁRIO
0	=	000000
1	=	000001
2	=	000010
3	=	000011
4	=	000100
5	=	000101
6	=	000110
7	=	000111
8	=	001000
9	=	001001
10	=	001010

20 = 010100
35 = 100011

O bit que estiver na extrema esquerda é chamado MSB ("most significant bit", bit mais significativo) e na extrema direita LSB ("least significant bit", bit menos significativo). Na figura 1-12, o MSB é 0 e o LSB é 1.

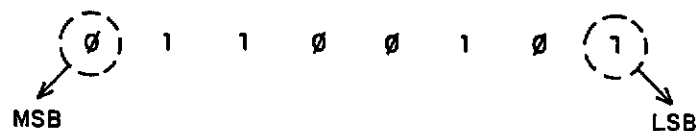


figura 1-12 - Dígito mais significativo - MSB e dígito menos significativo - LSB.

Dispositivos digitais usam esse sistema binário, representando o 1 por sinal em nível alto H e 0 por sinal em nível baixo L. Para nós, entretanto, é bastante tedioso trabalhar com numeração binária quando temos números muito grandes. Utiliza-se então, outros sistemas para representar a linguagem binária do dispositivo.

No código HEXADECIMAL cada grupo de 4 binários é convertido no seu valor de potência de 2, adotando-se 10=A, 11=B, 12=C, 13=D, 14=E, 15=F. Por exemplo:

1010 0010 binário = A2 hexadecimal; os quatro primeiros bits foram convertidos para A e os quatro restantes para 2.

Colocamos índice 2 para binário e ín-

dice 16 para indicar hexadecimal, ou pospomos a letra B aos binários e H aos hexadecimais. Exemplos:

DECIMAL	BINÁRIO	HEXADECIMAL
0	0000 0000 ₂ = 0000 0000B	00 ₁₆ = 00H
1	0000 0001 ₂ = 0000 0001B	01 ₁₆ = 01H
2	0000 0010 ₂ = 0000 0010B	02 ₁₆ = 02H
3	0000 0011 ₂ = 0000 0011B	03 ₁₆ = 03H
4	0000 0100 ₂ = 0000 0100B	04 ₁₆ = 04H
5	0000 0101 ₂ = 0000 0101B	05 ₁₆ = 05H
6	0000 0110 ₂ = 0000 0110B	06 ₁₆ = 06H
7	0000 0111 ₂ = 0000 0111B	07 ₁₆ = 07H
8	0000 1000 ₂ = 0000 1000B	08 ₁₆ = 08H
9	0000 1001 ₂ = 0000 1001B	09 ₁₆ = 09H
10	0000 1010 ₂ = 0000 1010B	0A ₁₆ = 0AH
11	0000 1011 ₂ = 0000 1011B	0B ₁₆ = 0BH
12	0000 1100 ₂ = 0000 1100B	0C ₁₆ = 0CH
13	0000 1101 ₂ = 0000 1101B	0D ₁₆ = 0DH
14	0000 1110 ₂ = 0000 1110B	0E ₁₆ = 0EH
15	0000 1111 ₂ = 0000 1111B	0F ₁₆ = 0FH
16	0001 0000 ₂ = 0001 0000B	10 ₁₆ = 10H
20	0001 0100 ₂ = 0001 0100B	14 ₁₆ = 14H
35	0010 0011 ₂ = 0010 0011B	23 ₁₆ = 23H

No livro usaremos a letra H para indicar hexadecimal e B para binário. Na falta de indicação entender-se-á como decimal.

BCD ("binary coded decimal", decimal codificado em binário) é outro código usual em Eletrônica Digital, no qual cada decimal é representado por quatro bits (em sistema de potências de 2, como nos hexadecimais).

DECIMAL	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	0001 0000
11	0001 0001
20	0010 0000
100	0001 0000 0000
1005	0001 0000 0000 0101

Letras e outros símbolos para redação de textos também possuem vários códigos. Microcomputadores costumam utilizar o código ASCII no teclado e no vídeo, cada caractere (letras, números, comandos) é representado por 7 bits, podendo haver um oitavo de conferência que completa o byte.

O código ASCII completo é apresentado a seguir, em hexadecimal (os primeiros caracteres se referem a comandos).

4 - TECNOLOGIAS DE CIs

As duas famílias lógicas mais usadas em aparelhos digitais são TTL e CMOS. Os CIs tipo TTL têm a numeração começando por 74 (exemplo: no ODISSEY temos os CIs 74LS148, 74LS175, 74LS365, 74LS00, 74LS74 e 74LS156). Os CIs CMOS começam por 40 (exemplos: 4049, 4001), ou 45 (exemplos: 4549, 4501).

Ao manipular um equipamento deve-se tomar o máximo cuidado com CIs CMOS. O simples fato de encostar a mão ou a ferramenta neles pode danificá-los (alguns deles, entretanto, vêm com circuito de proteção, tendo a letra B posposta ao número: 4049B).

Se seu instrumento de teste (por exemplo, ponta de prova lógica) usa tecnologia TTL haverá problemas ao pesquisar um circuito com CI CMOS, pois eles não são compatíveis. Sempre que possível selecione instrumentos compatíveis com as duas famílias e esteja atento a problemas de interfaceamento.

CÓDIGO ASCII

CARACTER	ASCII(H)	CARACTER	ASCII
NUL	00	SP	20
SOH	01	!	21
STX	02	"	22
ETX	03	#	23
EOT	04	\$	24
ENQ	05	%	25

ACK	06	&	26
BEL	07	/	27
BS	08	(	28
HT	09	)	29
LF	0A	*	2A
VT	0B	+	2B
FF	0C	,	2C
CR	0D	-	2D
SO	0E	.	2E
SI	0F	/	2F
DLE	10	0	30
DC1	11	1	31
DC2	12	2	32
DC3	13	3	33
DC4	14	4	34
NAK	15	5	35
SYN	16	6	36
ETB	17	7	37
CAN	18	8	38
EM	19	9	39
SUB	1A	:	3A
ESC	1B	;	3B
FS	1C	<	3C
GS	1D	=	3D
RS	1E	>	3E
US	1F	?	3F

CÓDIGO ASCII (continuação)			
CARACTER	ASCII(H)	CARACTER	ASCII (H)
	40	'	60
A	41	a	61
B	42	b	62
C	43	c	63
D	44	d	64
E	45	e	65
F	46	f	66
G	47	g	67
H	48	h	68
I	49	i	69
J	4A	j	6A
K	4B	k	6B
L	4C	l	6C
M	4D	m	6D
N	4E	n	6E
O	4F	o	6F
P	50	p	70
Q	51	q	71
R	52	r	72
S	53	s	73
T	54	t	74
U	55	u	75
V	56	v	76
W	57	w	77

CARACTER	ASCII(H)	CARACTER	ASCII(H)
X	58	x	78
Y	59	y	79
Z	5A	z	7A
[	5B	{	7B
\	5C		7C
]	5D	}	7D
^ {↑}	5E	~	7E
_ {←}	5F	DEL	7F

5 - MEMÓRIAS SEMICONDUTORAS

Memórias semicondutoras se classificam em ROM ("read only memory", memória de apenas leitura) e RAM ("random access memory", memória de acesso aleatório). A primeira tem dados gravados na fábrica e quando endereçada apresenta-os para leitura. A RAM não tem gravação permanente, sendo possível ao usuário a qualquer instante gravar (escrita) ou fazer a leitura dos dados.

Um tipo especial de ROM é a PROM ("programmable read only memory", ROM programável), que vem "em branco" da fábrica, cabendo ao próprio usuário a gravação. A EPROM ("erasable PROM", PROM apagável) pode ser não apenas gravada pelo consumidor, como posteriormente apagada e regravada. Ao contrário da RAM o apagamento e regravagem da EPROM é demorado e exige equipamentos especiais.

CIs ROM geralmente são de 1, 2, 4, 8, 16, 32 ou 64 Kbytes. 1 Kbyte significa que há 1024 endereços, cada um deles contendo 8 bits (1 byte) de dados. Os 1024 endereços correspondem a 10 linhas de

endereço (A9, A8, A7, A6, A5, A4, A3, A2, A1, A0). O estado lógico do conjunto de linhas forma um número binário igual ao endereço. Por exemplo:

se A9 = 0, A8 = 1, A7 = 1, A6 = 0, A5 = 0, A4 = 0, A3 = 1, A2 = 0, A1 = 0, A0 = 1, temos:

endereço = 0110001001 = 393.

Ao colocar os pinos de endereço do CI nos níveis acima ele apresentará nos terminais de dados os 8 níveis gravados no endereço número 393 (por exemplo: D0=1, D1=1, D2=0, D3=1, D4=0, D5=1, D6=1, D7=0).

6 - MULTIPLEXAGEM

Para economizar pinos e circuitos sistemas microprocessados utilizam o processo de multiplexagem. O CI recebe (ou emite) uma parte dos bits (geralmente endereço) que são armazenados; após um sinal de sincronização é enviado o restante dos bits; as duas partes são então juntadas e processadas. Um caso típico ocorre no ODISSEY que multiplexa endereços e dados para a RAM nos mesmos terminais (figura 1-13). Assim, inicialmente são enviados os 8 bits de endereços através dos pinos 12 a 19, que são armazenados noutro CI e constituem um endereço na RAM; em seguida e através ainda dos pinos 12 a 19 são enviados 8 bits de dados.

7 - LATCH

Vimos que no processo de multiplexagem é necessário guardar os bits da primeira "remessa" até que seja emitida a segunda e as duas juntadas. LATCH é um CI que faz essa função de armazenamento. Mais precisamente ele captura os dados na barra e os mantém em seus terminais de saída.

Um latch bastante usado em circuitos

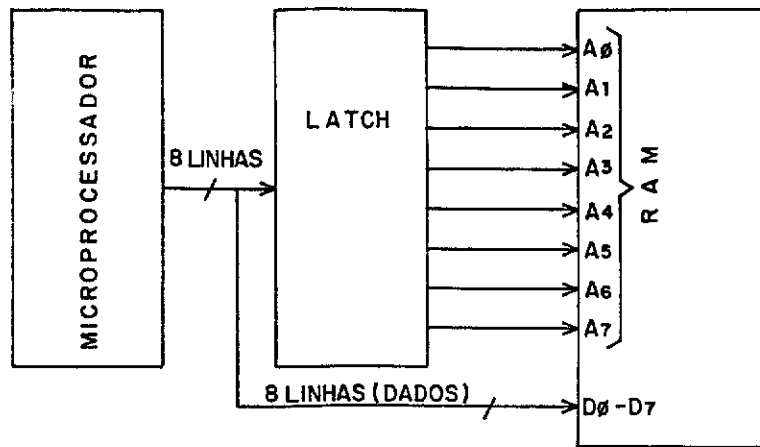


figura 1-13 - Multiplexagem de endereços e dados

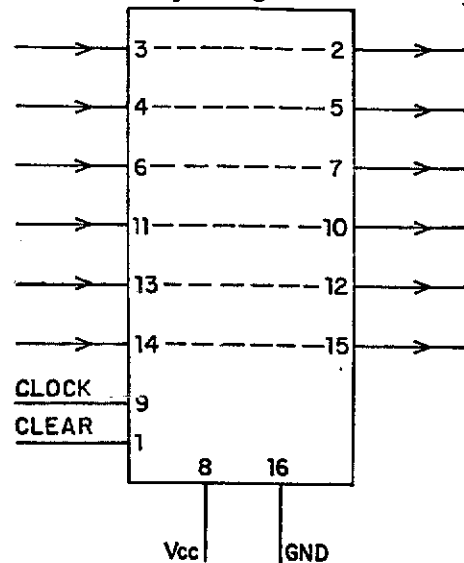


figura 1-14 - Latch 74LS175.

digitais, inclusive no exemplo da figura 1-13 do ODISSEY, é o 74LS175 (figura 1-14). Quando um pulso nível 0 aparece em seu pino 9 (CLOCK) ele dá entrada aos sinais existentes nos pinos 4, 5, 12, 13 e os apresenta por tempo indeterminado nos pinos 2, 7, 10, 15, respectivamente.

8 - BUFFER

O BUFFER é outro CI que recebe um sinal e retransmite-o, mas amplificando a corrente elétrica. Isso permite que um dispositivo com pouca corrente de saída comande uma grande quantidade de cargas. O símbolo do buffer comum é visto na figura 1-15 (não confunda com o inversor que tem uma "bolinha" na ponta).

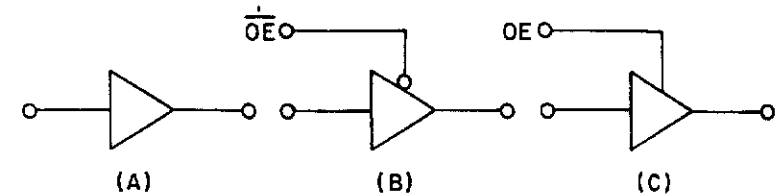


figura 1-15 - (A) símbolo do buffer (B) buffer tristate com $\overline{OE}$ (C) buffer tristate com OE

BUFFER TRISTATE é um modelo especial onde o sinal recebido é retransmitido (amplificado) somente se ocorre também OE em nível 0 (figura 1-15B) ou OE em nível 1 (figura 1-15C). Caso contrário a saída do buffer permanece em tristate, isolando o dispositivo na sua entrada do circuito na saída.

9 - FLIP-FLOP

FLIP-FLOP é a função mais usada em

Eletrônica Digital, mas raramente aparece sozinho. A grande maioria de dispositivos são grandes cadeias de flip-flops, devidamente organizados.

Um tipo comum é apresentado na figura 1-16. Quando o sinal de clock é recebido o flip-flop dá entrada no bit que estiver no pino de dado, armazenando-o e apresentando-o ao terminal Q. No terminal $\bar{Q}$ é apresentado o complemento (inverso) do bit.

Quando Q está em nível 0 (portanto $\bar{Q}$ em 1) dizemos que o flip-flop está em RESET. Com Q=1 (e $\bar{Q}=0$) ele estará em SET. Esta terminologia se estende a todos dispositivos eletrônicos em geral : saída nível 0-resetado, saída nível 1-setado.

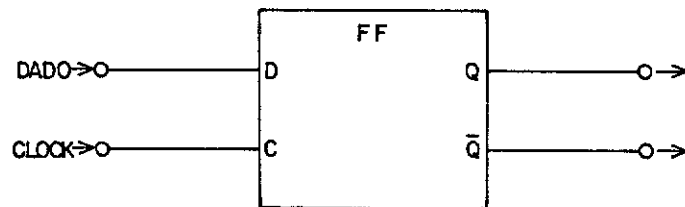


figura 1-16 - Símbolo do flip-flop

Usam-se outros tipos de flip-flop, inclusive com mais de um terminal de entrada de dados (J e K, S e R).

CAPÍTULO 2 – ARQUITETURA DE VÍDEO-GAMES

O esboço da arquitetura de um vídeo-game é apresentado na figura 2-1. Obviamente é bastante genérico, comportando alterações conforme uma marca em particular.

O cérebro do sistema é o MICROPROCESSADOR, que controla todas as operações através de sinais emitidos ou recebidos pelos pinos de endereços, dados ou controles. Esses pinos são ligados ao resto do sistema por trilhas de circuitos impressos que no esquema elétrico são denominadas BARRAS (de endereços, dados ou controles).

O microprocessador segue instruções gravadas no CARTUCHO -que é uma ROM, executando-as uma a uma na ordem em que estão gravadas. Cada jogo (SPACE INVADERS, STAR WAR, etc) tem sua própria sequência de instruções, dependendo dos objetivos e criatividade do seu inventor. Entretanto, apesar de haver variações na sequência as instruções propriamente ditas são fixas e características de cada microprocessador. Por exemplo, o microprocessador 6507 do ATARI tem uma lista de 56 instruções que podem ser combinadas de maneira a formar um jogo.

O microprocessador recebe também dados dos ATUADORES (JOYSTICK, TECLADO), que são comandos disponíveis para os jogadores participarem do jogo. Ao contrário das instruções do cartucho que são executados em sequência, os dados dos atuadores são

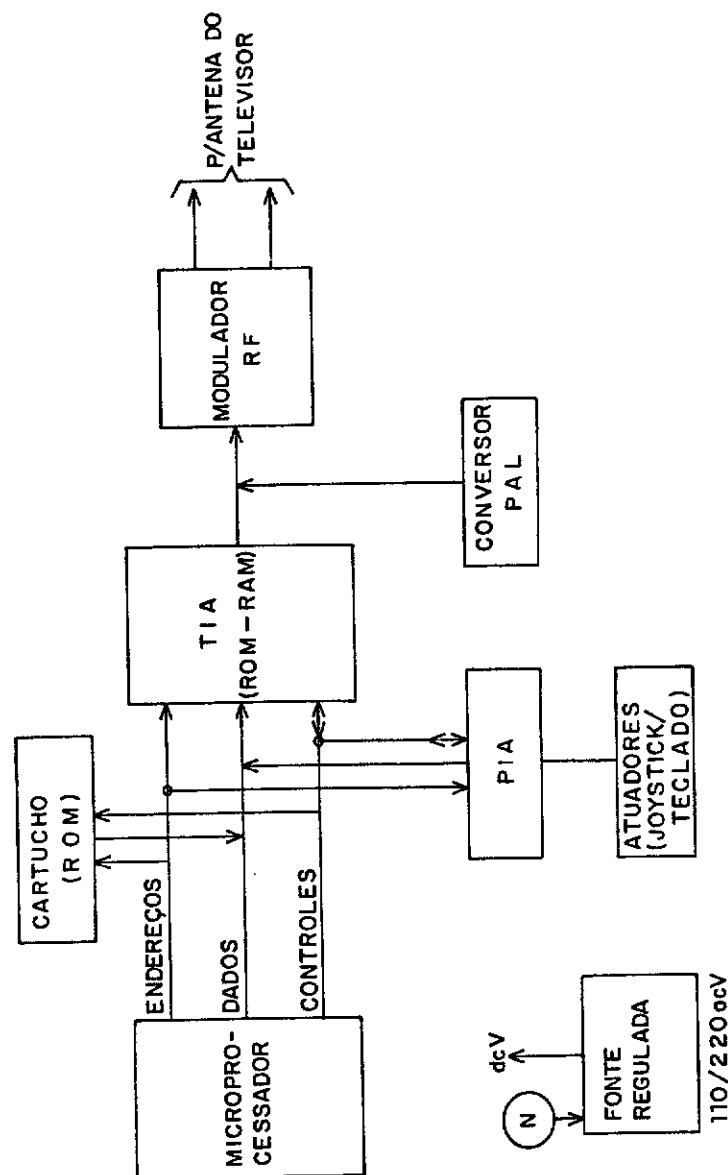


figura 2-1 - Arquitetura de vídeo-game.

buscados e processados pelo microprocessador em espaços de tempos determinados pelo programa do cartucho. Um caso típico seria o programa instruir o microprocessador para dar entrada nos dados dos atuadores a cada 10 milissegundos (10 segundos divididos por 1000) e conforme o resultado obtido alterar a imagem e o som sendo apresentados na TV. Para que os atuadores atuem no programa no momento desejado e com níveis de voltagem e corrente elétrica adequados coloca-se entre eles e o microprocessador a PIA ("peripheral interface adapter", interface adaptadora de periféricos).

Os dados recebidos do cartucho e dos atuadores são processados pelo microprocessador, que a partir deles deve produzir imagens e sons no receptor de TV. Sinais de vídeo, entretanto, são bastante complexos, sujeitos a uma série de padronizações e sinais de sincronização. Usa-se então a TIA ("television interface adapter", interface adaptadora de televisão) que, recebendo os dados do microprocessador transforma-os em sinal de vídeo com todos os requisitos e sincronizações.

O vídeo-game é conectado à antena do receptor de televisão. Portanto, ele deve funcionar exatamente como se fosse uma estação emissora. O sinal saindo da TIA é então modulado pelo MODULADOR DE RF, transformando-se de fato numa transmissão de televisão comum (canal 2, 3 ou 4).

Os vídeo-games não foram projetados no Brasil, tendo sido meramente copiados de modelos norte-americanos. Em especial as TIAs são circuitos integrados também projetados lá. Como o sistema de televisão dos Estados Unidos (NTSC) é diferente do nosso, é preciso usar um CONVERSOR que adapte o sinal da TIA para o nosso sistema PAL-M.

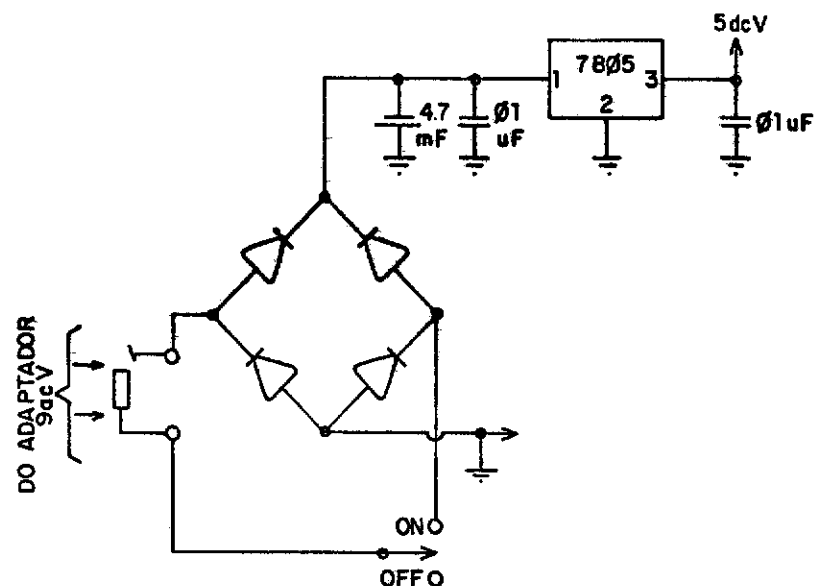


figura 2-2 - Fonte de alimentação

A fonte de alimentação elétrica de vídeo-games é externa (tomada de parede). Sendo esta de 110 V/220 V alternada, e como os CIs funcionam dentro de especificações bastante rígidas de tensão e corrente torna-se necessário retificação e regulação.

A prática usual (seguida pelo ATARI e pelo ODISSEY) é fornecer um ADAPTADOR externo (retificador 110V/220V alternados para 9 Volts dc). Internamente ao console do vídeo-game os 9 Volts são regulados para 5 Vdc através do REGULADOR DE TENSÃO comum do tipo 7805, e esta é a tensão que alimentará todo o sistema. Um esquema geral da fonte é apresentado na figura 2-2-

CAPÍTULO 3 – MICROPROCESSADORES

Os microprocessadores foram inventados em 1979 e desde então sofreram enorme evolução, caracterizada por "gerações". Por força de sua eficiência, aplicabilidade e baixo custo transformaram-se em núcleos de Eletrônica. Microprocessadores são "cérebros" não só de vídeo-games mas ainda de computadores, equipamentos industriais e brevemente também de televisores e automóveis.

Microprocessadores podem ser adquiridos em qualquer loja de componentes eletrônicos e os preços variam de 10% a 25% do salário mínimo para os mais populares, tal como dos vídeo-games. Algumas marcas especiais podem custar até 5 salários mínimos.

1 - EVOLUÇÃO DOS MICROPROCESSADORES

A primeira geração de microprocessadores começou com o 4004 da INTEL, sendo logo em seguida substituído pelo 4040. Esses dois modelos eram caracterizados como "microprocessadores de 4 bits", ou seja, tinham 4 pinos de dados.

A mesma fábrica lançou o 8080, um aperfeiçoamento dos anteriores dispondo de 8 pinos de dados - "microprocessador de 8 bits". A ZILOG projetou o Z-80, o mais popular microprocessador de 8 bits (usado no vídeo-game COLECOVISION), totalmen-

te compatível com o 8080 e com maior lista de instruções. Ainda nesta geração encontramos o 6502 da MOSTEK, usado nos famosos microcomputadores APPLE, do qual o microprocessador 6507 (fabricado pela ROCKWELL) do ATARI é um modelo simplificado possuindo a mesma lista de instruções.

Os microprocessadores citados até aqui necessitam de dispositivos externos onde esteja gravada a sequência de instruções que irão executar (ROM, PROM, EPROM) e onde possam armazenar dados temporariamente (RAM). A INTEL lançou o microprocessador 8048 que tem internamente sua própria ROM e RAM, ou seja, é um único CI com quase todos os estágios de um computador inteiro. Esse microprocessador é o pai da geração denominada "one single chip" e foi adotado pelo ODISSEY.

Da mais recente evolução na área surgiram os "microprocessadores de 16 bits", com 16 pinos de dados. O vídeo-game INTELLIVISION emprega um microprocessador desta geração, fabricado pela GENERAL INSTRUMENTS.

2 - ARQUITETURA DO MICROPROCESSADOR

Na sua arquitetura (figura 3-1), os microprocessadores dispõem de registradores (cadeias de flip-flops onde é possível armazenar bits) de uso geral (A-acumulador) de uso específico (IR - "instruction register" - registrador de instruções); PC - "program counter" - contador de programa); no nosso exemplo limitamo-nos aos registradores mais indispensáveis, mas geralmente aparecem em número bem maior.

No PC fica registrado (em binário) o endereço da instrução gravada na memória a ser executada.

tada. Após a execução de cada uma o conteúdo do PC é incrementado + 1, o que leva a uma execução sequencial do programa gravado. Instruções especiais de "jump" (salto) carregam qualquer endereço desejado no PC, permitindo escapar da sequência.

O conteúdo do PC é colocado na barra de endereços para a memória externa. Esta coloca na barra de dados o conteúdo do endereço solicitado pelo PC, que é armazenado no IR ("instruction register" - registrador de instruções). O DECODIFICADOR DE INSTRUÇÕES e o controle de TEMPORIZAÇÃO decodificam a instrução do IR, para que ela possa ser executada.

O processo acima é uma LEITURA de memória. Na ESCRITA o PC é desativado pelo MULTIPLEXER que apresenta o endereço onde se deseja "escrever" (gravar bits, numa RAM por exemplo).

A ALU ("arithmetic-logic unit", unida de lógica-aritmética) realiza todas as operações lógicas e aritméticas requeridas pelo sistema. Observe que um dos membros da operação é tomado da linha de dados e o outro do acumulador; o resultado obtido pela ALU é posto na linha de dados e geralmente escrito no acumulador. Isto explica duas características de programas para microprocessadores:

- 1 - carregar o acumulador com os dados que se quer operar.
- 2 - transferir para outro registrador o conteúdo do acumulador, deixando-o livre para receber outros resultados.

3 - TEMPORIZAÇÃO DO MICROPROCESSADOR

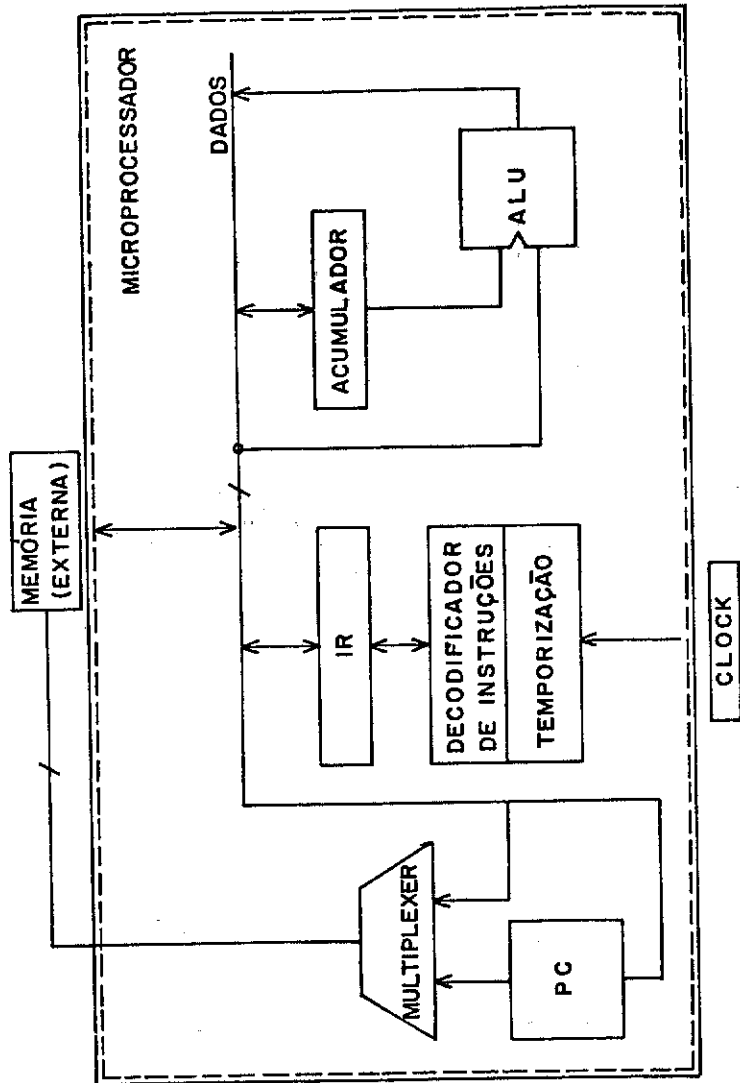


figura 3-1 - Arquitetura de microprocessador.

As operações do microprocessador são realizadas em sincronia com um sinal externo de CLOCK (relógio). A cada sinal do clock ocorre mais um passo na execução das instruções, classificando-se os tempos do microprocessador em três categorias (figura 3-2):

- 1 - o ciclo do clock propriamente dito (T1, T2, T3 , etc.
- 2 - o ciclo de máquina (M1, M2, M3), composto por 3 a 5 ciclos de clock, que indica em qual passo da execução da instrução o microprocessador se encontra. M1 sempre representa o início de busca da instrução gravada na memória para ser executada e os microprocessadores costumam emitir um sinal indicando que estão neste estado (por exemplo, sinal SYNC - PINO 7 no 6507 do ATARI).
- 3 - ciclo de instrução, compreendendo 3 a 5 estados de máquina, que englobam desde a busca da instrução (M1), a sua decodificação e execução (M2 , M3).

O sinal de clock varia de 1 MHZ a 10 MHZ (1 MHZ= 1 MEGA HERTZ = 1.000.000 de ciclos por segundo), dependendo da marca e do modelo do microprocessador. Além do mais muitos (como o Z-80 e o 6507) tem um limite máximo para o sinal do clock, mas operam normalmente com qualquer sinal abaixo desse limite. Adotemos uma média de 2 MHZ = 2.000.000 ciclos para a execução de uma instrução (4 ciclos de máquina x 5 ciclos de clock cada). Disso resulta que o microprocessador é capaz de executar 100.000 instruções por segundo.

Em microcomputadores e outros sistemas microprocessados o clock pode ser gerado por um

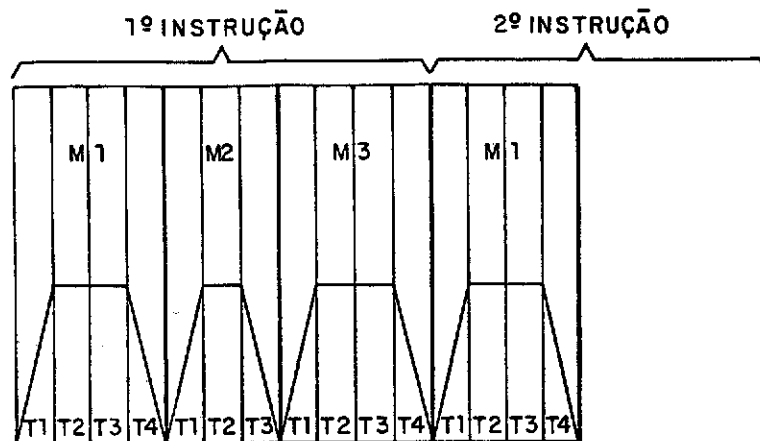


figura 3-2 - Temporização de microprocessador.

circuito multivibrador ou, em casos de maior precisão, por circuito oscilador contendo um cristal. Para vídeo-games, entretanto, a sincronização perfeita entre microprocessador e receptor de televisão é necessidade primordial. Isto porque, como vimos no esboço da arquitetura de vídeo-games, estes devem funcionar exatamente como se fosse estações transmissoras de TV (canal 2, 3 ou 4), emitindo não só sinais de vídeo e áudio, como também sinais de sincronização. Além do mais os programas (jogos) exigem coordenação bastante precisa entre os atuadores (joystick, teclado) e o vídeo, ambos sob comando sincronizado do microprocessador. Como é preciso um cristal na TIA para geração de cor, a prática é usá-lo também para geração do clock do microprocessador. Veremos que no ODISSEY um cristal de 7,151222MHZ fornece o sinal de clock para o pino do display gráfico (cor-

respondendo à TIA) e ao mesmo tempo para o pino ("XTAL 1", cristal 1) do microprocessador 8048.

4 - BARRAS DE ENDEREÇOS, DADOS E CONTROLES

Vimos na figura 2-1 que o microprocessador se comunica com todos os outros estágios através de trilhas de circuitos impresso, por onde a corrente elétrica circula indicando os níveis lógicos. Elas são divididas em 3 grupos, denominados barras.

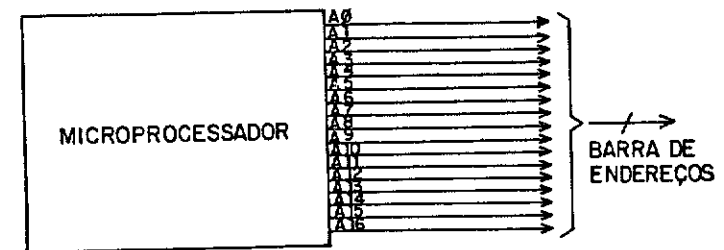


figura 3-3 - Barra de endereços.

A BARRA DE ENDEREÇOS (figura 3-3) transporta sinais no sentido do microprocessador para o resto do sistema, na forma de bits compondo um número binário, indicando com qual endereço ele quer se comunicar. Na maioria dos microprocessadores há 16 pinos de endereços, resultando uma barra de endereços com 16 trilhas de circuito impresso, o que possibilita o endereçamento de 65.536 posições = 64K posições (o cálculo é feito baseado na fórmula: TOTAL DE POSIÇÕES = 2^N , onde N é o número de pinos de endereços; no exemplo acima $2^{16} = 65.536$). Vídeo-games não se encaixam nessa maioria, pois o 6507 do ATARI tem somente 12 pinos de endereços (permitindo o endereçamento de $2^{12} = 4.096$ posições) e o 8048

tem uma manipulação toda especial das barras, como veremos adiante.

Para diferenciar os vários componen - tes com os quais o microprocessador se comunica, permitindo que cada um tenha seu próprio endereço, todos eles são conectados aos bits menos significativos da barra de endereços e os bits mais significativos são distribuídos para diferenciação. A figura 3-4 mostra um exemplo típico. Os 9 bits menos significativos (de A0 até A8) são comuns aos três CIs. A9 é conectado ao pino CS ("chip select", seleção de chip) do C13, A10 ao C12 e A11 ao C11. Portanto, o endereço existente de A0 até A8 só pertencerá ao C13 quando A9 estiver em nível 1. Disso resulta que este CI conterá os endereços A9=1, A8=1/0, A7=1/0, A6=1/0, A5=1/0, A4=1/0, A3=1/0, A2=1/0, A1=1/0, A0=1/0, ou seja de 1 000 000 000 até 1 111 111 111 B (em hexadecimal 200H até 3FFH), que na nossa numeração decimal corresponde de 512 até 1.023. Para o C12 temos A10=1, A9=1/0, A8=1/0, A7=1/0, A6=1/0, A5=1/0, A4=1/0, A3=1/0, A2=1/0, A1=1/0, A0=1/0, ou seja, de 1000000000 B até 1111111111 B (em hexadecimal 400 H até 7FF H), correspondendo em decimal de 1.024 até 2.043. Na verdade apenas os endereços de 000000000 B = 000H = 0 até 111111111 B = 1 FFH = 511 são válidos e internos aos CIs, o MSB servindo apenas para selecionar um dos três. Por exemplo, se o microprocessador apresentar na barra de endereços A8=0, A7=1, A6=0, A5=0, A4=0, A3=0, A2=0, A1=0, A0=1 (01000001 B = 081 H = 129), as respectivas posições 129 dos tres CIs serão ativadas, mas só será efetivamente habilitada aquela pertencente ao CI que tem o MSB em nível 1; as outras posições não se comunicarão com o microprocessador.

A BARRA DE DADOS (figura 3-5) é composta de 8 trilhas de circuito impresso que levam e

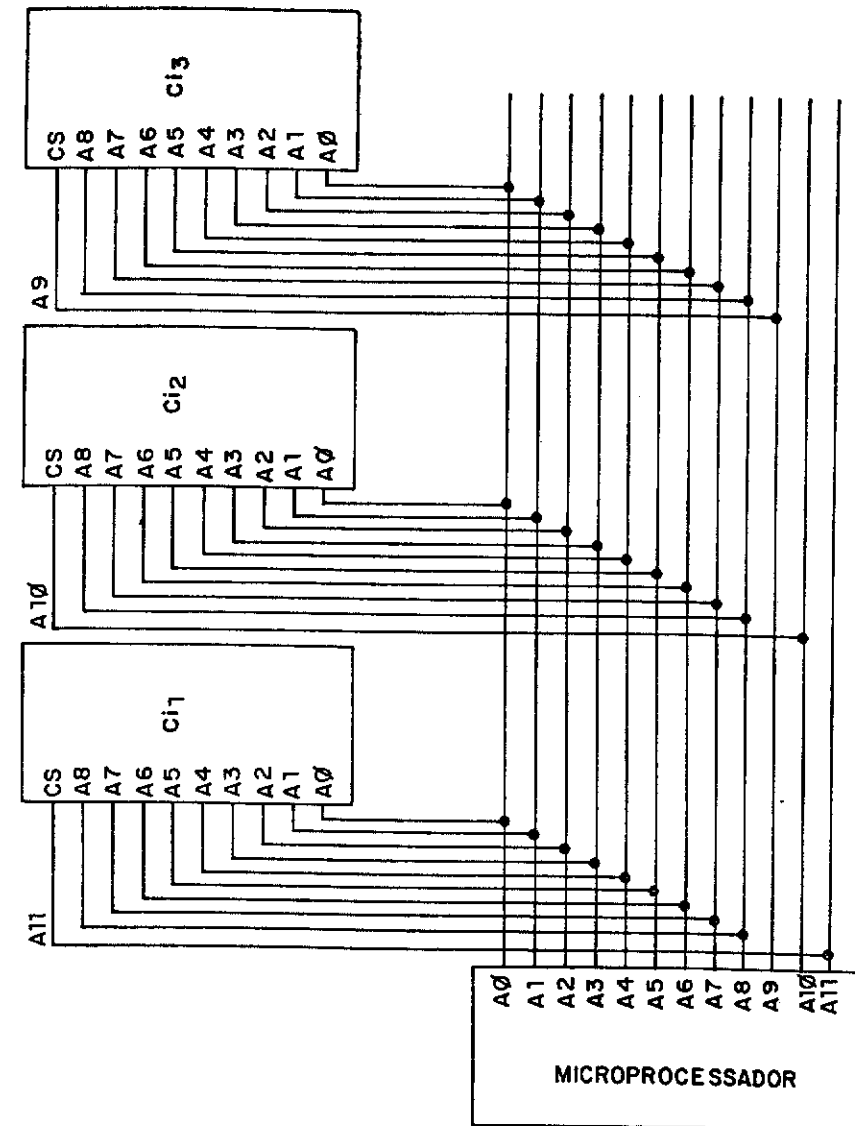


figura 3-4 - Diferenciação de endereços por distribuição de terminais de seleção CS.

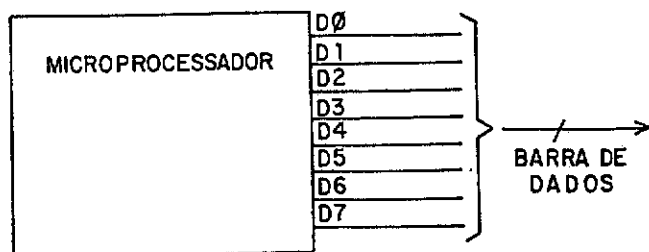


figura 3-5 - Barra de dados.

trazem a corrente elétrica contendo os bits de dados. Observe que a barra de endereços conduz sinais sempre na direção do microprocessador para o resto do sistema - dizemos que é unidirecional - enquanto na barra de dados o fluxo de sinais pode ser tanto saindo quanto chegando ao microprocessador - bidirecional. Na figura 2-1 vemos que os dados do cartucho vão dele para o microprocessador, já que ele é uma ROM onde apenas se lê e não se escreve; o mesmo sentido para o microprocessador é seguido pelos dados da PIA, pois também só ocorre leitura de dados dos atuadores; para a TIA, e se houver RAM independente (como no ODISSEY), será possível não só a leitura como a escrita, originando um fluxo bidirecional de dados.

A BARRA DE CONTROLES (figura 3-6) é usada para controlar os sinais das outras barras, indicando o sentido do fluxo, o tempo em que cada sinal deve ser considerado e outros comandos específicos de cada microprocessador. O número de pinos de controle, e correspondentemente de trilhas do circuito impresso, depende do tipo de microprocessador, estando na faixa de 3 a 8.

De importância fundamental são os controles READ (leitura) e WRITE (escrita). Quando

o microprocessador emite o sinal READ (figura 3-7) significa que ele está lendo os bits da barra de dados originados no endereço indicado pela barra de endereços (o fluxo é no sentido resto do sistema para o microprocessador), ou ordenando que a PIA receba dados dos periféricos. A emissão de WRITE pelo microprocessador (figura 3-8) sinaliza que ele está enviando dados para serem gravados na posição indicada pela barra de endereços (fluxos no sentido microprocessador para o resto do sistema), ou ordenando que a PIA envie dados para os periféricos.

Além de emitir sinais de controles o microprocessador também pode recebê-los. Um deles - READY (preparado, pronto; também pode aparecer como WAIT - espere) é de grande importância em trabalhos de consertos. Quando esse sinal chega ao microprocessador ele para de executar as instruções (permanece indefinidamente no estado M1), até que seja retirado.

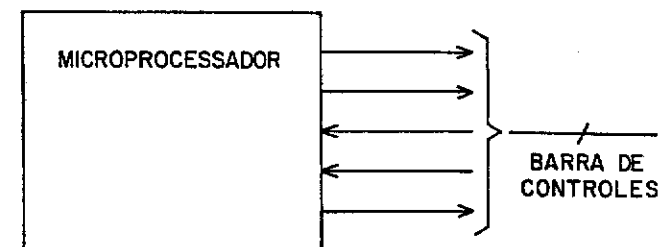
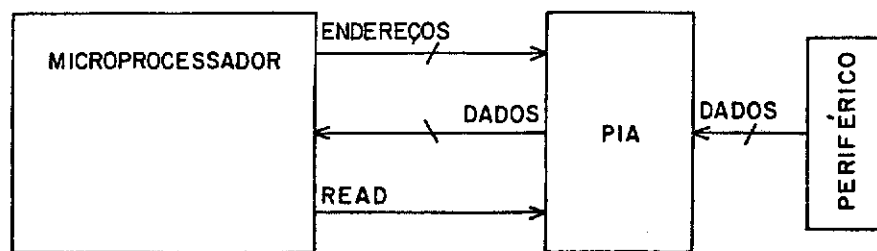
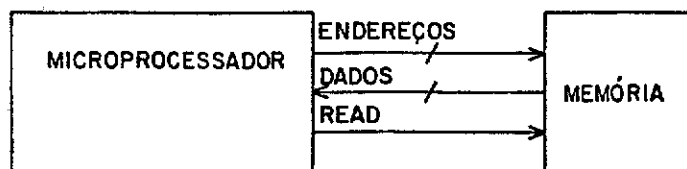


figura 3-6 - Barra de controles.

É necessário considerar com atenção o tipo de lógica-positiva ou negativa - empregada pelos controles do microprocessador. Assim, se ele tem o terminal $\overline{\text{READ}}$ (lógica negativa indicada pela barra) significa que ocorrerá uma leitura (ou input) quando o respectivo pino estiver em nível 0 e NÃO

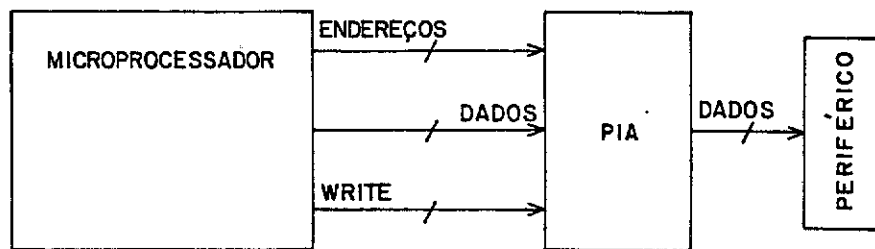


(B)

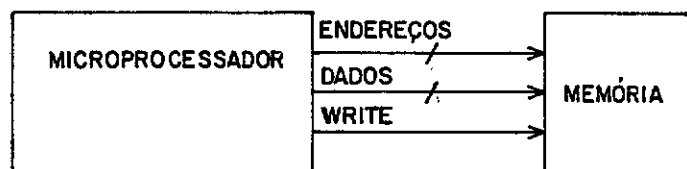


(A)

figura 3-7 - Controle READ (A) Leitura de memória
(B) Entrada de dados de periférico.



(B)



(A)

figura 3-8 - Controle WRITE (A) escrita na memória
(B) Saída de dados para periféricos

ESTARÁ ocorrendo uma leitura (ou input) quando o nível for 1. Um caso interessante é a existência do terminal $\text{READ}/\overline{\text{WRITE}}$ (usado no 6507 do ATARI, pino 26). O pino em nível 1 indica leitura e em nível 0 escrita.

CAPÍTULO 4 – CARTUCHO

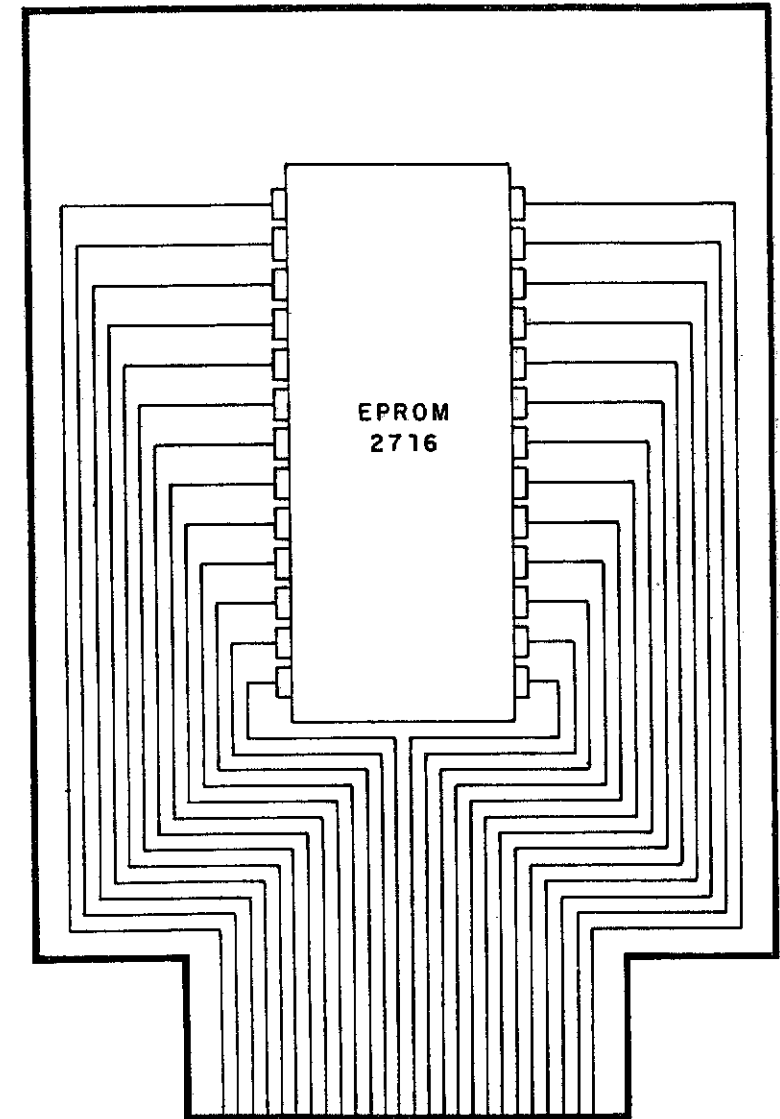
O cartucho do vídeo-game é uma ROM soldada numa placa de circuito impresso que encaixa nele através de uma fenda com conectores (figura 4-1). Quando encaixamos o cartucho no console estamos acrescentando a ROM ao circuito do vídeo-game.

Os dados são programados na ROM pela empresa fabricante de cartucho segundo um programa elaborado por especialistas. Cartuchos contendo os mais variados jogos são vendidos em lojas (cine-fotos, lojas de computadores, etc) e devem ser adquiridos para a marca do vídeo-game em que serão usados (cartuchos para o ATARI não são compatíveis com o ODISSEY, e vice-versa).

A elaboração do programa é bastante trabalhosa e exige profundos conhecimentos da lista de instruções do microprocessador e de todo o sistema do vídeo-game. Esta é a parte de SOFTWARE contraposta à parte eletrônica denominada HARDWARE. Os famosos "piratas" são pessoas que copiam em outra ROM a gravação dos bits de um cartucho, colocam a cópia no seu próprio e o comercializa.

1 - PROGRAMA

A gravação é feita em binários na sequência de instruções a serem executadas pelo microprocessador. Cada endereço na ROM terá gravado 1



4-1 - Cartucho.

byte (8 bits), correspondente a uma instrução. Algumas instruções usam 2 ou 3 bytes, ocupando portanto 2 ou 3 endereços na ROM. Com o programa em andamento o microprocessador lê, decodifica e executa a instrução conforme vimos no texto referente à figura 3-1.

Um programa para o microprocessador Z-80 ler o byte de dados do endereço 0000 0000 0100 0001B = 0041H = 65, complementá-lo (transformar zeros em 1s e vice-versa), e gravar o resultado no endereço RAM 0001 0000 0001 0000B = 1010H = 4128 seria gravado na ROM da seguinte maneira (supondo que esta parte do programa começa no endereço 0000 1000 0000 0000B = 0800H = 2.048 da PROM):

ENDEREÇO DA INSTRUÇÃO (PROM) - BINÁRIO	INSTRUÇÃO	(COMENTÁRIO)
0000 1000 0000 0000	0011 1010	Carrega acumulador
0000 1000 0000 0001	0100 0001	com conteúdo do endereço 0041 H.
0000 1000 0000 0010	0000 0000	Complementa byte
0000 1000 0000 0011	0010 1111	carregado no acumulador.
0000 1000 0000 0100	0011 0010	Cópia o byte carregado na RAM.
0000 1000 0000 0101	0001 0000	
0000 1000 0000 0110	0001 0000	

(os comentários não pertencem ao programa, sendo apenas para orientação do leitor).

Os 7 bytes da coluna INSTRUÇÃO são gravados cada um num endereço da ROM (indicado à esquerda). O microprocessador busca uma a uma, sempre

na sequência, executando-as. O byte correspondente à instrução é tirado da lista de instruções do microprocessador, fornecida pelo fabricante.

2 - NÍVEIS DE LINGUAGEM

Trabalhar com o programa em números binários (ou hexadecimais) é cansativo e pouco produtivo, pois não há uma indicação imediata do significado das instruções. Usa-se então uma "tradução" das instruções em binário para outra forma mais compreensível e fácil.

O programa em binários é denominado CÓDIGO DE MÁQUINA, sendo este o mais baixo nível de programação e o único que é efetivamente entendido pelo microprocessador. O exemplo anterior (só com a coluna INSTRUÇÃO) é uma amostra típica (na verdade um programa completo teria cerca de 1 Kbyte de instruções).

Numa "tradução" para um nível intermediário-denominado LINGUAGEM ASSEMBLER - cada instrução é representada por um grupo de letras (MNEMÔNICO) abreviando o significado. A abreviatura, entre tanto, deriva da palavra escrita em inglês. Por exemplo, a instrução (Z-80) "1100 0011B = C3H" é escrita em assembler "JP", abreviatura de "jump" (pulo, salto, em inglês), associando-se ao seu significado que é obrigar o microprocessador a dar um pulo para outra parte do programa, saltando diversas instruções. Cada instrução opera sobre um OPERANDO - conteúdo de um registrador, byte na memória ou dado imediato acrescentado por intermédio de periférico. No exemplo acima teríamos de indicar o endereço do jump, que seria o operando. Para evitar registrar endereços como posição absoluta em binário nós os escrevemos como LABEL (rótulo, título, em inglês), um nome qualquer que passa a representar aquele endereço. Só é

colocado LABEL para as instruções que precisarem usá-lo. COMENTÁRIOS, sempre antecidos de ponto-e-vírgula, não fazem parte do programa propriamente dito, servindo de orientação para o leitor, mas desconhecido pelo microprocessador. Abaixo temos um programa assembler para Z-80 (semelhante ao programa anterior em código de máquina) que carrega o acumulador com conteúdo de memória, complementa-o e copia-o 5 vezes na posição 1010H.

LABEL	MNEMÔNICO	OPERANDO	COMENTÁRIO
	LD	B,5H	;carrega 5H no registrador B
CARGA	LD	A,0041H	;carrega acumulador com conteúdo do endereço 0041H
	CPL		;complementa acumulador
	LD	(1010H),A	;copia conteúdo do acumulador na posição 1010H
	DEC	B	;decrementa B de -1
	JPNZ	Carga	;salta para instrução CARGA se B não for zero
	END		;termina programa se B é zero.

LD (de "LOAD", carga) no início do

programa é a instrução que carrega o número 5 no registrador B (interno ao microprocessador, semelhante ao acumulador); obviamente 5 é copiado como binário 0000 0101B. Este registrador será usado como CONTADOR, indicando quantas vezes a operação foi realizada; quando marcar 0 será a indicação que já houve tantas repetições quantas queríamos. CPL (de "complement", complemento) representa a instrução de transformar zeros em 1s e vice-versa. Observe que no caso não é usado operando, pois já se subentende que seja o acumulador. DEC ("decrement", decremento) diminui -1 o registrador B a cada vez que a instrução é executada. JPNZ (de "jump non zero", salte não zero) carrega o PC (contador de programa) com o endereço referente a CARGA, fazendo o programa voltar à 2ª instrução, se o registrador B é diferente de zero; caso contrário (B=zero) encerra-se o programa. Assim, com B=5, o acumulador é carregado, complementando-o, copiado na memória e B é decrementado para 4, ocorrendo então uma nova carga, complementação, cópia e decretação de B para 3, e sucessivamente até B atingir zero. Um esquema de repetição de uma sequência de instruções como este é chamado LOOPING.

Tanto o código de máquina quanto a linguagem assembler são complexos demais para leigos em Eletrônica Digital. Para eles usa-se a LINGUAGEM DE ALTO NÍVEL, que pode ser BASIC, PASCAL, ALGOL, COBOL, FORTRAN, etc. Entretanto, quanto maior o nível da linguagem menor é a eficiência da programação, demandando maior tempo de execução e mais memória.

3 - PROGRAMAÇÃO PARA VÍDEO-GAMES

O programa-jogo do vídeo-game é gravado em binário na ROM. O tipo de ROM mais popular, inclusive usada no cartucho do ATARI, é a família 27xx

da INTEL. A marcação xx seguindo 27 indica a capacidade de dados armazenados, em Kbits. Assim, temos 2708 (8 Kbits= 1 Kbyte), 2716 (16 Kbits= 2 Kbytes) e 2732 (32 Kbits= 4 Kbytes).

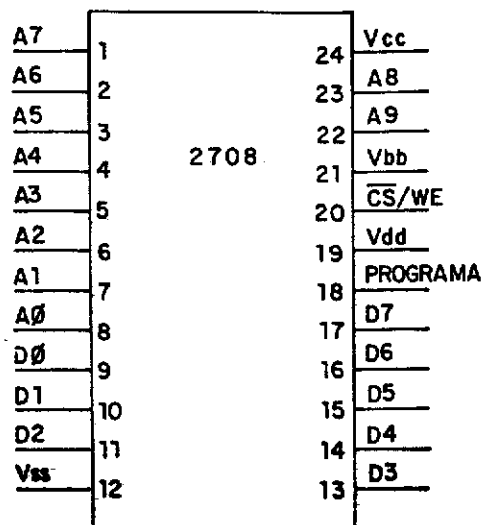


figura 4-2 - ROM 2708

A ROM 2708 de 1 Kbyte (figura 4-2) tem 24 pinos, sendo 10 deles para endereçamento (A0-A9) e 8 para dados (D0-D7). Em operação ela precisa de + 5 Volts em Vcc (pino 24), 0 Volts em Vss (pino 24), 0 Volts em Vss (pino 12, aterrado) e - 5 Volts em VBB (pino 21). Quando ocorre um sinal nível baixo (0) em $\overline{CS}/WE$ ("chip select/write enable", seleção de chip/habilitação de escrita) aparecem nos pinos de dados os níveis lógicos correspondentes ao endereço que está sendo aplicado. Observe que temos lógica negativa para $\overline{CS}$.

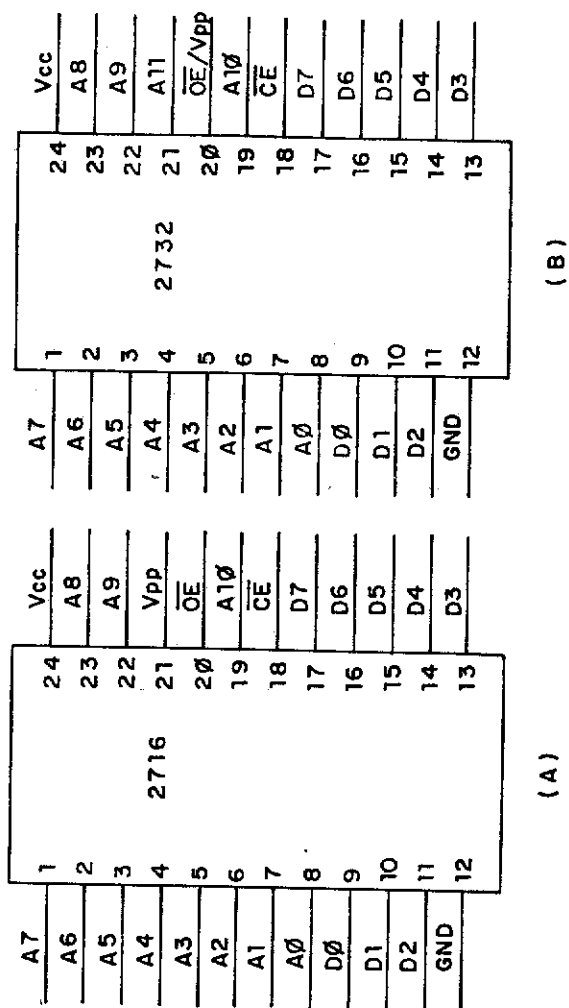
A ROM 2716 de 2 Kbytes (figura 4-3A)

também tem 24 pinos (11 de endereços, A0-A10, e 8 de dados, D0-D7), mas necessita apenas de alimentação de + 5 Volts. Para que haja leitura de dados deve o correr pulso nível baixo (0) tanto em $\overline{OE}$ ("output enable", habilita saída, pino 20) quando $\overline{CE}$ ("chip enable", habilita CI, pino 18).

A ROM 2732 de 4 Kbytes (figura 4-3B) tem pinagem semelhante à 2716, exceto o pino 21 que passa a ser A11, já que ela deve ter mais 1 terminal de endereço (completando 12, A0-A11).

A gravação de ROM 2716/2732 é extremamente simples: para a 2716 o pino 21 é alimentado com 25 Volts dc, o endereço e os dados a serem gravados são mantidos nos níveis lógicos desejados, $\overline{CE}$ é levado de 1 para 0 e depois retornado a 1; para a 2732 segue-se o mesmo processo, porém colocando a alimentação de 25 Volts no pino 20.

Na verdade uma gravação manual seria bastante trabalhosa (2 mil posições, cada uma com 8 bits de dados), e a prática corrente é usar uma GRAVADORA DE EPROM, que automaticamente emite os sinais de endereços e controles, bastando ao usuário acrescentar o byte de dados para as respectivas posições. Hobbistas e interessados em trabalhos não automatizados devem tomar o máximo cuidado na gravação de cada posição, pois não há correção para a PROM e um único erro pode desvirtuar completamente o programa. Uma solução é utilizar EPROM, que permite apagamento (sendo, entretanto, necessário adquirir uma APAGADORA EPROM). Para desenvolvimento de programas-jogos em fase experimental (debugging de software) é possível substituir a PROM 2716 pela RAM 6116, que tem a mesma pinagem; os dados são escritos e alterados à vontade, ao contrário da gravação definitiva. "Piratas" geralmente empregam uma COPIADORA DE EPROM



que copia automaticamente em uma (ou várias) PROMs virgens o programa gravado em outra (eles podem também "ler" o programa da ROM, fazendo uma lista de bytes correspondentes a cada endereço e depois gravar normalmente).

4 - RETIRADA DA ROM DO CARTUCHO

Para retirar do cartucho a placa de circuito impresso com a ROM descole o papel propaganda do jogo, na frente da caixinha, e solte o parafuso que está debaixo dele. A caixinha se abrirá em duas metades, com a placa encaixada numa delas.

figura 4-3 - (A) ROM 2716 (B) ROM 2732.

CAPÍTULO 5 – ATUADORES; JOYSTICK

O programa gravado no cartucho controla a sequência de operações realizadas pelo microprocessador. Uma parte das instruções orienta-o a buscar dados no "exterior" do sistema - ATUADORES comandados pelos jogadores - e conforme eles alteram as condições de jogo (posição de objetos na tela, sons, resultados, etc.). JOYSTICK e KEYBOARD ("teclado") são os atuadores usuais em vídeo-games.

1 - INTERFACEAMENTO DE ATUADORES

Embora os jogadores estejam continuamente mexendo no joystick ou no teclado o microprocessador só lhes dá atenção a cada 20 milissegundos, aproximadamente, analisando os dados dos atuadores, e introduzindo-os no programa. Assim, a participação dos jogadores se reduz a uma intervenção a cada 20 milissegundos, e neste ínterim o microprocessador executa cerca de 1.000 instruções. A figura 5-1 ilustra esse processo.

Para controlar a participação dos atuadores, mantendo-os desconectados eletricamente das barras (de endereços, dados e controles) e só ativando-os no tempo exato indicado pelo microprocessador é preciso usar uma INTERFACE. Esta pode ser um simples buffer tristate, como no ODISSEY, ou um sofisticado CI chamado PIA ("peripheral interface a

dapter", interface adaptador de periférico), como no ATARI.

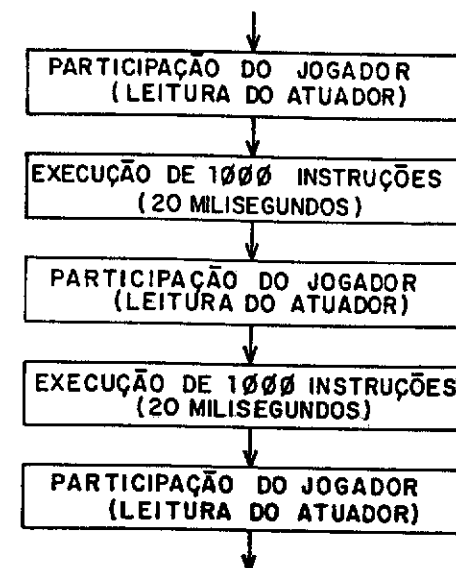


figura 5-1 - Participação dos jogadores no programa.

Em vista da importância do interfaceamento do sistema com o exterior, como no caso dos atuadores, cada fabricante desenvolveu uma PIA para seu próprio microprocessador (para alguns ela é denominada PIO - "peripheral input-output", entrada-saída de periférico). Além das funções de controle de tempo dos periféricos a PIA pode selecionar qual dos periféricos deve participar, servir de buffer, etc. Em microcomputadores é comum o uso de PIA que tanto pode dar entrada de dados quanto saída. Nos vídeo-games só interessa a entrada de dados.

Os dados entrando no sistema pela PIA ou pelo buffer são agrupados em PORTAS. Cada atuador é considerado uma porta e compete ao microprocessa -

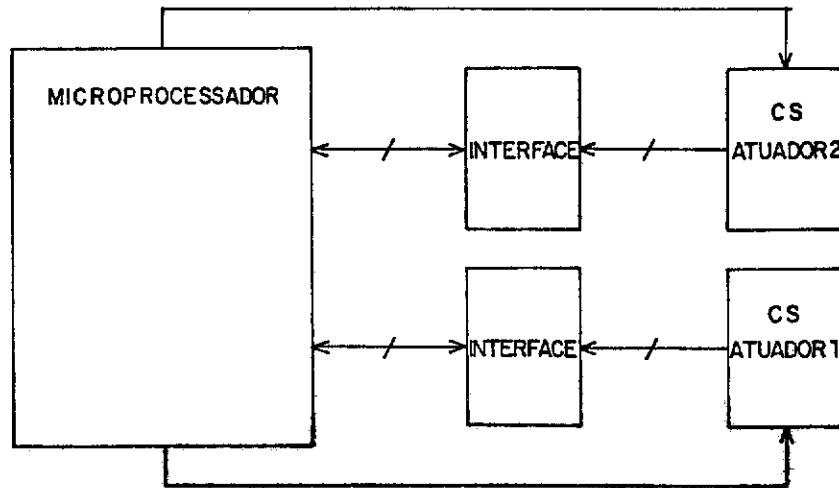


figura 5-2 - Seleção de atuador pelo microprocessador.

dor indicar qual deles participará em determinado instante do jogo (figura 5-2, como no ODISSEY), a través de sinal de controle tipo CS. Outra solução (figura 5-3, como no ATARI) é dividir a porta de entrada da interface para os dois atuadores, deixando os 4 bits menos significativos para um e os 4 mais significativos para o outro; o programa do cartucho deve providenciar instruções que separem os dois grupos e os processem separadamente.

2 - JOYSTICK

Os vídeo-games consideram 4 movimentos de um objeto na tela (figura 5-4): U - para cima (up), D - para baixo (down), R - para direita (right) e L - para esquerda (left). Movimentos oblí

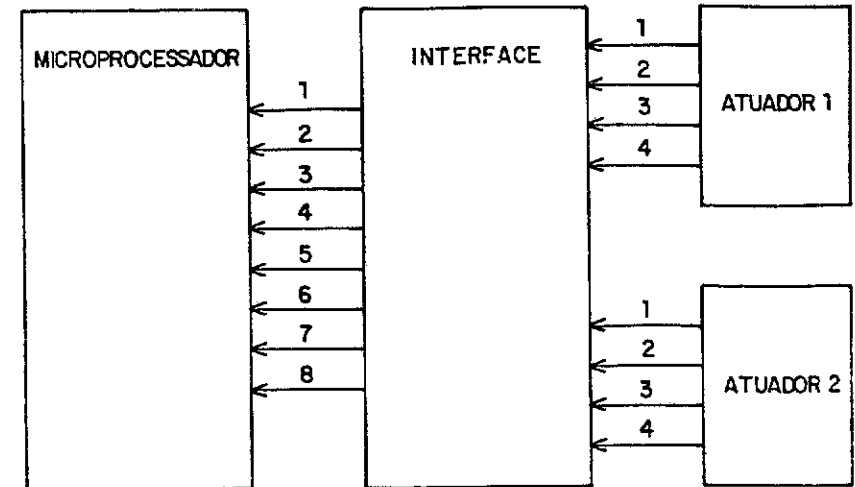


figura 5-3 - Divisão da porta em dois grupos de bits correspondendo aos dois atuadores.

quos são obtidos pela combinação de duas direções; a figura 5-5 mostra um movimento oblíquo resultante do deslocamento para a direita (R) e para cima (U).

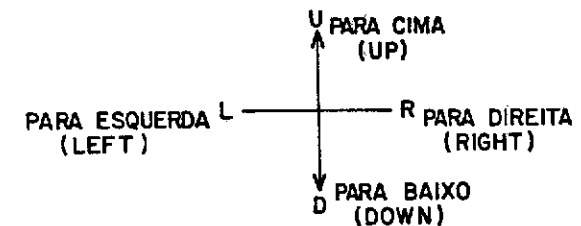


figura 5-4 - quatro direções de movimento

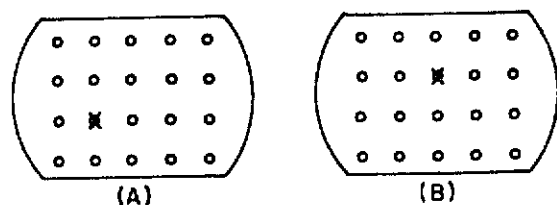


figura 5-5 - movimento R-U na tela (A) antes (B) depois.

JOYSTICK ("joy" - brincadeira, "stick" - vara) é um atuador manejado pelo jogador que controla os movimentos de objetos na tela. Pode ter várias formas, mas o mais comum é visto na figura 5-6.

No interior de sua caixa o joystick tem um circuito com 4 pontos normalmente em voltagem

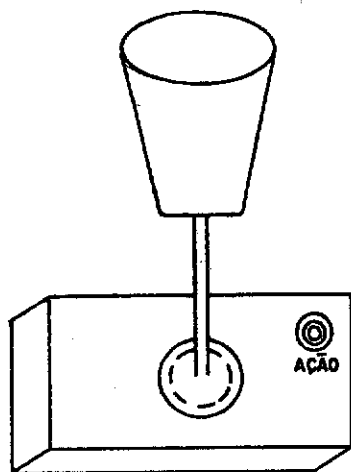


figura 5-6 - joystick

+ 5 Volts, nível 1 - setados (figura 5-7). Ao deslocá-lo o jogador provoca contacto entre um dos pontos e o terra, caracterizando ali nível 0 enquanto os outros permanecem em 1. Os pontos estão dispostos de tal modo que o contacto corresponde à direção de deslocamento do joystick.

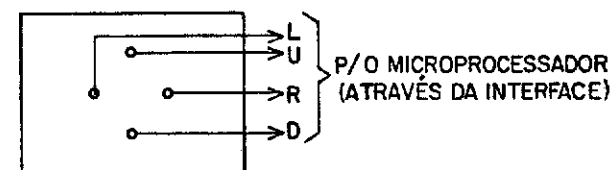


figura 5-7 - contactos internos do joystick.

Na figura 5-8 vemos como seriam os níveis dos pontos se o joystick fosse deslocado para a direita (R).

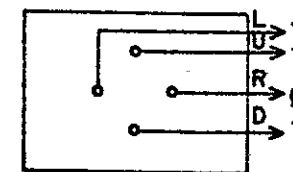


figura 5-8 - movimento para a direita.

A cada 20 milissegundos, aproximadamente, o microprocessador lê esses quatro bits e atualiza a posição dos objetos na tela conforme o resultado da leitura. O programa gravado no cartucho fornece ao microprocessador a sequência de instruções necessárias para a operação. Na figura 5-9 apresentamos o

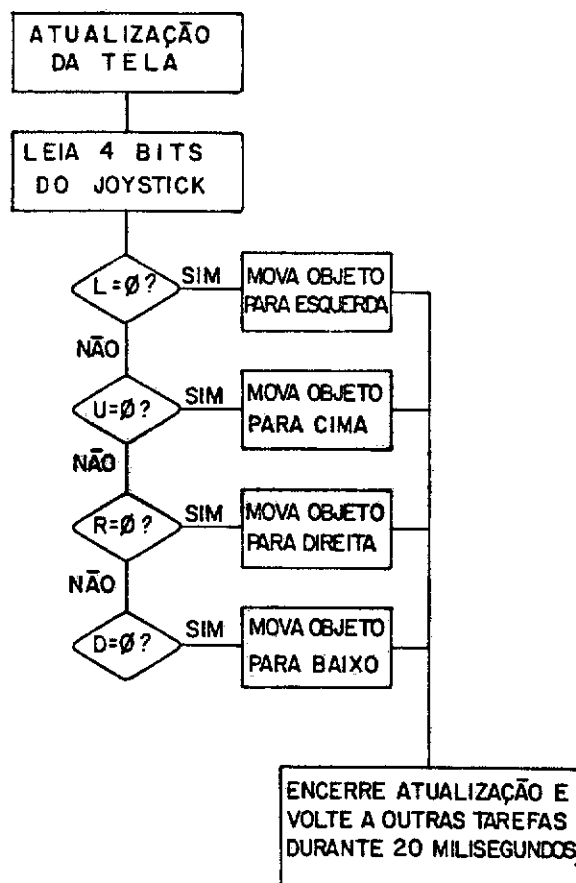


figura 5-9 - fluxograma de atualização da tela.

fluxograma simplificado do movimento do objeto na tela segundo o deslocamento do joystick. Note que cada passo do fluxograma corresponde a uma série de instruções em código de máquina do microprocessador.

A entrada dos 4 bits é feita através de uma interface ou PIA, como na figura 5-2. O intervalo de tempo para participação é obtido emitindo o sinal CS (ainda na figura 5-2) a cada 20 milissegundos.

Observe na figura 5-6 que há um "botão de atuação" ("action butom"). Quando pressionado ele fecha um 5º ponto de contacto no circuito do joystick, constituindo-se em mais um bit de entrada para ser processado. O programa no cartucho determina seu significado e quais providências o microprocessador deve tomar ao percebê-lo.

3 - TECLADO (KEYBOARD)

Outro atuador de vídeo-games é o TECLADO (ou KEYBOARD, pronuncia-se "kibôrd"), usado no ODISSEY mas não no ATARI. Tal como o joystick serve para entrada de dados, servindo ainda para entrada de comandos e funções.

O teclado é uma matriz onde, ao se apertar uma tecla é fechado contacto entre determinada coluna e fileira, provocando um sinal elétrico na quele cruzamento (figura 5-10). VARREDURA e INTERRUPTÃO são os 2 processos usados pelo microprocessador para identificar qual a tecla apertada.

Na varredura (figura 5-11) o microprocessador leva a nível lógico 0 a primeira coluna e testa uma a uma todas as linhas. Se alguma delas apresentar nível 0 é indicação que tem tecla apertada no cruzamento com aquela coluna (a parte inferior do

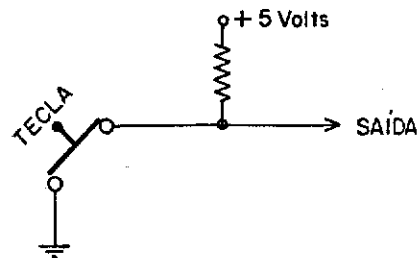


figura 5-10 - circuito de tecla.

resistor conectado à tensão 5 Volts estaria aterrado). A operação é repetida para as demais colunas a té varrer totalmente a matriz. Para identificar qual é o carácter referente à tecla apertada pode ser usado um CI decodificador tipo ROM, no próprio teclado ou a decodificação se realizará pelo microprocessador, através de instruções no programa.

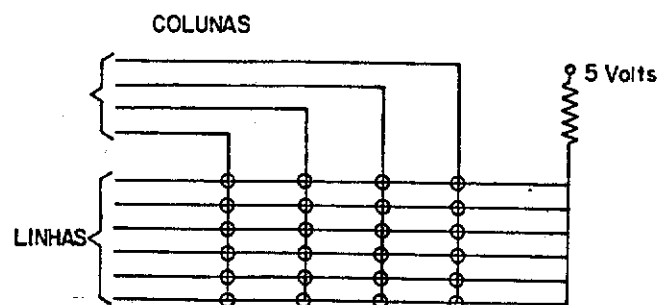


figura 5-11 - teclado por varredura.

A varredura é executada a cada 5 milisegundos, aproximadamente. Como é um espaço de tempo pequeno não há risco de se perder algum sinal de tecla apertada. Por outro lado, é um longo espaço de tempo para o microprocessador, não atrapalhando a execução de outras (ele pode executar cerca de 250 instruções no intervalo de 5 milisegundos entre cada varredura).

No modo de interrupção a tecla é codificada com um código (pode ser o ASCII) e guardada no decodificador de teclado (figura 5-12), ao mesmo tempo que é gerado um sinal de interrupção para o microprocessador.

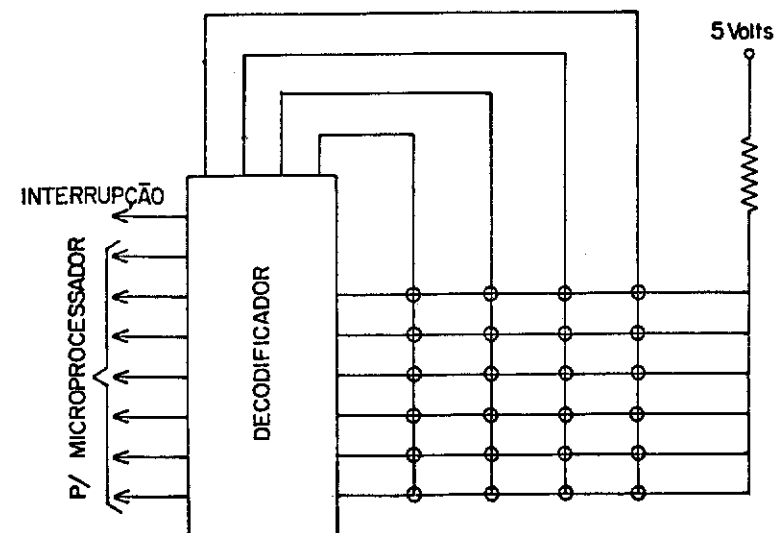


figura 5-12 - teclado por interrupção.

Este interrompe a execução das instruções e passa a interagir com o teclado, dando entrada ao código e armazenando-o na memória conforme programação.

CAPÍTULO 6 – TELEVISÃO PRETO E BRANCO

O receptor de televisão é utilizado pelo vídeo-game para apresentar imagens e sons gerados no jogo. Assim, é necessário o conhecimento teórico de transmissão televisada. Aos interessados em serviços práticos em vídeo-games cabe ressaltar que é indispensável conhecimento teórico; simples prática em manipulação do receptor de TV ajuda muito pouco.

Apresentaremos neste capítulo um rápido esboço de TV preto e branco (TPB), discutindo estágios e funções, e no próximo veremos TV a cores (TVC).

1 - A FORMAÇÃO DA IMAGEM NA TELA

O receptor de televisão é um CRT ("cathode ray tube", tubo de raios catódicos, figura 6-1), onde um feixe de elétrons é emitido no cátodo por efeito térmico (aquecimento), e atinge a tela fabricada de material fluorescente. Variações na tensão da grade de controle entre o cátodo e a tela permitem alterar a intensidade do feixe eletrônico, inclusive apagá-lo completamente. Na intensidade máxima uma grande quantidade de elétrons atingem a tela fluorescente, tornando-a bastante luminosa no ponto atingido, o que é percebido como "branco". No apagamento do feixe o ponto da tela não se ilumina, ficando na sua cor normal escura (como a TV desligada), ca

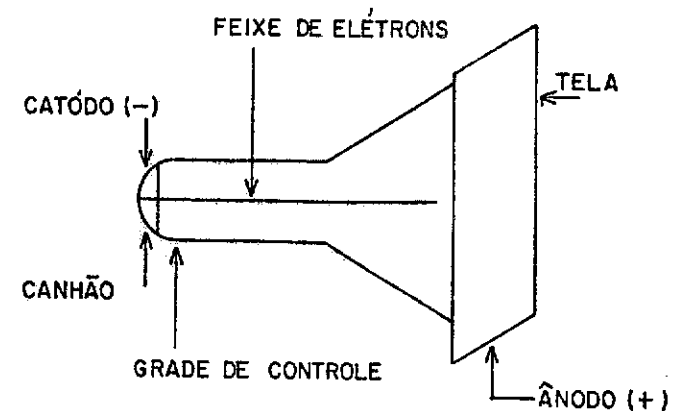


figura 6-1 - tubo de raios catódicos (CRT).

racterizando o "preto". Observe atentamente, que preto e branco não são cores, mas tonalidades, sendo essa distinção importante para a compreensão de TVC. Tonalidades intermediárias (cinza, por exemplo) são conseguidas com intensidades médias do feixe.

Circuitos de deflexão (usando princípios de deflexão de cargas elétricas por campos eletromagnéticos) fazem com que o feixe varra constantemente a tela da esquerda para a direita (traço), voltando em seguida à margem esquerda (retraço), e de cima para baixo, retornando ao canto superior. No retraço e no retorno para cima o feixe é apagado.

Esses movimentos são executados em tempos bastante precisos. Da esquerda para a direita gasta-se aproximadamente 64 microsegundos (1 microse

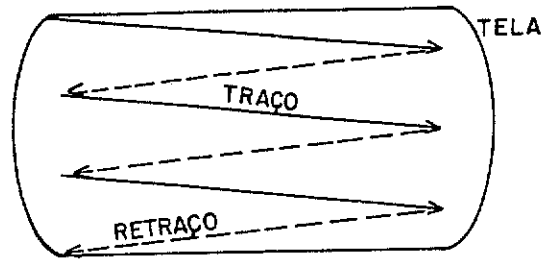


figura 6-2 - varredura da tela pelo feixe eletrônico.

gundo = 1 segundo dividido por 1 milhão), disso resultando que o feixe varre 15.750 linhas por segundo. Dizemos que a frequência horizontal é 15.750 HERZ. Quanto ao movimento vertical ele é realizado 60 vezes por segundo, ou seja, a frequência vertical é 60 HZ. Portanto, o que vemos na tela é o efeito de um feixe eletrônico excitando-a da esquerda para a direita e de cima para baixo, gastando 1/60 segundos para sair do canto superior e chegar ao canto inferior direito.

Enquanto o feixe faz seu trajeto a grade de controle vai intensificando-o ou apagando-o, criando pontos de luminosidade na tela. A vista humana retém durante algum tempo uma imagem já passada (persistência). Desta forma quando o feixe chega ao canto inferior direito nós ainda "estamos enxergando" (retendo na vista) todos os pontos de luminosidade que ele emitiu desde o início do trajeto no canto superior esquerdo. O conjunto de todos os pontos forma na nossa vista uma imagem total, que acreditamos estar vendo. Note que a televisão é uma ilusão óptica: a imagem não existe realmente na televisão, nossa vista é que a forma (é interessante constatar que os gatos, por não terem a mesma capacidade de retenção na vista como nós, são incapazes de ver uma imagem na tela).

A tela é composta por 525 linhas que deveriam ser percorridas 60 vezes por segundo pelo feixe. Mas esse esquema causa efeitos ópticos indesejáveis. Divide-se, então, a tela em dois "campos": o primeiro campo é constituído pelas linhas ímpares (1, 3, 5, ..., 525) e o segundo pelas linhas pares (2, 4, 6, ..., 524). Os dois campos juntos formam um "quadro". O feixe eletrônico passa a varrer o primeiro campo, depois o segundo; ou seja, percorre todas as linhas ímpares e em seguida as pares. Como são apresentados muito rapidamente nós os vemos como uma só imagem, integrando-os (portanto, percebemos o quadro e não 2 campos isolados). A frequência vertical continua a ser 60 HZ, mas como cada campo numa varredura é percorrido e na outra não, alternando, temos uma frequência de 30 HZ para os campos. O processo de alternância de campos formando um quadro é denominado "entrelaçamento".

2 - SINAL COMPOSTO DE VÍDEO

O sinal transmitido pela emissora deve informar ao aparelho receptor quanto é a intensidade de feixe eletrônico e à qual posição na tela refere-se a informação. A primeira parte corresponde ao SINAL DE VÍDEO - Y - e a segunda ao SINCRONISMO - SINC.

O sinal de vídeo é transmitido como nível de voltagem. A voltagem necessária ao apagamento do feixe no retraço é tomada como base e denominada "apagamento" ou "blanking", tendo 75% do total do sinal (figura 6-3). O preto (apagamento do feixe, deixando a tela na sua cor natural escurecida) é padronizado 7,5% abaixo do nível de blanking; o branco é tomado como 12,5% do blanking. Obviamente um tom acinzentado estará na casa de 50%. Observe que apesar do preto já ser suficiente para apagamento do feixe o blanking é posto um pouco acima para e

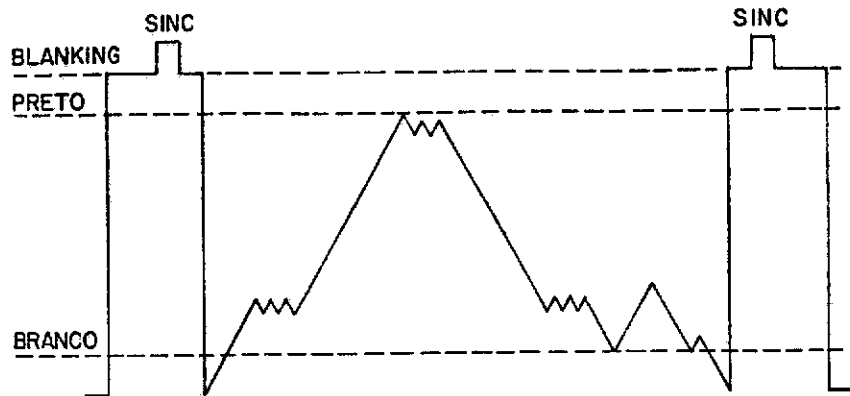


figura 6-3 - níveis do sinal de vídeo.

evitar interferência de níveis.

Durante o blanking (portanto com o feixe apagado) é emitido o sinal SINC de sincronismo que indica o momento exato da passagem de uma linha para outra. Como a mudança de linha é feita 15.750 vezes por segundo também esta deverá ser a frequência do SINC. Ou seja, a cada 64 microsegundos (frequência de 15.750 HZ) o sinal de vídeo (representado pelos picos dentados na figura 6-3) é interrompido, a voltagem do sinal é posta em nível de apagamento e um sinal de sincronismo enviado. Ao receber este SINC o receptor fica sabendo que este é o momento exato de mudar o feixe eletrônico de linha. As informações de vídeo entre dois SINC são então distribuídos nos seus pontos corretos, pois o feixe faz o trajeto de uma linha a velocidade uniforme.

A sincronização da deflexão vertical também segue o mesmo esquema anterior, mas para diferenciar seu sincronismo do horizontal usa-se um grupo de 6 pulsos cerrados (figura 6-4) e mais longos. Antes e após o grupo de sincronismo vertical aparecem pulsos de "equalização" para harmonizar campos pares e ímpares.

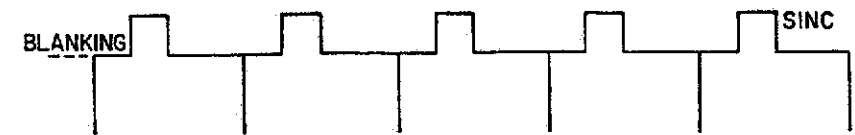


figura 6-4 - sincronismo vertical.

3 - MODULAÇÃO DO SINAL COMPOSTO DE VÍDEO

Além de sinais de vídeo e sincronismo é necessário transmitir informação de áudio (som). Usa-se sistema diferente de modulação para o vídeo (AM) e para o áudio (FM).

No AM ("amplitude modulation", modulação de amplitude) a frequência característica da emissora (cada uma tem a sua, outorgada pelo governo) - chamada PORTADORA (figura 6-5) - tem seus níveis de amplitude de voltagem alternados conforme o sinal modulador (figura 6-6), formando um "envelope" que a contém (figura 6-7). A amplitude do sinal modulador indica a intensidade do feixe eletrônico, que resultará em níveis de luminosidade na tela; sua frequência determinará os detalhes da informação (quantidade de vezes que a luminosidade da tela é alterada, resultando regiões claras e escuras que



figura 6-5 - frequência portadora



figura 6-6 - sinal modulador.

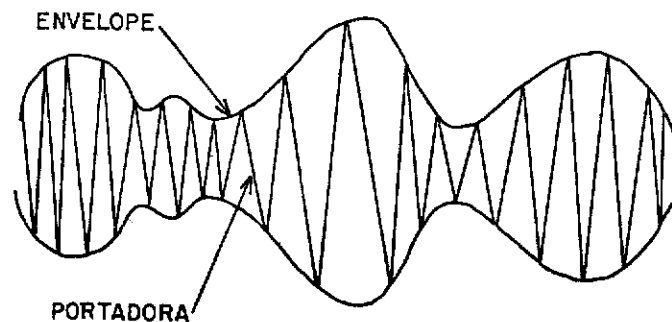


figura 6-7 - portadora modulada - AM.

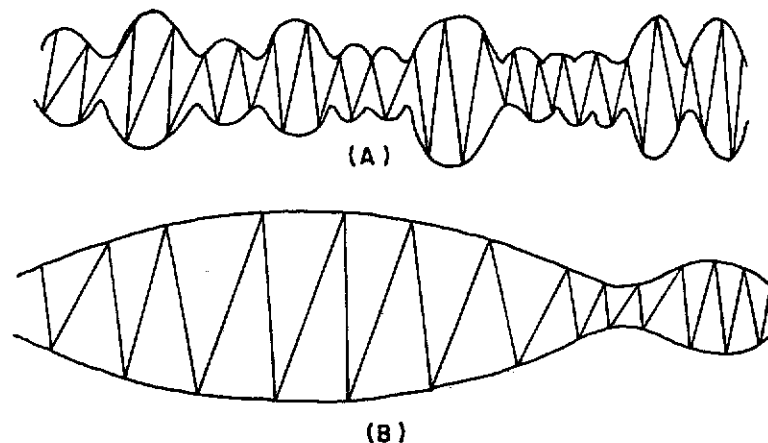


figura 6-8 - Sinal de vídeo (A) alta frequência, de pequenos detalhes (B) baixa frequência, de grandes objetos.

formam a imagem). Observe que frequências muito altas do sinal modulador corresponderão a pequenos detalhes na tela (figura 6-8A), enquanto frequências baixas se referem a grandes objetos (figura 6-8B). A frequência máxima de vídeo é cerca de 3 MHz.

O contorno da modulação sobre os picos de tensão da portadora é denominado "envelope".

No FM ("frequency modulation", modulação de frequência) os picos de tensão da portadora não são alterados, permanecendo em níveis constantes. A ação da modulação se faz na própria frequência da portadora (figura 6-9), provocando desvios para frequências maiores e menores. Um sinal modulador forte

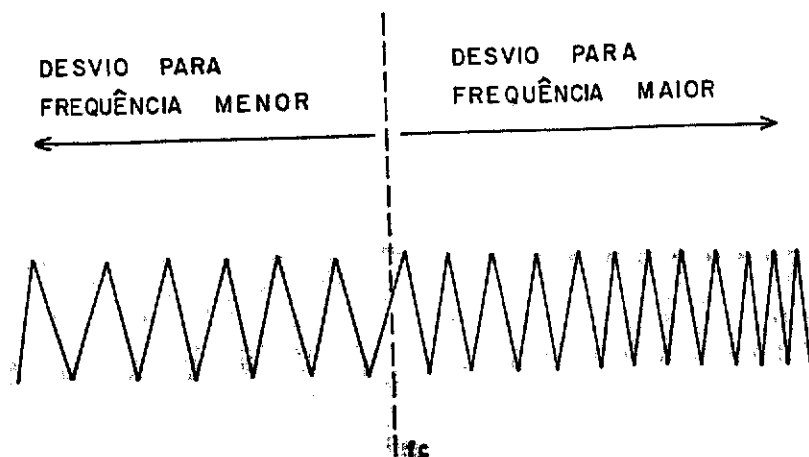


figura 6-9 - Modulação FM.

(barulhento) leva a desvios grandes em torno da frequência central f_c , enquanto sinais fracos (cochi - cho) resulta em desvios pequenos. Sinal modulador de frequência alta (tipo voz feminina) provoca muitos desvios, rapidamente; modulação de baixa frequência (tipo voz masculina) tem menor número de desvios por segundo.

A emissora tem à sua disposição uma largura de banda de 6 MHz (por exemplo, de 60 MHz a 66 MHz, ou de 66 MHz a 72 MHz). A portadora é modulada nos seguintes padrões (figura 6-10):

- O sinal de imagem modula (AM) a frequência 1,25 MHz acima do início da banda.
- O sinal de áudio modula (FM) a frequência 5,75 MHz acima do início da banda.

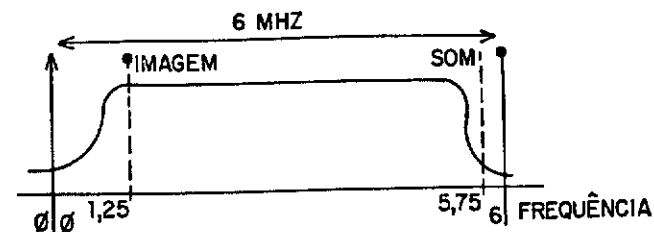


figura 6-10 - Banda de 6 MHz de canal.

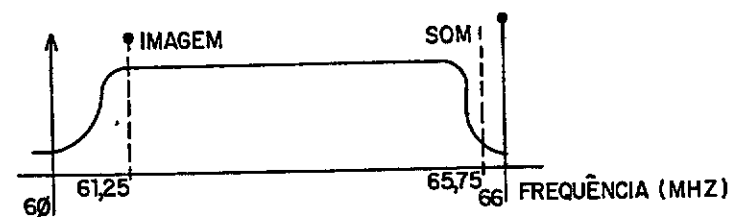


figura 6-11 - Canal 3.

Tomando o canal 3 como exemplo (figura 6-11), temos o início da banda de 6 MHz em 60 MHz, a portadora de imagem em 61,25 MHz e a frequência central de áudio em 65,75 MHz, terminando o canal em 66 MHz. Observe que entre o som e a imagem há 4,5 MHz de separação.

O sinal modulado visto acima é emitido entre os sinais de blanking, com respectivos sincronismos horizontais e verticais, formando o sinal composto de vídeo.

4 - RECEPTOR PRETO E BRANCO

No SINTONIZADOR (figura 6-12) um sistema de bobinas e capacitores são variados até en-

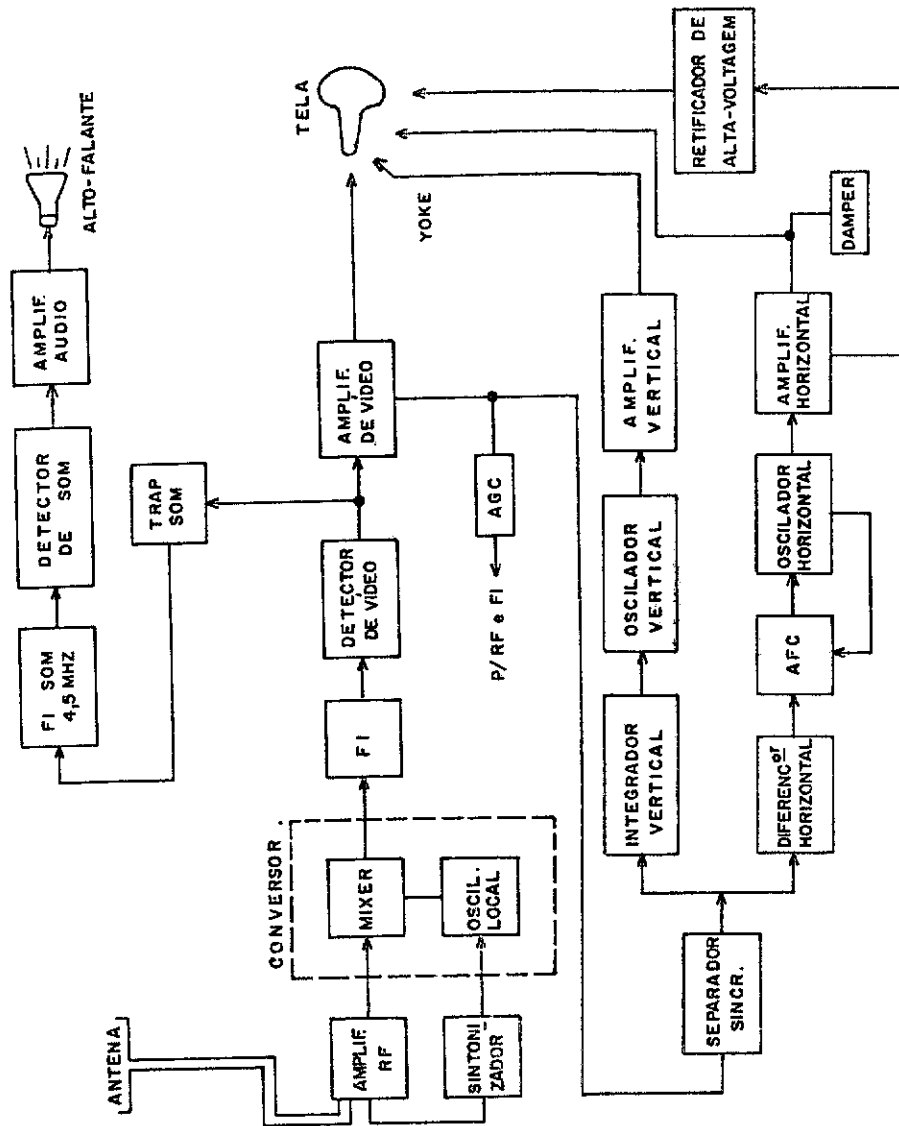


figura 6-12 - Esquema de receptor preto e branco.

tratar em ressonância com a emissora desejada, dando ganho máximo ao seu sinal e quase nenhum aos sinais de outras estações. Essa variação provoca simultaneamente uma alteração no OSCILADOR LOCAL, fazendo-o oscilar 45,75 MHz acima da portadora de imagem (para o canal 3 da seção anterior: 45,75 MHz + 61,25 MHz = 107 MHz). O sinal sintonizado é amplificado no AMPLIFICADOR DE RF e misturado com as oscilações no MIXER (um único estágio englobando o mixer e o oscilador é denominado CONVERSOR). Quando misturamos frequências diferentes um dos sinais resultantes tem a frequência da diferença entre elas. Assim, da mistura entre as saídas do oscilador local e do amplificador de RF resulta sempre FREQUÊNCIA INTERMEDIÁRIA (F.I.) constante, uma para imagem (45,75 MHz) e outra para áudio (41,25 MHz). Ainda no exemplo do canal 3 teríamos:

- Imagem $107 \text{ MHz} - 61,25 \text{ MHz} = 45,75 \text{ MHz}$.
- Áudio $107 \text{ MHz} - 65,75 \text{ MHz} = 41,25 \text{ MHz}$.

FI (frequência intermediária) é o estágio que amplifica somente as frequências intermediárias (de imagem e som), quase extinguindo qualquer outro sinal; isso dá máximo ganho a imagem e áudio, diminuindo ruídos e interferências.

O DETECTOR DE VÍDEO elimina a frequência portadora (agora como FI) e destaca a amplitude do envelope transmitido, que contém as informações necessárias à reprodução de imagem e som no receptor (incluindo-se aí os sincronismos). A sua saída é ramificada para os estágios de áudio, amplificador e vídeo, separador de sincronismo.

Como no detector de vídeo estão presentes duas frequências diferentes (som e imagem) o correrá na saída um sinal correspondente à diferen-

ça entre elas. A imagem tem frequência portadora fixa (45,75 MHz, sistema AM) enquanto o som oscila em torno de uma frequência central (41,25 MHz, sistema FM). A diferença entre elas será 45,75 MHz menos um valor em torno de 41,25 MHz, variando de acordo com a modulação FM (figura 6-13). Ora, isso corresponde a um sinal FM de portadora 4,5 MHz ($45,75 \text{ MHz} - 41,25 \text{ MHz} = 4,5 \text{ MHz}$) modulado pela frequência de áudio (variações em torno de 41,25 MHz). A detecção do som inclui, assim, duas conversões:

- A primeira misturando sinal sintonizado com o oscilador local, produzindo as FIs 45,74 MHz e 41,25 MHz;
- A segunda misturando esses dois sinais, obtendo-se o FI (final) de som de 4,5 MHz.

O TRAP DE SOM ("armadilha de som") é um estágio que filtra a FI de 4,5 MHz, encaminhando-a para circuitos de áudio funcionando exatamente como um rádio receptor FM (apenas faltando os estágios de sintonização e conversão, já realizados anteriormente). O sinal é amplificado na FI DE SOM, o DETECTOR DE SOM elimina a portadora de 4,5 MHz e destaca o sinal modulador, este último é amplificado no AMPLIFICADOR DE ÁUDIO e enviado ao alto-falante.

No detector de vídeo a portadora FI de vídeo (45,75 MHz) é eliminada, permanecendo seu envelope que é o sinal de vídeo contendo informações sobre a imagem. Ele é amplificado pelo AMPLIFICADOR DE VÍDEO e acoplado à grade ou ao cátodo do CRT. Entre esses dois mantém-se uma tensão de aproximadamente - 40 Volts (por exemplo, 160 Volts no cátodo e 120 Volts na grade de controle). Com acoplamento ao cátodo (figura 6-14A) o sinal de vídeo terá de ser mais positivo para ocorrer apagamento

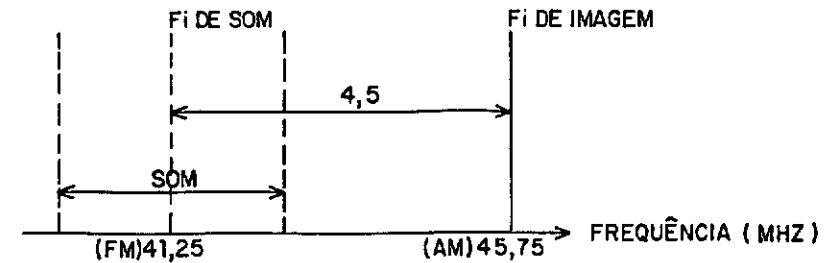


figura 6-13 - FI de som e FI de vídeo.

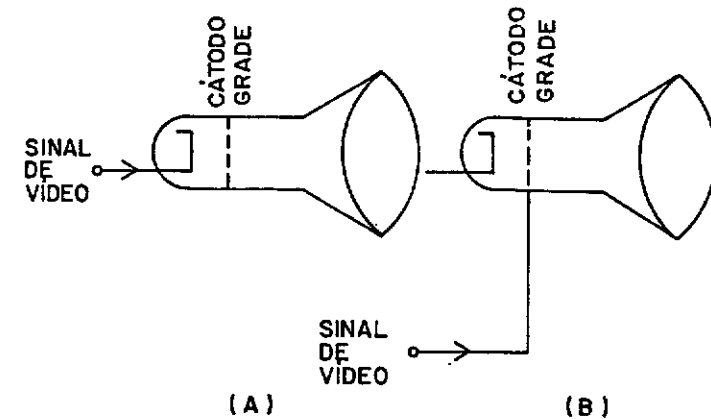


figura 6-14 - Sinal de vídeo (A) acoplado ao cátodo (B) à grade.

ou imagem escura (a grade fica muito negativa em relação ao cátodo, não havendo atração de elétrons-que sempre procuram tensões mais positivas); por outro lado diminuindo a voltagem do sinal de vídeo torna a grade mais positiva em relação ao cátodo, atraindo mais elétrons e resultando um feixe mais intenso,

o que caracteriza imagem branca (mais luminosa). Com acoplamento à grade (figura 6-14B) o sinal de vídeo mais positivo aumenta a atração de elétrons (imagem branca) e se tiver nível de voltagem mais baixo leva a menor atração de elétrons e imagens escuras.

O AGC ("automatic gain control", controle automático de ganho, algumas vezes aparece como CAG) é uma tensão de proporcional ao nível médio do sinal de vídeo, alimentando o RF e a FI. Níveis muito altos de vídeo provocam uma voltagem AGC alta que diminui a amplificação no RF e FI; níveis baixos de vídeo resultam em pouca voltagem de AGC, propiciando maior amplificação no RF e FI. Esse processo mantém o sinal de vídeo constante nos estágios finais, tendo em consequência uma recepção mais uniforme.

Ainda do amplificador de vídeo temos uma saída para o SEPARADOR DE SINCRONISMO, estágio que amplifica apenas sinais de nível maior que do blanking (a "ponta" de sincronismo colocada acima do patamar de apagamento). Na sua saída sinais de baixa frequência (portanto sincronismo vertical, 60 HZ) são filtrados pelo INTEGRADOR VERTICAL e colocados como controle do OSCILADOR VERTICAL, que é amplificado pelo AMPLIFICADOR VERTICAL e excita deflexões verticais no yoke. Sinais de alta frequência (sincronismo horizontal, 15.750 HZ) são selecionados pelo DIFERENCIADOR HORIZONTAL, entrando no AFC ("automatic frequency control", controle automático de frequência, também conhecido como CAF). Este com para o sinal de sincronismo horizontal com a saída do OSCILADOR HORIZONTAL, obrigando o último a operar na mesma frequência de 15.750 HZ do sincronismo. As oscilações são amplificadas pelo AMPLIFICADOR HORIZONTAL e enviadas ao yoke para a respectiva deflexão horizontal. Entre o yoke e o amplificador,

é incluído o DAMPER ("amortecedor"), estágio para uniformização das oscilações.

O RETIFICADOR DE ALTA-VOLTAGEM aproveita o sinal de 15.750 HZ do estágio horizontal para geral a alta voltagem (aproximadamente 20.000 V) necessária à tela (ânodo) para atrair elétrons do cátodo. Para tanto usa-se um transformador (conhecido por FLY-BACK) e um diodo retificador.

O esquema apresentado acima é genérico, omitindo detalhes. Entretanto, serve perfeitamente bem à visualização do processo de transmissão e recepção de televisão preto e branco.

CAPÍTULO 7 – TELEVISÃO A CORES

A televisão a cores se sobrepõe à transmissão preto e branco, completando-a e ao mesmo tempo sendo compatível com ela. Ao contrário da TPB que é igual para todos países, a TVC pode ser implementada por vários sistemas que não são compatíveis entre si.

Os vídeo-games foram projetados nos Estados Unidos, sendo os modelos brasileiros simples cópias dos originais. O sistema da TVC dos norte-americanos é o NTSC ("National Television Systems Committee", Comitê Nacional de Sistemas de Televisão). Assim, a geração de imagens pelos vídeo-games segue os padrões daquele sistema, que é diferente do nosso (PAL - "phase alternating by line", alternância de fase por linha), sendo necessário um estágio - CONVERTSOR PAL - que altere o sinal gerado no vídeo-game para que ele possa ser processado por receptor brasileiro de TVC.

Apresentaremos inicialmente os princípios do NTSC e em seguida do PAL-M, destacando as diferenças e circuitos necessários à compatibilização entre os dois.

1 - PM - MODULAÇÃO DE FASE

Dissemos anteriormente que a imagem é AM e o som FM. A cor terá um terceiro processo - PM

("phase modulation", modulação de fase). Portanto uma compreensão do que é fase é fundamental.

Dividindo o círculo em 4 partes iguais (figura 7-1A) obtemos os ângulos 0°, 90°, 180° e 270°. Observe que 0° coincide com 360°, e que 90° e 270° são opostos entre si, bem como 0° e 180°. O sinal pode ser associado a um círculo (figura 7-1B). O tempo que ele demora para repetir-se (voltar ao mesmo nível inicial) corresponde a 360° e é chamado "período". Dividindo o período em 4 partes iguais, tal como no círculo, teremos os ângulos 0, 90°, 180°, 270°, 360°. Na figura 7-1B, no tempo 0 0° indica o início do período com o sinal começando a aumentar a amplitude (de voltagem ou corrente); no tempo 1, 90°, foi atingida a amplitude máxima, que começa a decrescer; no tempo 2, 180°, atingiu-se 0 e está em decréscimo, partindo para amplitude negativa; no tempo 3, 270°, ocorre a maior amplitude negativa, e tem início um crescimento em direção a valores positivos; no tempo 4, 360° aparece novamente amplitude 0, com o sinal começando a aumentar, exatamente como no início do período a 0°.

Tomando outro sinal com a mesma frequência (figura 7-1C) e comparando sua amplitude com a figura anterior nos 4 ângulos notaremos que ele se encontra avançado 1 tempo - ou 90°. Verifique que a amplitude máxima ocorre imediatamente no início 0 na figura 7-1C, enquanto isso só é registrado em 90° na figura 7-1B. Da mesma forma temos amplitude 0 aos 90°, enquanto no 1º exemplo esse valor será obtido aos 180°. Ao longo de tempo em toda a extensão dos sinais permanecerá sempre uma diferença de 90° entre eles. Dizemos que há uma DEFASAGEM de 90°, ou que a fase em 7-1C está 90° avançada sobre a referência de 7-1B.

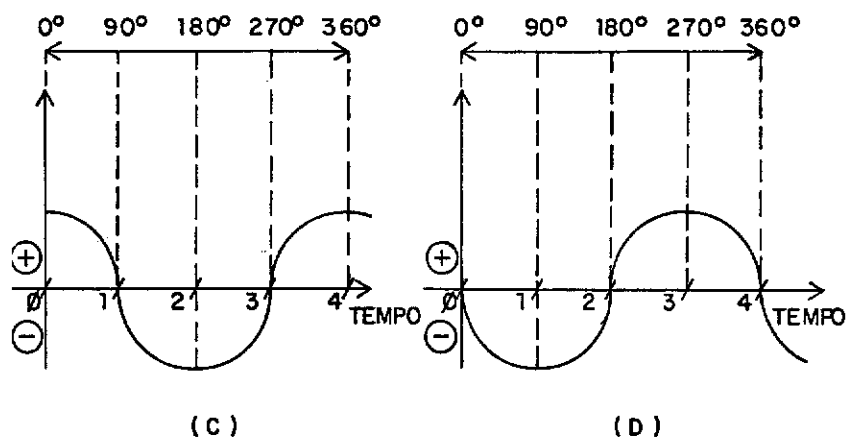
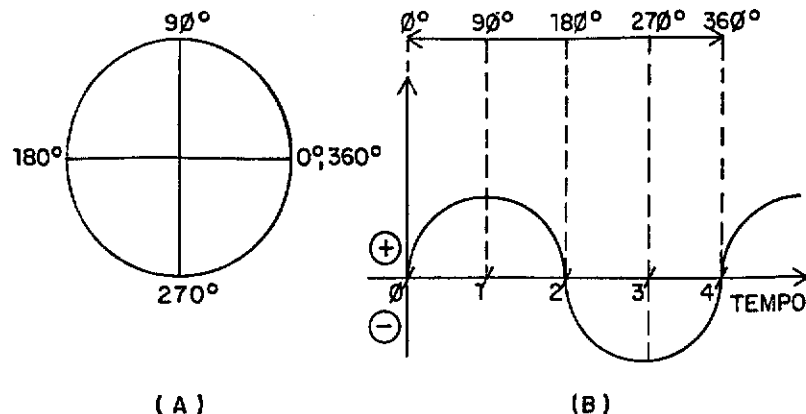


figura 7-1 - (A) divisão do círculo em quadrantes
(B) fases dosinal (C) defasagem de 90°
(D) defasagem de 180°.

Na figura 7-1D temos uma defasagem de 180° em relação a 7-1B. Note que amplitude 0 tendendo a valores negativos ocorre logo no início no 3º exemplo, enquanto só aparece em 180° no 1º exemplo. A diferença de 180° permanece para qualquer ângulo de comparação entre os sinais.

Sinais com defasagem de 90° são ditos em QUADRATURA, e de 180° OPOSTOS. Obviamente, é possível obter qualquer defasagem intermediária (33°, 45°, 179°, etc), bastando deslocar o gráfico na extensão desejada.

Eletronicamente, a maneira de se obter sinais defasados é tomar uma fonte osciladora (figura 7-2) e emitir seu sinal como referência. Da mesma fonte tira-se uma entrada para um dispositivo onde sua velocidade de propagação é menor, atrasando-o (conforme o meio os sinais se propagam com velocidades diferentes). O resultado será dois sinais de mesma frequência, mas defasados. A escolha do tempo de atraso (pela seleção do material de propagação) determina o ângulo de defasagem.

2 - TRANSMISSÃO A CORES

Na TVC a informação sobre cor é denominada CROMINÂNCIA-C e transmitida como defasagem. Um sinal de referência - BURST - de 8 a 11 ciclos e frequência 3,58MHZ é acrescentado na parte traseira do sincronismo horizontal (figura 7-3) e a crominância enviada como sinal também na frequência de 3,58 MHZ. No receptor a fase da crominância C é comparada com o burst, resultando daí a informação de cor, conforme a figura 7-4. Quando se deseja uma cor escura a crominância é transmitida 180° defasada do burst; para o vermelho usa-se o ângulo de 90°. A amplitude do sinal de crominância indica a saturação da cor (vermelho mais vívido ou mais pálido; verde

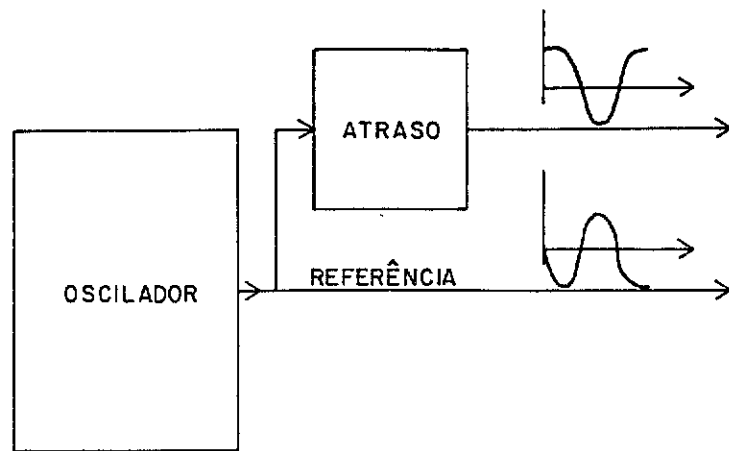


figura 7-2 - Dispositivo para geração de sinais defasados.

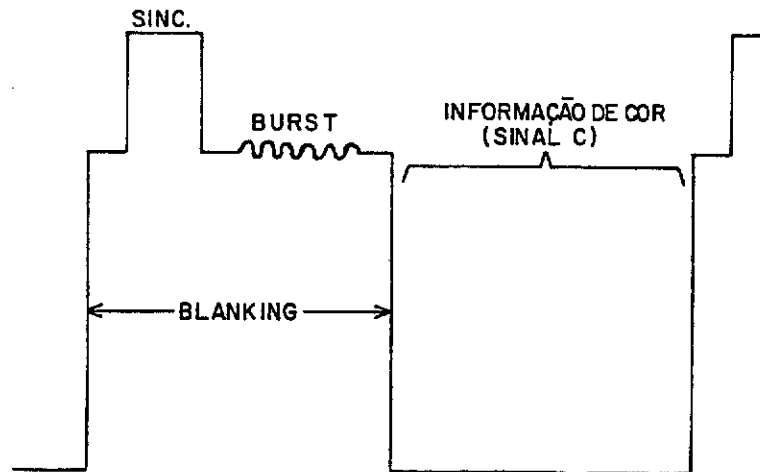


figura 7-3 - Burst.

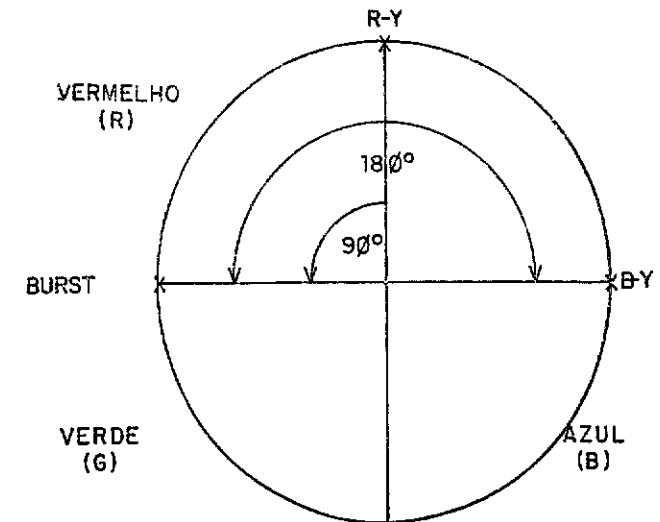


figura 7-4 - Fase das cores na TVC (sistema NTSC).

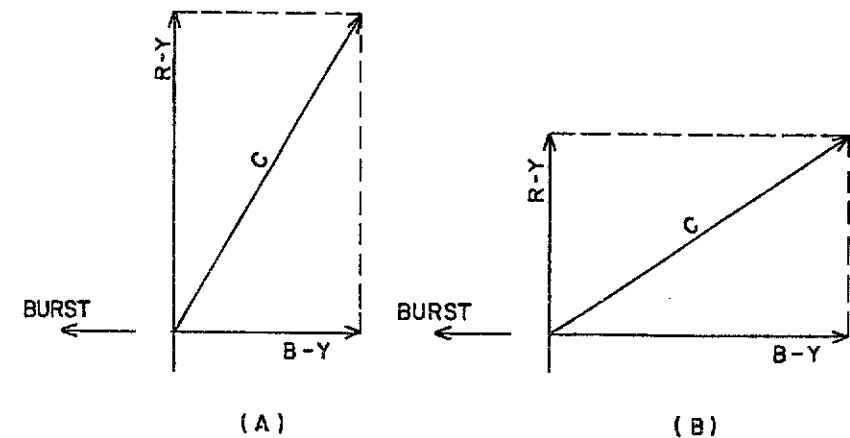


figura 7-5 - Decomposição vetorial da croma (A) muito avermelhado, pouco azulado (B) muito azulado, pouco avermelhado.

esbranquiçado ou bem forte).

A crominância é transmitida como uma composição vetorial do azul e do vermelho (figura 7-5). Quanto maior a defasagem entre C e o burst maior será o componente B-Y (azulado) e menor R-Y (a vermelhado). No caso extremo, com C totalmente oposto (180°) ao burst, teremos somente B-Y e nada de R-Y. Emitindo C 90° defasado do burst haverá só R-Y e não existirá B-Y.

Dissemos que a transmissão é uma composição de azul e vermelho apenas para simplificar. Na verdade a câmara de tomada da imagem tem três filtros ópticos que recebem a luz (uma composição de cores) e separam o vermelho R ("red"), azul B ("blue") e verde G ("green"), conforme a figura 7-6. Através de uma rede de resistores as cores são misturadas na seguinte proporção:

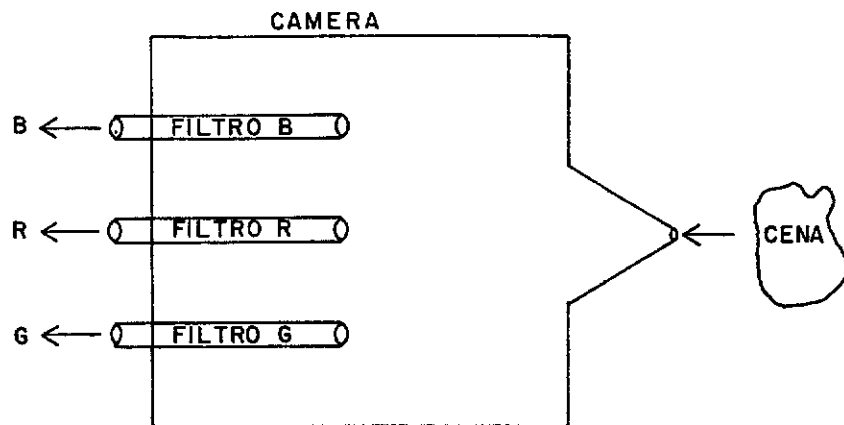


figura 7-6 - Separação das cores por filtro óptico.

$$Q = 21\% R - 53\% G + 31\% B$$

$$I = 60\% R - 28\% G + 32\% B$$

Além do mais, utilizando-se os princípios de luminosidade relativa das cores, obtém-se o sinal de LUMINÂNCIA Y (correspondendo ao sinal de transmissão branco e preto) pela mistura:

$$Y = 30\% R + 59\% G + 11\% B$$

Os sinais Q e I são mantidos 90° defasados entre si (em quadratura, daí as letras Quadratura e In phase) e modulam a subportadora de cor (figura 7-7).

A crominância C é somada à luminância Y nos estágios finais. Isso permite que receptores preto e branco recebam e reproduzam Y normalmente, embora não consigam processar C. Assim, a transmissão a cores fica compatível com o sistema preto e branco. No receptor a cores o sinal Y é colocado simultaneamente no cátodo e na grade, cancelando-se, de modo que somente a crominância atua efetivamente no TVC.

O sinal de vídeo incluindo a luminância Y, a crominância C e os sincronismos modulam a portadora da emissora, usando o mesmo canal de 6MHz. O gráfico do sinal composto de vídeo é visto na figura 7-8. Para o canal 4, por exemplo, teríamos:

BANDA: de 66MHz a 72MHz

PORTADORA DE IMAGEM: 67,25MHz

SUBPORTADORA DE COR: 70,83MHz

SOM: 71,75MHz

Esse esquema tem interesse mais teórico, pois vídeo-games atuam como estações transmissoras de TVC. Devem portanto preencher todos pa-

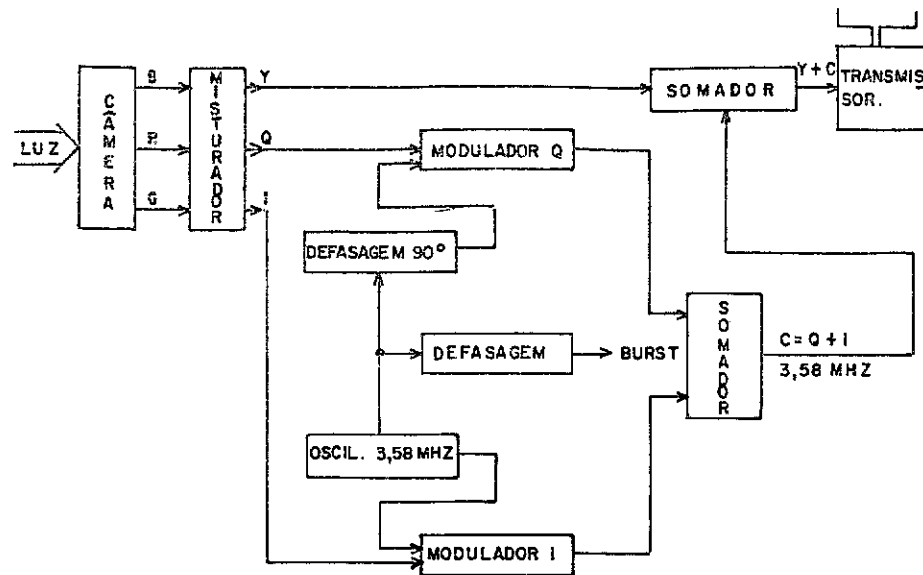


figura 7-7 - Codificação do sinal composto de vídeo.

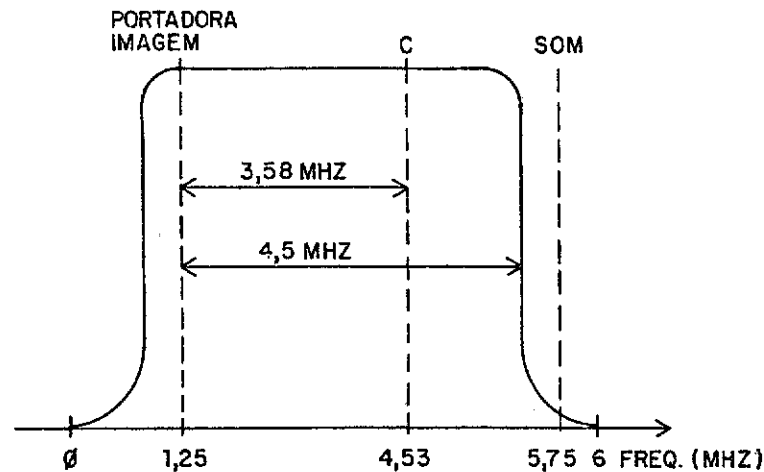


figura 7-8 - Banda de 6MHz de canal de TVC.

drões vistos. Os canais usados por vídeo-games são o 2, 3 ou 4, geralmente havendo uma chave que permite ao usuário optar entre eles. As frequências desses canais são apresentadas abaixo:

	CANAL 2 (EM MHZ)	CANAL 3 (EM MHZ)	CANAL 4 (EM MHZ)
BANDA	54-50	60-66	66-72
PORTADORA DE IMAGEM	55,25	61,25	67,25
SUBPORTADORA DE COR	58,83	64,83	70,83
SOM	59,75	65,75	71,75

3 - RECEPTOR TVC

A tela do receptor TVC é composta de trios de material fluorescente, nas cores vermelho (R), azul (B), e verde (G). Tal como no branco e preto existem 525 linhas, com cerca de 500 desses trios por linha. Cada cor tem seu próprio canhão excitador que não interfere nas outras, de maneira que poderíamos considerar a TVC como 3 televisores superpostos - um azul, um vermelho e outro verde. Entretanto, devido ao pequeno tamanho dos trios e à pequena separação entre os pontos que os compõem, nossa vista enxerga-os como uma só imagem, "somando" as cores e luminosidade (com uma lente forte próxima à tela percebe-se claramente os trios coloridos; experimentalmente). Para se obter a sensação de qualquer cor basta colocar percentagens adequadas de R, G, B. Em especial, o branco é o resultado dessas cores nas mesmas percentagens (100%R + 100% B + 100% G) e o preto a ausência de todos (apagamento), como no TPB.

Os estágios iniciais do receptor TVC (figura 7-10) são semelhantes aos do TPB, exceto pelo AFT ("automatic fine tuning", sintonização fina automática). A existência deste estágio é justificada pela necessidade de perfeita sintonização, evitando que a subprodutora de cor seja recebida com fre-

do as outras (lembre-se que há dois sinais de 3,58 MHz: a informação de cor propriamente dita e o burst). Sua saída é para os demoduladores e para o separador de burst.

O SEPARADOR DE BURST é chaveado pelo sincronismo horizontal, amplificando apenas durante o retraço. Deste modo o sinal C é bloqueado (pois é apresentado só no traço) e o burst amplificado (já que é parte posterior do blanking, durante o retraço).

O burst, devidamente separado, sincroniza o AFPC ("automatic frequency and phase control", controle automático de frequência e fase) que por sua vez controla um sistema oscilador de 3,58 MHz, composto por um CRISTAL e um circuito OSCILADOR DE 3,58 MHz. O controle é feito através de níveis de voltagens: se as oscilações passam de 3,58 MHz o AFPC aumenta sua voltagem para o oscilador, diminuindo a frequência do sistema.

Outra saída do AFPC é para o INIBIDOR DE COR. Quando a transmissão é preto e branco não há burst e isto acarreta uma voltagem no AFPC suficiente para excitar o inibidor de cor, que mantém o amplificador de croma em corte (não funciona). Assim, não há amplificação de cor, somente a luminância Y, evitando que no receptor a cores haja interferência da crominância em programas preto e branco. Durante transmissão a cores a voltagem do AFPC não é suficiente para ativar o inibidor.

Os moduladores recebem o sinal de crominância do amplificador de cor e fazem sua modulação (eliminação da portadora e saída apenas para o sinal modulador). Para tanto o DEMODULADOR B-Y precisa receber sinal em fase do oscilador 3,58 MHz (B-Y está defasado 180° do burst, ou seja, é a

mesma fase mas com sinal trocado). O DEMODULADOR R-Y recebe a saída do oscilador através de uma DEFASAGEM de 90° do burst.

Os sinais dos demoduladores são amplificados no AMPLIFICADOR B-Y e AMPLIFICADOR R-Y. Estes dois são entradas para o SOMADOR G-Y que, processando-os devidamente, obtém a cor esverdeada (a partir de duas cores é possível obter qualquer outra). Os três sinais, B-Y, G-Y e R-Y são encaminhados para os cátodos ou grades dos respectivos canhões, excitando a parte dos trios RGB correspondente a cada um.

4 - SISTEMA PAL

No sistema NTSC que acabamos de ver a possibilidade de erro nas cores é grande, pois o sistema depende exclusivamente da sintonização exata na fase da subportadora de cor. Qualquer falha na fase desta resulta em cores erradas, sem possibilidade de correção.

O PAL foi desenvolvido para corrigir essa deficiência do NTSC. O burst é emitido em ângulo de 135° com B-Y numa linha e 225° na linha seguinte (figura 7-11), alternadamente, portanto uma variação de 90°. R-Y também alterna de linha para linha, mas em ângulo de 180°. B-Y não sofre alterações. O resultado geral é que desvios de fase da crominância C são automaticamente compensados, mantendo-se uma "média". A figura 7-12 ilustra o processo.

R-Y, alternando a cada linha juntamente com o burst, é adicionado a B-Y para formar o sinal de crominância C que é transmitido e captado, a qual nós representamos por Cpr - C PRESENTE. No receptor uma linha de atraso retarda a crominância C

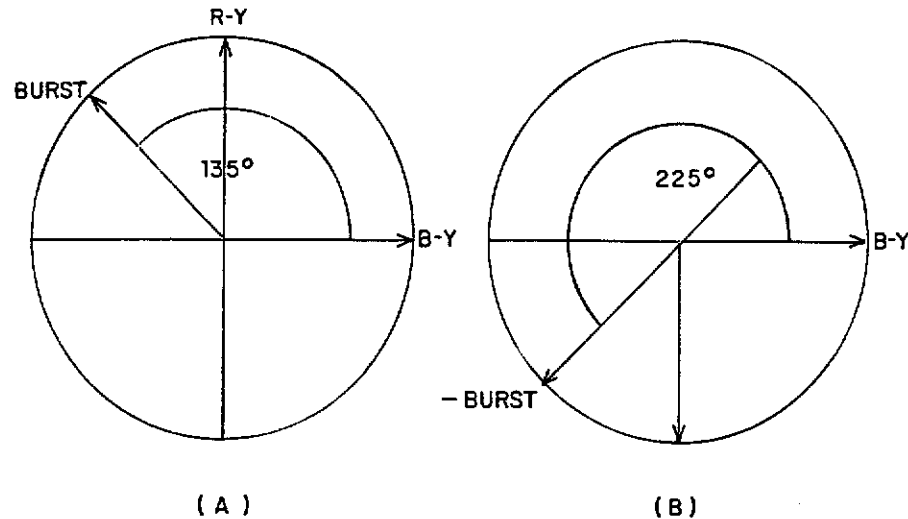


figura 7-11 - Fase do burst e de R-Y (A) em uma linha horizontal (B) na linha seguinte.

por 64 microsegundos-tempo de varredura horizontal-de maneira que teremos ao mesmo tempo o sinal da linha presente (Cpr) e o da linha passada (Catr - C ATRASADA). No circuito B-Y a linha passada é somada com a linha presente (Catr+Cpr), resultando B-Y puro, sem R-Y. No circuito R-Y a linha passada é invertida 180° (Catrinv-C ATRASADA INVERTIDA) antes de ser somada à linha presente, formando-se R-Y. Neste último o sinal R-Y será dado alternadamente em fases opostas, a cada linha.

Na figura 7-13 vemos os estágios de cor do sistema PAL (os demais são omitidos pois são exatamente iguais ao NTSC). O separador de burst, CAF e inibidor de cor operam como já vimos anteriormente cortando o amplificador de crominância em programas

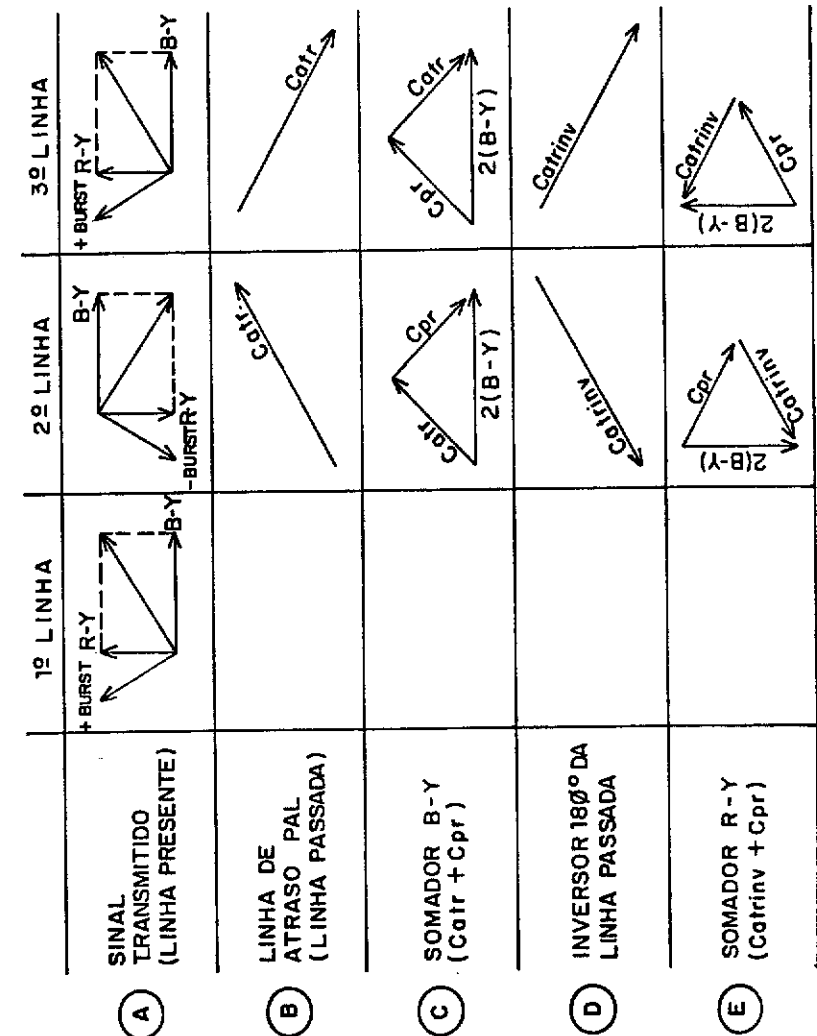


figura 7-12 - Fases e processamento de sinais na TVC(PAL).

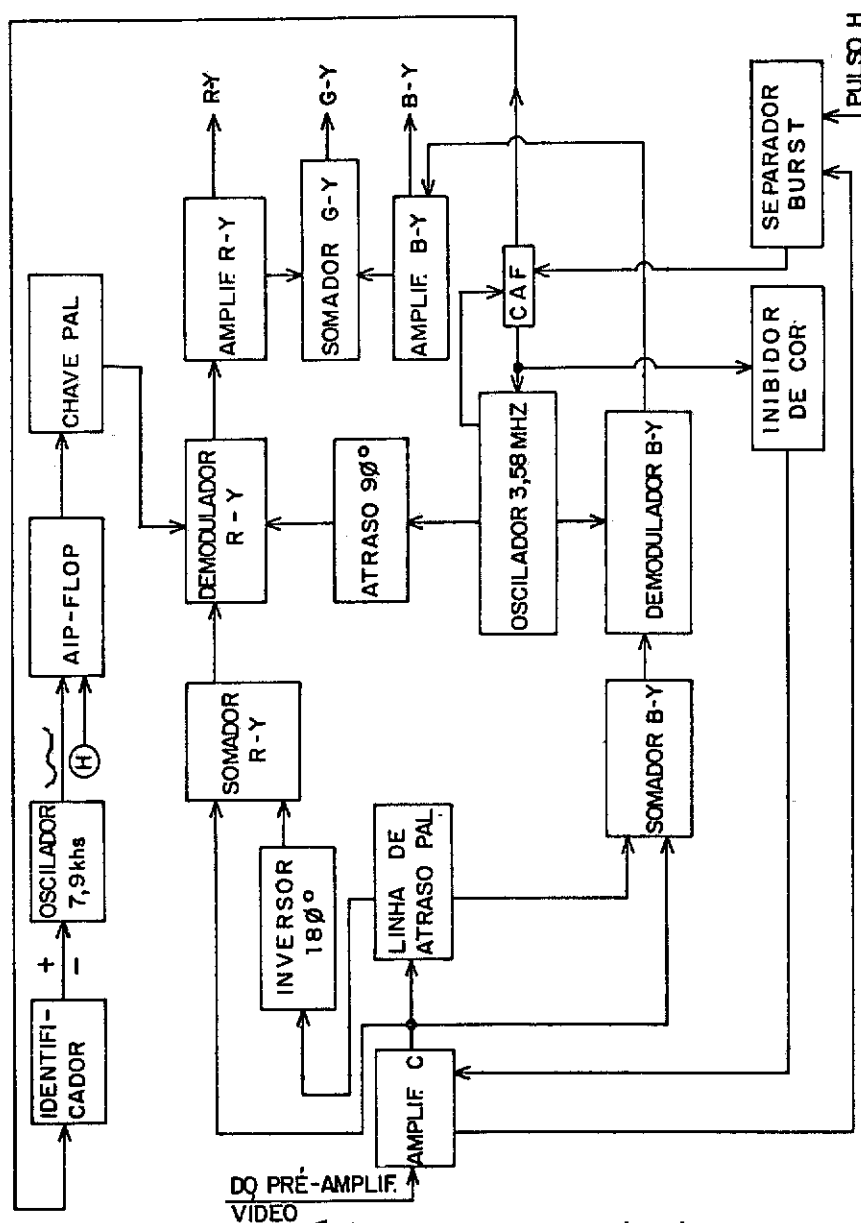


figura 7-13 - Estágios de cor no TVC(PAL).

preto e branco e sincronizando o oscilador. Este fornece sinal de frequência 3,58MHz em fase para o demodulador B-Y e defasado 90° para o demodulador R-Y. A crominância C é tirada do amplificador para os somadores. A linha de atraso PAL, que fornece o sinal de linha passada, tem saída direta para o somador B-Y, enquanto para o somador R-Y é interposto um inversor de 180°. Como R-Y será obtido alternadamente em fases opostas, a cada linha, existe um circuito para "desinvertê-lo", começando no CAF. Ali o burst provoca pulsos de saída positivo ou negativo (figura 7-14), conforme estiver em 135° (linhas pares) ou 225° (linhas ímpares). Esses pulsos chegam ao identificador que os amplifica e sincroniza um oscilador senoidal de 7,9KHZ (metade da frequência de varredura horizontal) com a mesma fase do burst. O flip-flop é chaveado pelos pulsos do fly-back (para retraço horizontal) e deveria mudar de estado a cada linha. Entretanto, nas linhas ímpares a senóide recebida do oscilador 7,9 KHZ tem voltagem negativa suficiente para bloquear o flip-flop, impedindo que ele mude de estado. Assim, a chave PAL receberá pulsos somente nas linhas pares, e nelas chaveará o demodulador para "desinverter" R-Y.

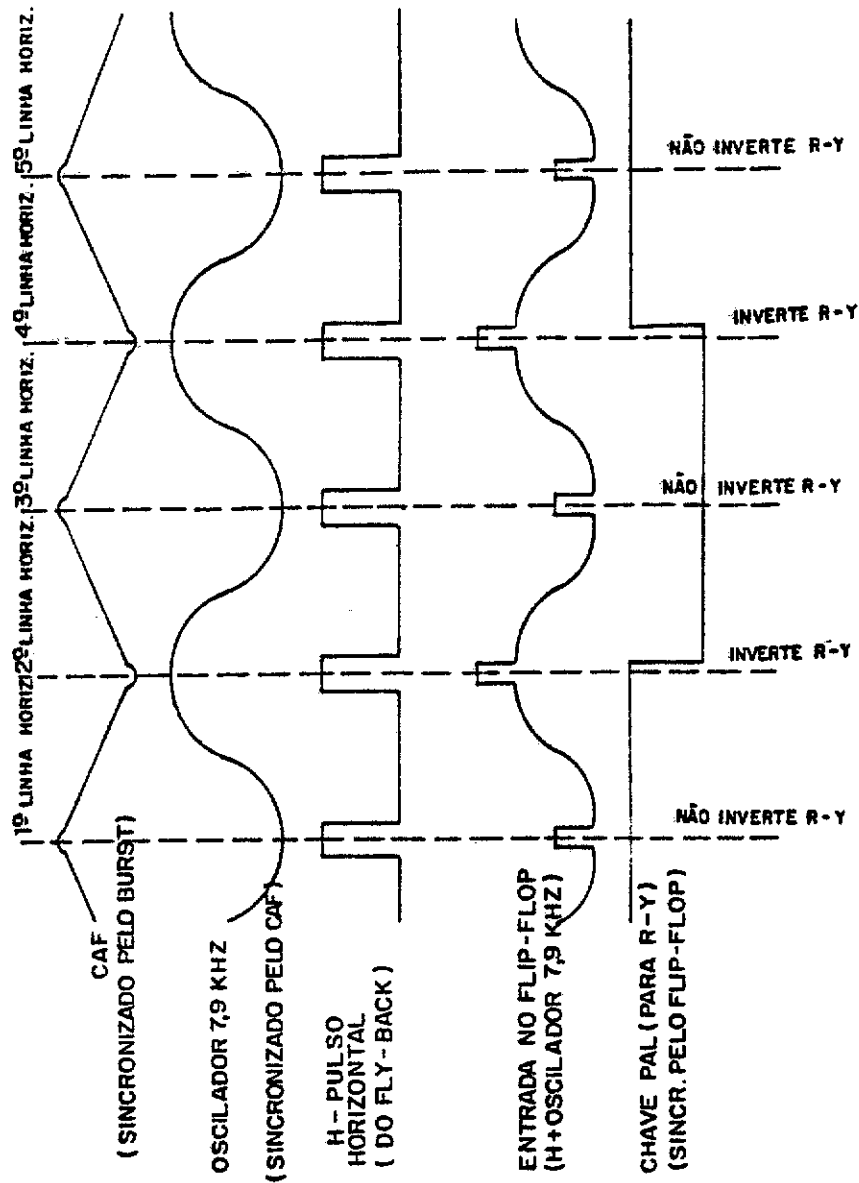


figura 7-14 - Sincronismos na chave PAL.

CAPÍTULO 8 – GERAÇÃO DE IMAGEM E SOM

Geração de imagens coloridas é a característica fundamental do vídeo-game. Para realizá-la o aparelho deve equiparar-se a uma emissora de TVC. Gerenciar as imagens de modo a transformá-las num jogo é obra de Informática, ou seja, um programa de computador.

Apresentaremos os princípios de geração de imagens e sons, tópico extremamente complexo e dependente de conhecimentos prévios de Televisão, Eletrônica Digital e Programação. O leitor deve ter um pouco de paciência neste capítulo e, sempre que necessário, interrompê-lo para rever conceitos anteriores.

Para tornar a matéria mais assimilável adotamos alguns padrões (número de células, capacidade da RAM de vídeo, tipos da ROM - gerador gráfico, etc) que possibilitem uma exposição mais detalhada e concreta. Entretanto, cada marca de vídeo-game terá seus padrões específicos.

1 - MAPEAMENTO DA TELA

O vídeo-game trata a tela como uma matriz de colunas e linhas, formando quadrículas denominadas "células". Para efeito de explicação consideraremos um esquema de 16 colunas x 4 linhas, portanto 64 células, como ilustrado na figura 8-1 (num ví-

LINHAS \ COLUNAS																
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
01	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
10	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
11	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63

figura 8-1 - Tela de 64 células, 16 colunas x 4 linhas.

deo-game real é mais provável encontrarmos 64 colunas x 16 linhas).

Numerando com binários, escrevendo primeiro os dois bits da linha e depois os quatro bits da coluna obteremos um mapeamento de endereços. A sequência de numeração é semelhante à leitura da página de livro: da esquerda ao alto até o fim da linha, repetindo e seguindo as linhas de cima para baixo até atingir a última posição na direita em baixo. Observe que o número binário formado por linha e coluna é igual ao valor decimal da posição da célula: 00 0000B = 0, 00 0001B = 1, 00 0010B = 2, -----, 11 1111B = 63.

2 - CÉLULAS E FIGURAS

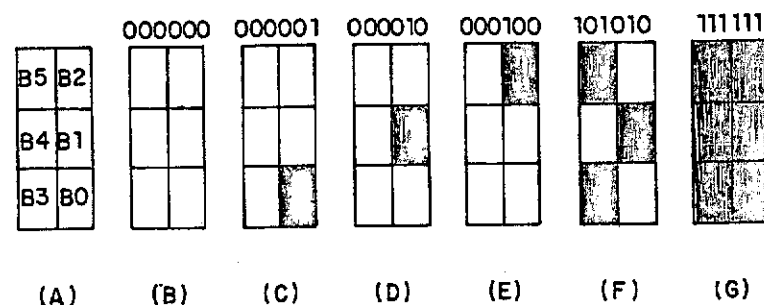


figura 8-2 - Codificação de células (a) posição dos bits na célula (b) até (g) exemplos com respectivos códigos.

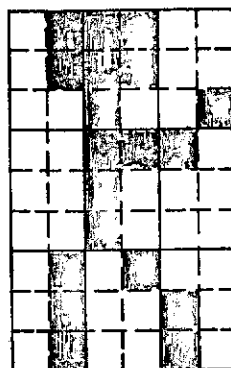
A célula é dividida em 6 segmentos, a cada um deles correspondendo um bit na sequência b5 b4b3b2b1b0 (figura 8-2-a). Quanto o bit = 1 o segmento é colorido, caso contrário é apagado. Assim, a célula codificada 000100 (figura 8-2e) tem o bit b2 em 1, significando que seu segmento no alto à direita é colorido e os outros apagados. A combinação dos 6 bits permite formar $2^6 = 64$ tipos de células.

As figuras do jogo são construídas de um bloco de células. Na figura 8-3a vemos 9 células formando um "homem". Observe em 8-3b o mesmo bloco representado pelo código das células.

Esse processo permite "desenhar" as

000110	111110	000001
000000	111100	100000
000111	000100	011000

(B)



(A)

figura 8-3 - Figura de homem (a) bloco de 9 células
(b) código das células.

mais variadas figuras pela simples escolha de alguns dos 64 tipos de células e ajuntando-as em blocos. Às vezes uma única célula representa a figura. Por exemplo, a célula 000010 (figura 8-2-d) pode ser uma bolinha de futebol, ou uma bala de canhão.

3 - PIXEL

Até aqui apresentamos a célula do ponto de vista teórico. Na prática ela será um pequeno retângulo na tela sendo varrida pelo feixe eletrônico.

Na sua trajetória o feixe eletrônico precisa "saber", a cada instante, se aquele ponto na tela é colorido ou apagado. A célula é então di-

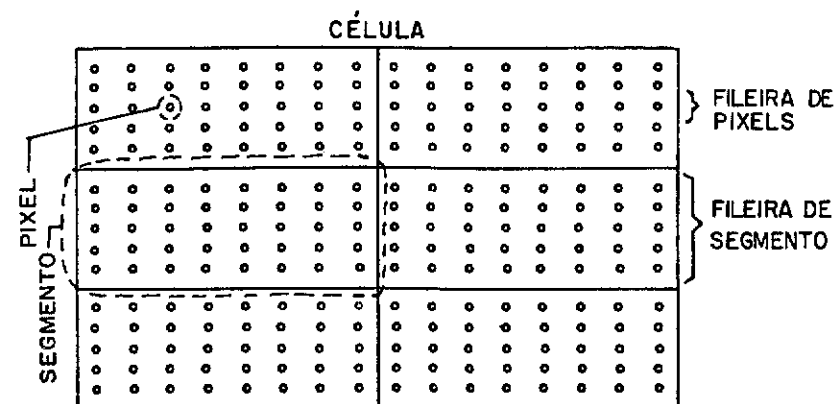


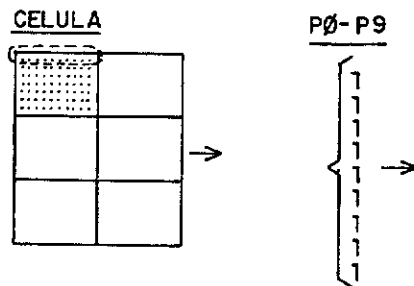
figura 8-4 - Célula de 240 "pixels"

vidida em "pixels" ("picture elements", elementos de imagem), que correspondem ao estado colorido/apagado possível para o ponto na tela.

Na figura 8-4 é vista uma célula com - posta de 6 segmentos, cada um formado pela matriz 8 colunas x 5 fileiras de pixels. Observe que o mesmo tipo de célula (da figura 8-2) pode ter qualquer número de pixels. Quanto maior for o número deles melhor será a "resolução", isto é, a tela apresentará a figura mais detalhadamente.

4 - ROM GERADOR GRÁFICO

Os 64 tipos de células são gravados numa ROM especial chamada "gerador gráfico". O conteúdo da ROM-gerador gráfico é o conjunto de pixels da aquele tipo de célula, fazendo-se: pixel colorido = 1,



REGISTRO DE DESLOCAMENTO

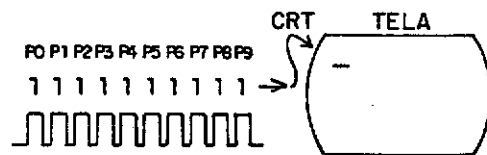
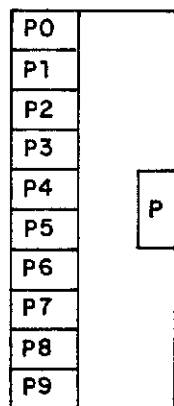


figura 8-5 - Apresentação na tela de uma fileira da célula (a) célula 100000 (b) bits P0-P9 referentes aos pixels (c) entrada paralela dos bits (d) saída serial dos bits (e) efeito visual da fileira na tela.

pixel apagado = 0.

O código da célula funciona como endereço para a ROM do gravador gráfico. Os 6 bits (b0-b5 na figura 8-2A) determinam 1 entre 64 posições onde estão gravadas todas as informações sobre aquele tipo de célula.

Para apresentação da célula na tela são seguidos os seguintes passos (figura 8-5):

- 6 bits de endereço selecionam um dos 64 tipos de célula gravados no gerador gráfico.
- a célula endereçada apresenta 1 fileira de pixels na forma de bits.
- os bits apresentados são carregados em paralelo (todos de uma só vez) no registro de deslocamento.
- o registro de deslocamento armazena o conjunto de bits e passa a apresentá-los um de cada vez, em série, ao CRT (tubo de raios catódicos, do receptor de TV).
- o bit nível 1 recebido pelo CRT é um sinal de 5 Volts que, acoplado à grade de controle, provoca condução do feixe eletrônico, resultando num pixel colorido na tela; o bit nível 0 é zero Volt na grade, não havendo condução do feixe eletrônico e em consequência pixel escuro na tela.

5 - ESTÁGIO DE GERAÇÃO DA IMAGEM

Recapitulemos o que vimos até agora :
 (1) a tela é mapeada em quadrículas denominadas células, dispostas em colunas e linhas (geralmente 64 x 16); (2) cada célula é subdividida em segmentos (geralmente 6); (3) colorindo alguns segmentos e mantendo apagados outros obtém-se tipos de células (64 tipos, se são usados 6 segmentos); (4) um bloco agrupando vários tipos de células dá a impressão visual

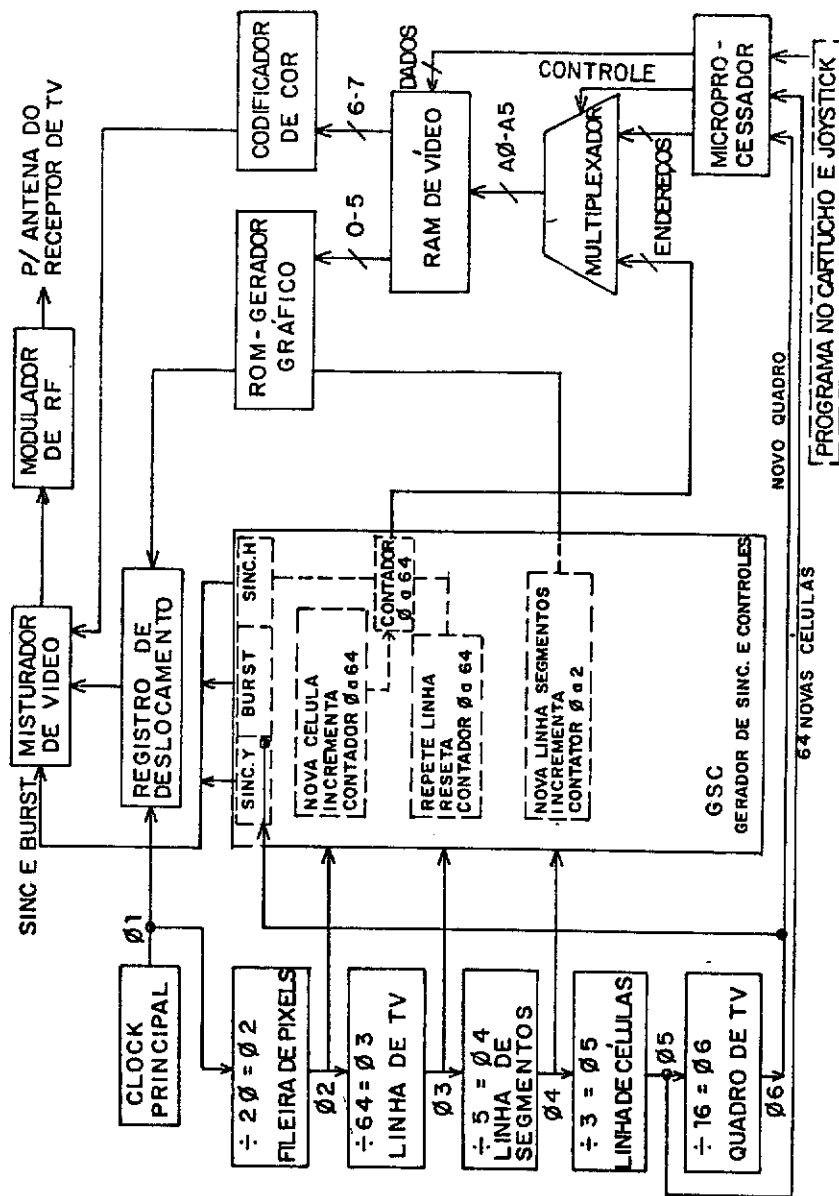


figura 8-6 - Estágio de geração da imagem.

de uma figura; (5) dentro de cada segmento de célula encontramos o pixel que informa ao feixe eletrônico se ele deve ficar aceso - ponto colorido na tela - ou apagado; (6) gerador gráfico é uma ROM cujo conteúdo 1 ou 0 corresponde ao pixel colorido ou apagado.

O leitor que analisar a figura 8-5 constatará que esse complexo esquema apresentado só funcionará se houver sincronismo perfeito entre o endereço da célula e a varredura da tela pelo feixe eletrônico. A figura 8-6 mostra em detalhes o estágio de geração da imagem no vídeo-game, encarregado de controlar a operação e baseado num mapeamento de tela de 64 colunas x 16 linhas de células de 2 x 3 segmentos, estes últimos com 10 colunas x 5 linhas de pixels (figura 8-7).

Inicialmente o microprocessador assume o controle do multiplexador e escreve na RAM, em ordem crescente de endereços, 64 bytes de dados correspondentes aos códigos das 64 células a serem apresentadas no topo da tela (de cada byte 6 bits é o código da célula mais 2 bits de informação sobre a cor). Terminada a escrita o microprocessador vai a atender outros estágios, passando o controle do multiplexador ao gerador de sincronismos e controles - GSC.

Assumindo o controle o GSC aciona seu contador de 64 endereços que apresenta à RAM de vídeo o endereço inicial zero, onde o microprocessador escreveu o código da fórmula zero a ser mostrada no topo à esquerda da tela. A RAM envia à ROM-gerador gráfico o conteúdo da posição endereçada, que é o código da célula zero (mais 2 bits de cor). A ROM-gerador carrega em paralelo no registro de deslocamento os 20 bits da primeira fileira de 20 pixels da célula, correspondentes à primeira fileira dos segmentos b5 e b2 (figura 8-2a e 8-5a).

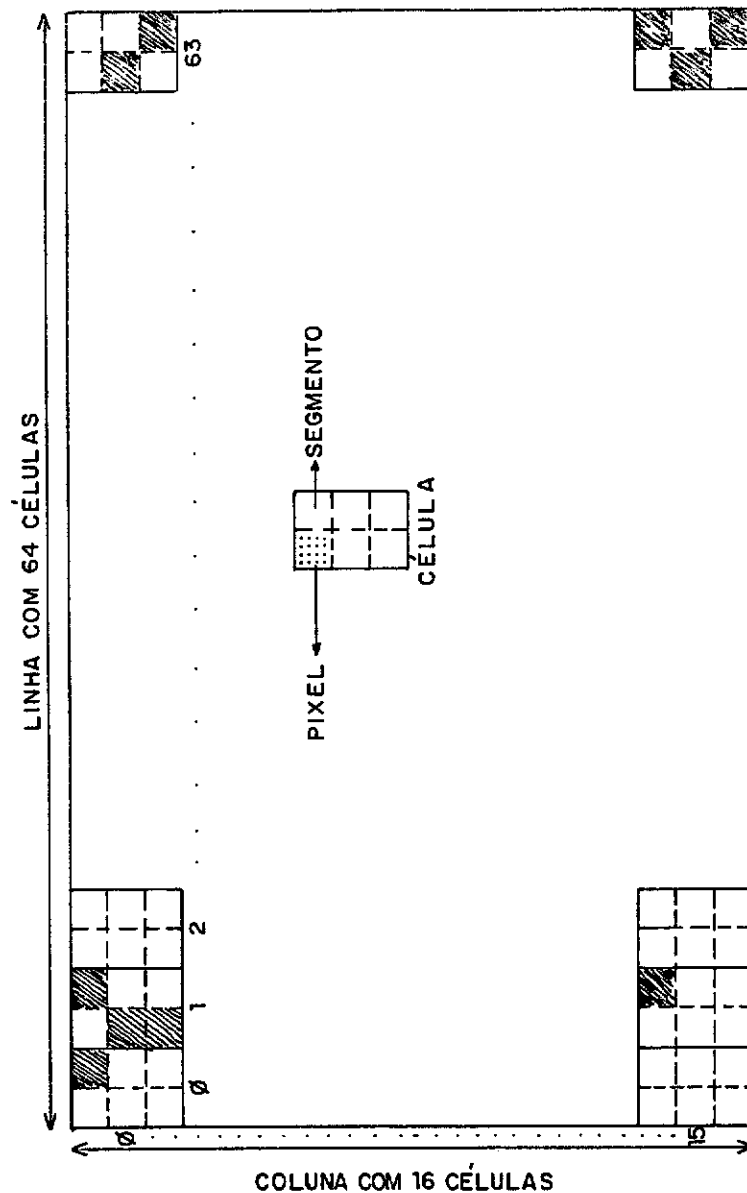


figura 8-7 - Tela mapeada 64 colunas x 16 linhas de células de 2 x 3 segmentos; cada segmento tem 10 colunas x 5 linhas de pixels.

O clock envia sinais $\phi 1$ para o registro de deslocamento, fazendo-o deslocar em série 1 bit (dos 20 recebidos do display gráfico) a cada pulso. O circuito que gera $\phi 1$ é feito de tal maneira que ele esteja em perfeita sincronização com a varredura do feixe eletrônico. Deste modo o bit que o registro de deslocamento entrega ao misturador de vídeo corresponde exatamente à informação que o feixe eletrônico deve mostrar na tela naquele instante (que é o 1º pixel da 1ª fileira do segmento b5 da célula zero).

O misturador de vídeo recebe o bit do registro de deslocamento - bit 1 = 5 Volts, bit 0 = 0 Volt - e mistura-o com os sinais de sincronismos e a saída do codificador de cor, formando o sinal composto de vídeo que entra no modulador de RF para modular uma portadora; a partir daí o sinal é exatamente como de uma emissora de TV (geralmente canal 2 ou 3).

Depois de 20 pulsos $\phi 1$ todos os bits referentes à 1ª fileira dos segmentos b5 e b2 da célula zero já foram deslocados serialmente. Neste momento o divisor de frequência por 20 (para cada 20 pulsos recebidos ele emite 1) envia o sinal $\phi 2$ para GSC, indicando que a fileira de pixels da próxima célula deve ser carregada no registro de deslocamento. O GSC incrementa o contador de endereços, que passa a indicar 000001B = 1. Ao receber este novo endereço a RAM apresenta seu conteúdo ao display gráfico, ou seja, ela indica o código da célula 1 (2ª célula a partir da esquerda no topo da tela). O gerador gráfico carrega no registro de deslocamento os 20 bits da 1ª fileira dos segmentos b5 e b2 desta célula. A cada $\phi 1$ eles são apresentados serialmente, correspondendo aos pixels da posição na tela agora varrida pelo feixe eletrônico.

A operação acima se repete 64 vezes, até completar a apresentação na tela da 1a. fileira de pixels das primeiras 64 células. Neste momento o divisor de frequência por 64 (emite um pulso por cada 64 que recebe) envia o sinal Ø3 para o GSC, indicando que acaba de ser varrida uma linha da tela.

O GSC tem agora duas tarefas a cumprir: (1º) emitir o sinal de sincronismo horizontal H para o misturador de vídeo, para ocorrer o retraço do feixe eletrônico (2º) resetar o contador de endereço, que recomeça a endereçar a RAM a partir de zero. Observe que todas as operações para apresentação da 1a. fileira de pixels das 64 primeiras células serão repetidas, mas agora o feixe eletrônico está posicionado na 2a. fileira. Isto se deve ao fato que o segmento de célula é composto de 5 linhas idênticas de pixels (veja figura 8-4).

Após a repetição de 5 fileiras de pixels o divisor de frequência por 5 envia o pulso Ø4 para o contador 0 a 2 do GSC que sinaliza para a ROM que a 1a. linha de segmentos terminou. Novamente o contador de 0 a 64 do GSC vai endereçando sucessivamente a RAM, em ordem crescente; a cada um deles ela apresenta à ROM o conteúdo daquele endereço, que é o código de célula; a ROM, recebe esse código como se fosse endereço de 1 entre 64 posições onde estão gravados os bits/pixels daquele tipo de célula. Agora, entretanto, o contador de 0 a 2 do GSC indica à ROM que a linha de segmentos a ser apresentada é a 2a.

A 2a. linha de segmentos é colocada na tela, seguindo o mesmo roteiro que vimos anteriormente para a 1a. linha. Quando ela acaba de ser mostrada ocorre novo pulso Ø4, sinalizando o início da 3a. linha de segmentos.

Quando o divisor de frequência por 3 recebe o 3º pulso ele emite o sinal Ø5 para o microprocessador, comunicando-lhe que a 1a. linha de 64 células já foi mostrada. O microprocessador lê na ROM-cartucho quais os códigos das novas 64 células a serem apresentadas, assume o controle do multiplexador e escreve-os na RAM de vídeo.

O processo de apresentação na tela da 1a. linha de 64 células será repetido 16 vezes, com o microprocessador atualizando a RAM de vídeo a cada nova linha.

Na 16a. linha de células o divisor de frequência por 16 emite Ø6 para o GSC, sinalizando que um quadro completo foi posto na tela. O GSC envia o sinal de sincronismo vertical V ao misturador de vídeo, para ocorrer o retraço vertical (o feixe está na extrema direita inferior da tela e é apagado - blanking - para voltar ao topo à esquerda).

O quadro completo sinalizado por Ø6 significa que a tela está totalmente ocupada pelas $64 \times 16 = 1024$ células, formando uma imagem no vídeo. Se esse mesmo quadro continuar sendo repetido, sem alterar as posições ou tipos de células, teremos uma cena estática. É assim, por exemplo, ao ligar o vídeo-game e colocar o cartucho, antes de apertar o botão "início".

O leitor deve observar que várias funções vistas na figura 8-6 são implementadas num único CI. Num caso extremo, o ATARI usa o CI TIA - 6592 que engloba todas as funções da figura 8-6, exceto o microprocessador, o misturador de vídeo e o modulador de RF. É comum o uso do "display gráfico", como no ODISSEY, que contém a ROM-gerador gráfico, o GSC e o registro de deslocamento.

6- PROGRAMAÇÃO DE CENAS

Até aqui temos uma cena estática. Como movimentá-la? Obviamente alterando as posições ou tipos de células. Esta é a função do programa gravado (em código de máquina) na ROM - cartucho. Obedecendo as regras de programação do microprocessador (a lista de instruções que ele segue), e as limitações do estágio de geração da imagem (número de células por coluna e linha, tipos de células) só resta a imaginação e habilidade do programador.

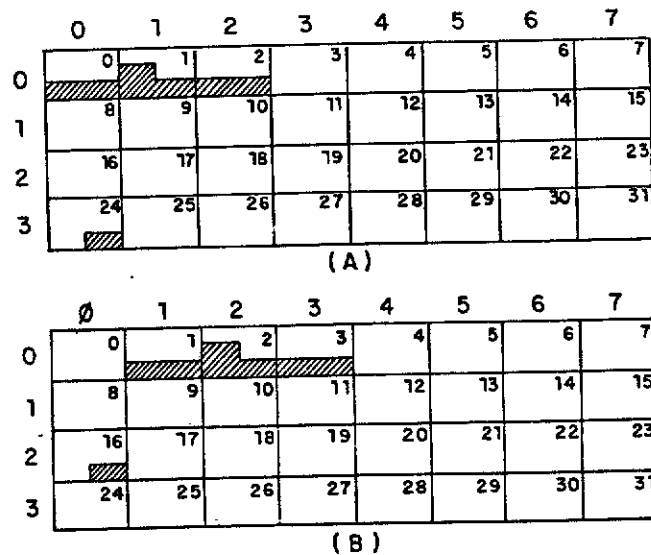


figura 8-8 - Bombardeio de navio (a) quadro inicial
(b) navio e bala deslocados

Imaginemos um jogo extremamente simplificado, um navio sendo bombardeado por uma bala de canhão (figura 8-8). Ainda para facilidade de explicação suponhamos a tela formada por 8 colunas x 4 linhas de células, e estas são iguais àquelas da figura 8-2 que foram tomadas como exemplo para

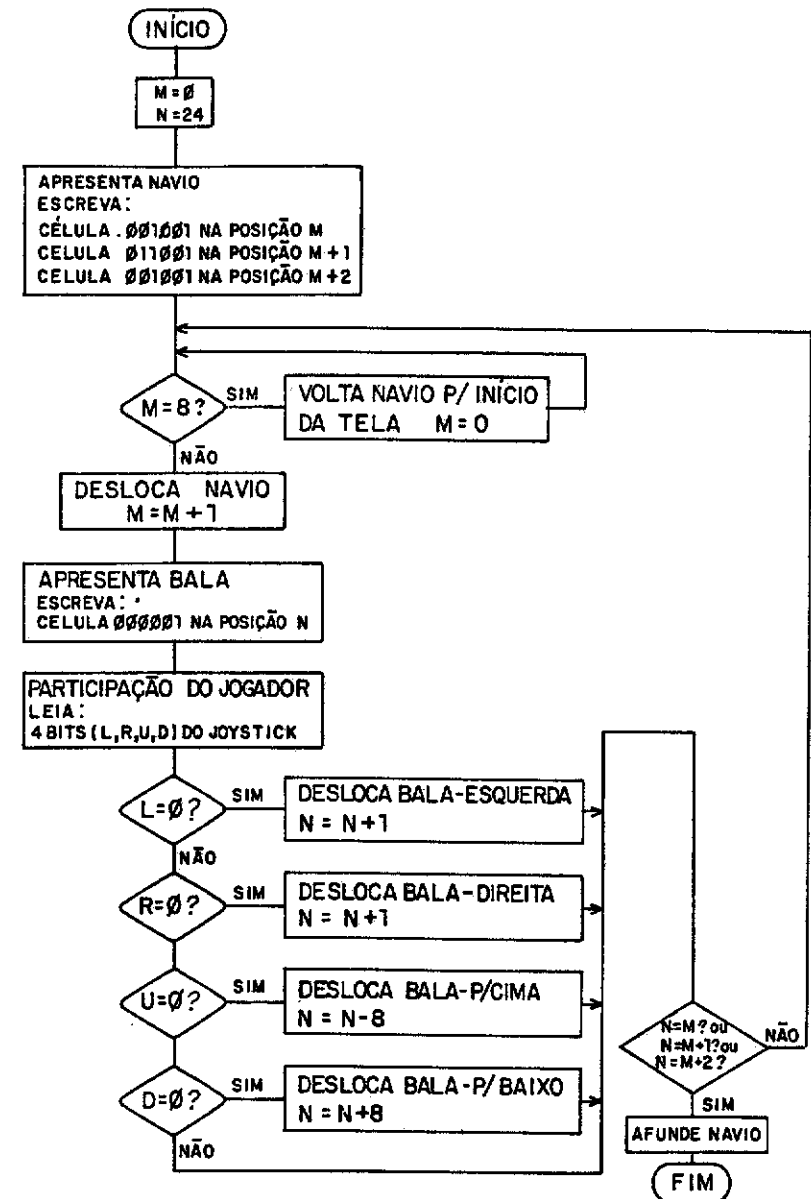


figura 8-9 - Fluxograma do jogo "bombardeio de navio" da figura 8-8.

ilustração da seção anterior.

Na figura 8-9 temos o fluxograma (simplificado e desconsiderando uma série de detalhes) do nosso jogo. Após completá-lo o programador deveria transformar cada bloco numa sequência de instruções em linguagem assembler, posteriormente traduzir para o código de máquina do microprocessador e gravar o programa na ROM do cartucho.

No jogo o navio é colocado inicialmente no topo à esquerda da tela, e a bala de canhão em baixo também à esquerda. Por ordem do programa o navio sempre se move uma posição até atingir a margem direita, quando então é repostado no início da margem esquerda. A bala movimenta-se para uma das 4 posições indicadas pelo joystick ou permanece parada se ele não for acionado. Se a bala atingir o navio ele afundará e o jogo se encerra.

No fluxograma começamos atribuindo valores iniciais para as variáveis M e N, que indicarão as posições do navio e da bala, respectivamente. No programa de cartucho isso equivaleria a carregar dois registros do microprocessador com os valores:

<u>LABEL</u>	<u>MNE</u>	<u>OP</u>	<u>COMENTÁRIOS</u>
INÍCIO	LD	M,0	;carrega registro M com 0
	LD	N,24	;carrega registro N com 24

O bloco seguinte - apresentação do navio - consistiria numa rotina para escrever na RAM (no momento adequado) os códigos das três células. O registro M contém a posição das células do navio e é usado como ponteiro para as posições na RAM:

<u>LABEL</u>	<u>MNE</u>	<u>OP</u>	<u>COMENTÁRIOS</u>
NAVIO	LD	(M),001001B	;escreve a 1a. célula do ;navio no endereço RAM ;indicado pelo conteúdo ;do registro M
	LD	(M+1),011001B	;escreve a 2a. célula do ;navio no endereço RAM ;indicado pelo conteúdo+ ;1 do registro M
	LD	(M+2),001001B	;escreve a 3a. célula do ;navio no endereço RAM ;indicado pelo conteúdo ;+ 2 do registro M

É necessário agora deslocar o navio uma posição para a direita, o que é obtido pelo simples incremento do registro M. Ou seja, somando + 1 a M todas as células do navio serão apresentadas uma posição mais à direita, dando a impressão de deslocamento. Entretanto, se o navio estiver na extrema direita da tela (M=8) ele deve retornar à margem esquerda:

<u>LABEL</u>	<u>MNE</u>	<u>OP</u>	<u>COMENTÁRIOS</u>
QUAD	CP	M,8	;compara 8 com conteúdo ;do registro M, seta flag ;Z se M = 8
	JNZ	DSL N	;salta próxima instrução ;RET N se flag Z não está ;setada
RET N	SUB	M,8	;subtrai 8 do conteúdo ;de M, retornando navio ;à esquerda
DSL N	INC	M	;incrementa M, deslocando ;do navio para direita

A apresentação da bala segue o mesmo esquema do que vimos para o navio:

<u>LABEL</u>	<u>MNE</u>	<u>OP</u>	<u>COMENTÁRIOS</u>
BALA	LD	(N),000001B	;escreve código da ;célula "bala" no en ;dereço RAM indicado ;pelo conteúdo do re ;gistro N

O quadro da participação do jogador (que determinará o deslocamento da bala conforme a posição do joystick) é uma simples entrada de dados pela porta ocupada pelo joystick. Lembre-se ainda que se o circuito adotar o mapeamento de memória (portas em posição de memória) a operação se converterá numa leitura de memória. Suporemos que o joystick seja porta nº 1:

<u>LABEL</u>	<u>MNE</u>	<u>OP</u>	<u>COMENTÁRIOS</u>
	IN	1	;lê pela porta 1 os ;4 bits do joystick ;e guarda-os no acu- ;mulador.

Se algum dos 4 bits do joystick estiver em nível 0 a bala deve ser deslocada para a direção correspondente. Observe na figura 8-8 que desloca a bala para a esquerda (L) ou direita (R) significa subtrair ou somar uma unidade a N, respectivamente. Da mesma forma, uma posição para cima (U) é dada por N-8 e para baixo por N+8. Os 4 bits são carregados na ordem no acumulador, tal que o conteúdo será 0000LRUD. L será o bit nº 3, R bit nº 2, U bit nº 1 e D bit nº 0.

<u>LABEL</u>	<u>MNE</u>	<u>OP</u>	<u>COMENTÁRIOS</u>
ESQU	TST	BIT 3	;se bit 3 do acumulador é ;0 a flag Z é setada
	JNZ	DIRE	;salta para instrução DIRE ;se flag Z não está setada
	DEC	N	;subtrai 1 de N
DIRE	TST	BIT 2	;se bit 2 do acumulador é 0 ;a flag Z é setada
	JNZ	CIMA	;salta para instrução CIMA ;se flag Z não está setada
	INC	N	;soma 1 a N
CIMA	TST	BIT 1	;se bit 1 do acumulador é 0 ;a flag Z é setada
	JNZ	BAIX	;salta para instrução BAIX ;se flag Z não está setada
	SUB	N,8	;subtrai 8 de N
BAIX	TST	BIT 0	;se bit 0 do acumulador é ;0 a flag Z é setada
	JNZ	PRÓX	;salta para instrução PRÓX ;se flag Z não está setada
	ADD	N,8	;soma 8 a N
PRÓX	...		;.....

Navio e bala já foram deslocados. Resta agora saber se o tiro acertou o alvo. Isso ocorrerá quando a posição da célula-bala for igual a uma das 3 posições ocupadas pelas células do navio, ou seja, $N = M$ ou $N = M+1$ ou $N = M+2$. Se o navio não for acertado novo quadro será posto na tela, apresentando as figuras já deslocadas.

<u>LABEL</u>	<u>MNE</u>	<u>OP</u>	<u>COMENTÁRIOS</u>
JOYS	LD	A,M	;copia conteúdo de M no acu- ;mulador
	CP	A,N	;seta flag Z se $A = N$
	JZ	ACER	;salta para instrução ACER ;se flag Z está setado

<u>LABEL</u>	<u>MNE</u>	<u>OP</u>	<u>COMENTÁRIOS</u>
INC	A		;soma 1 ao acumulador
CP	A,N		;seta flag Z se A = N
JZ	ACER		;salta para instrução ACER
			;se flag Z está setada
INC	A		;soma 1 ao acumulador
CP	A,N		;seta flag Z se A = N
JZ	ACER		;salta para instrução ACER
			;se flag Z está setada
JMP	QUAD		;salta para instrução QUAD
			;no começo do programa se
			;A ≠ N

Quando o alvo for atingido deve ser executada a rotina ACER de afundamento do navio. Uma técnica comum é "acender" e "apagar" o navio alternadamente em intervalos indicador por um contador C, enquanto o microprocessador emite para uma porta 2 de saída sinais para geração de som.

<u>LABEL</u>	<u>MNE</u>	<u>CP</u>	<u>COMENTÁRIOS</u>
ACER	LD	C,100	;carrega 100 no registro
			;contador C
	CALL	NAVIO	;chama rotina de apresentação do navio
	DEC	C	;subtrai 1 de C e seta flag
			;Z se C = 0
	JNZ	ACER	;looping - volta para ACER
			;se C ainda não é 0
SOM	LD	C,100	;carrega 100 no registro C
	OUT	2,FFH	;dá saída de byte FFH pela
			;porta C para geração de
			;som
	DEC	C	;subtrai 1 de C e seta flag
			;Z se C = 0
	JNZ	SOM	;looping - volta para SOM
			;se C ainda não é 0
	JMP	ACER	;looping - volta para ACER

No exemplo acima as figuras, embora deslocando-se na tela, são inertes. Na figura 8-10 ilustramos a técnica para dar a impressão de animação, com o respectivo fluxograma na figura 8-11.

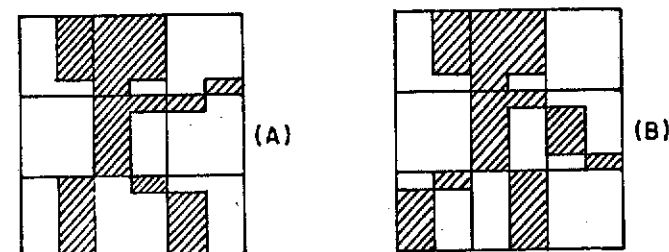


figura 8-10 - Figuras de homem diferentes simulando movimento (A) homem 1 (B) homem 2.

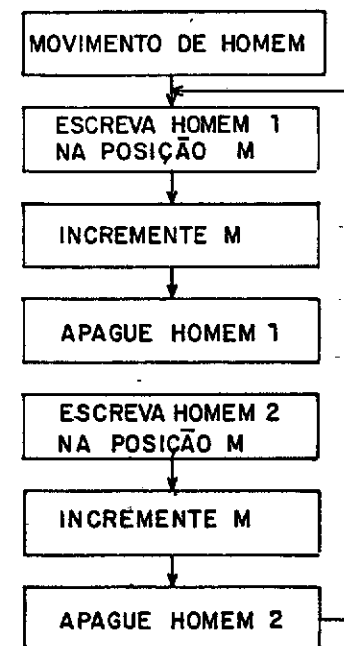


figura 8-11 - Fluxograma da cena homem caminhando.

A figura do homem é mostrada na tela. Quando ocorre o deslocamento (incremento de M), entretanto, a primeira figura é apagada e outra ligeiramente diferente ("homem 2") é apresentada. Durante todo o deslocamento as duas figuras são usadas alternadamente. No vídeo fica a impressão de uma figura animada (agitada) deslocando-se.

7 - CODIFICAÇÃO DA COR

Vimos anteriormente que a cor é codificada como defasagem de um sinal de referência - burst (figura 8-12).

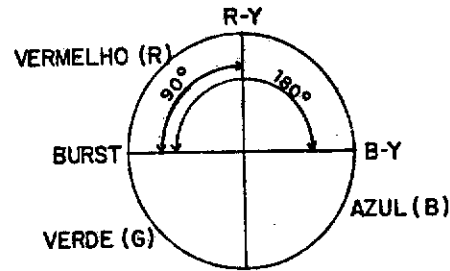


figura 8-12 - Defasagem de cores (NTSC)

Uma implementação simples consiste em fazer o sinal da portadora de crominância C 3.58MHz atravessar gates semicondutores (portas tipo buffer inversor ou não), onde ele sofrerá atraso de aproximadamente 40 nanosegundos em cada um (figura 8-13). Selecionando-se o número de gates que o burst e as cores básicas R-G-B devem atravessar obtêm-se a defasagem desejada (na verdade aproximada, existindo uma ligeira distorção das cores devido à diferença entre a defasagem ideal e a real). A tabela abaixo fornece os dados da codificação, válida para o sistema NTSC.

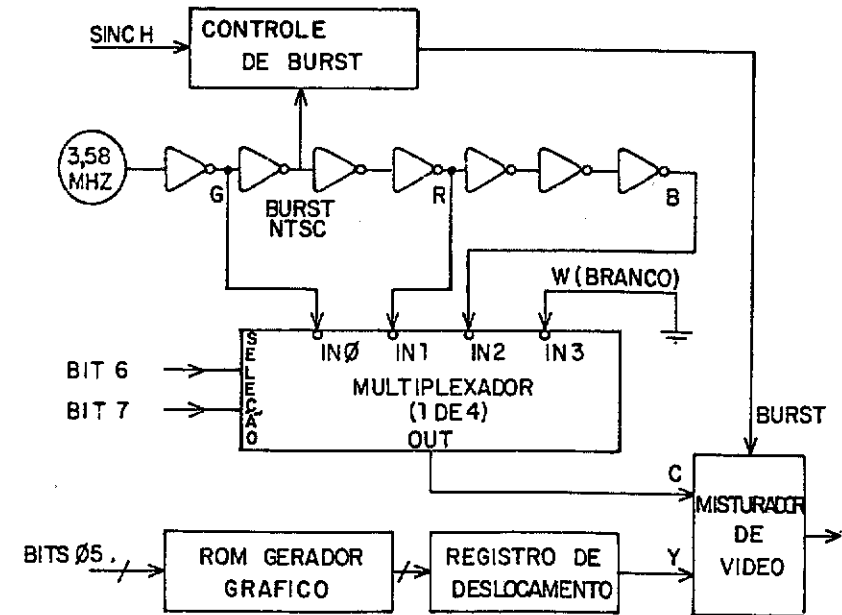


figura 8-13 - Codificação da cor no estágio de geração da imagem; C é crominância e Y luminância; sistema NTSC.

COR	DEF. BURST (°)	ATRASO IDEAL RELAT. AO BURST (NANO-SEG)	GATES (40 NANO- SEG. CADA)	ATRASO RELAT. A FONTE 3,58 MHz (NANO-SEG)	ATRASO REAL RELAT. AO BURST (NANO-SEG)
BURST(NTSC)	0	0	2	80	0
VERMELHO(R)	75	58	4	160	80
AZUL (B)	195	151	7	280	200
VERDE (G)	-45	244	1	40	-40

Voltando à figura 8-6 note que os bits b6 e b7 do conteúdo da RAM são destinados à formação de cor. Os dois formam $2^2 = 4$ alternativas que selecionam qual das 4 entradas (IN0, IN1, IN2, IN3) do multiplexador será apresentada na sua saída. Assim, para bit 6 = bit 7 = 0 é selecionada a entrada IN0, que é o sinal 3,58 MHz na fase referente ao verde (G); ele aparecerá na saída OUT, sendo misturado junto com os bits da fileira de pixels, formando parte do sinal de vídeo composto. Observe que os bits da ROM - gerador gráfico correspondem à luminância Y. Ao selecionar IN3 teremos um sinal aterrado (zero Volt), que na saída será percebido como ausência da portadora 3,58 MHz, caracterizando 'branco'.

8 - CONVERSOR PAL

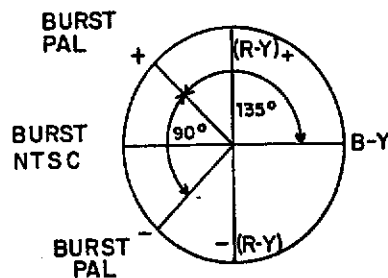


figura 8-14 - Fase do burst no sistema PAL e no NTSC

Vídeo-games são projetados para codificar a cor no sistema NTSC (usado nos EUA). Para acoplá-los ao receptor de TV brasileiro, que é baseado no sistema PAL, são necessárias as seguintes adaptações (figura 8-14):

- 1 - avançar 45° o burst NTSC, originando o burst PAL B⁻

- 2 - defasar o burst PAL em 90°, alternadamente por linhas, surgindo o burst PAL B⁺
- 3 - defasar o sinal de croma R em 180°, alternadamente por linhas.

Um circuito que realiza as alterações acima é denominado "conversor PAL", existindo vários tipos. Na figura 8-15 mostramos uma dessas implementações. O ponto de tomada do burst NTSC é cancelado, e os burst PAL B⁻ e B⁺ são gerados 1 gate antes e 1 gate após, respectivamente; o sinal R torna-se R⁺ e é criado R⁻ 4 gates. Esses 4 novos sinais dão entrada na "chave PAL" que, sincronizada pelo sincronismo horizontal seleciona qual o par B⁺, R⁺ ou B⁻, R⁻ aparecerá na saída.

Nos capítulos sobre o ATARI e o ODIS SEY veremos outras implementações de conversor PAL.

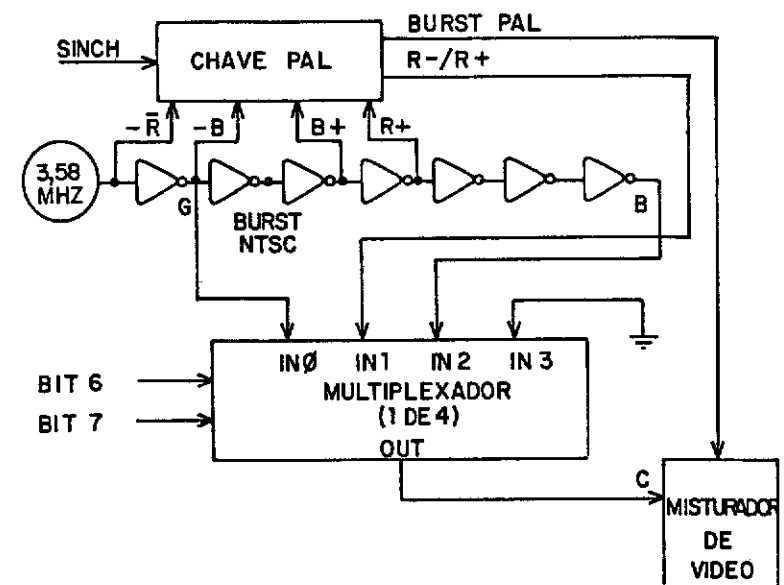


figura 8-15 - Conversão da codificação de cor NTSC para PAL.

9 - EFEITOS SONOROS

Para obtenção de efeitos sonoros em vídeo-games usa-se uma tabela na memória ou chaveamento de osciladores. A segunda alternativa é preferida por ser mais econômica e não "amarrar" o microprocessador.

Na ROM é gravada uma tabela de bytes que, devidamente manipulada pelo microprocessador é enviada a um conversor digital - para-análogo (D/A) onde transforma-se em onda senoidal (figura 8-16). A tabela é um simples conjunto de bytes que, quando convertidos para valores analógicos corresponderão a pontos formando a senóide (figura 8-17). Para que o som gerado adquira performance aceitável é necessário que haja muitos bytes na tabela, desperdiçando espaços de memória, ou que o microprocessador opere sobre uns poucos bytes transformando-os em uma sequência numerosa, neste caso gastando precioso tempo de processamento.

Uma maneira mais operacional de gerar sons é manter dois osciladores (100HZ e 300HZ);

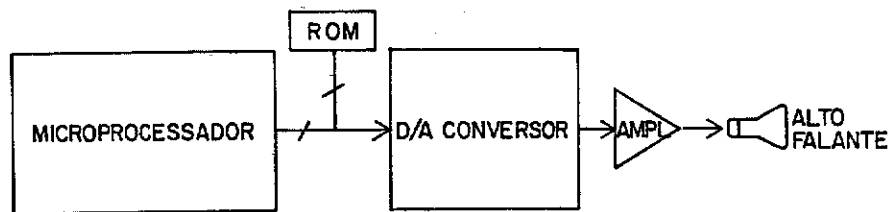


figura 8-16 - Geração de som por tabela na ROM.

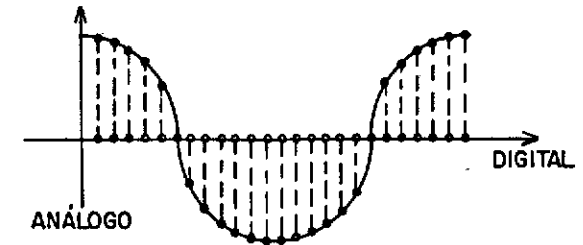


figura 8-17 - Valores digitais convertidos em senóide (análoga).

o microprocessador seleciona através do multiplexador qual deles (ou nenhum dos dois) será amplificado (figura 8-18). Na figura 8-19 temos um aperfeiçoamento do esquema anterior. Aqui o microprocessador apenas gatilha um dos one-shot (flip-flop monoestável, permanece em nível lógico ativo após o gatilhamento e volta a nível inativo depois de certo tempo). Durante 20 milissegundos, aproximadamente, o oscilador gatilhado produz ondas senóidais para o amplificador de áudio.

Os vídeo-games geralmente adotam a alternativa de controle por monoestável, embutindo todo o circuito de áudio junto com o estágio de geração de imagem (no ATARI a TIA inclui todos os circuitos de vídeo e som; no ODISSEY o som está no display gráfico junto com a ROM-gerador gráfico e o gerador de sinc e controles).

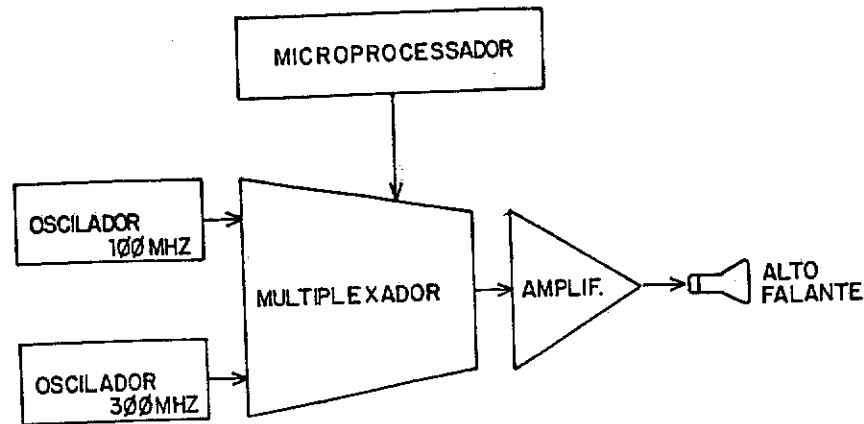


figura 8-18 - Geração de som por habilitação de oscilador.

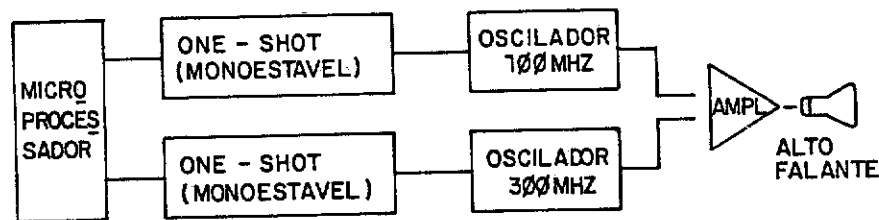


figura 8-19 - Geração de som por gatilhamento de oscilador.

CAPÍTULO 9 – ATARI

O ATARI da POLYVOX (grupo GRADIENTE) segue uma concepção de poucos CIs altamente especializados, formando um circuito compacto. O que foi dito aqui para o ATARI é totalmente válido para o DACTARI, versão nacional (cópia) daquela marca norte-americana.

Veremos o ATARI modelo 2.600, que é o de maior aceitação no mercado. Outros modelos têm produção limitada.

1 - ESQUEMA GERAL

O microprocessador 6507 (figura 9-1) controla todas as operações, seguindo o programa gravado na ROM 2716 do cartucho. Dados iniciais para início de jogo (número de participantes, colorido ou branco e preto, etc) formam a porta A para a interface adaptadora de periférico PIA 6532. A outra porta, B, é atribuída a 2 joysticks. As informações processadas são transformadas em sinal de vídeo composto pela interface adaptadora de televisão TIA 6592, que recebe ainda a saída do conversor para adaptação do NTSC ao PAL. O clock é acoplado à TIA, chegando ao microprocessador através dela.

Os CIs citados acima, da ROCKWELL, estão sendo substituídos pelos seguintes, da SIGNETICS (PHILIPS):

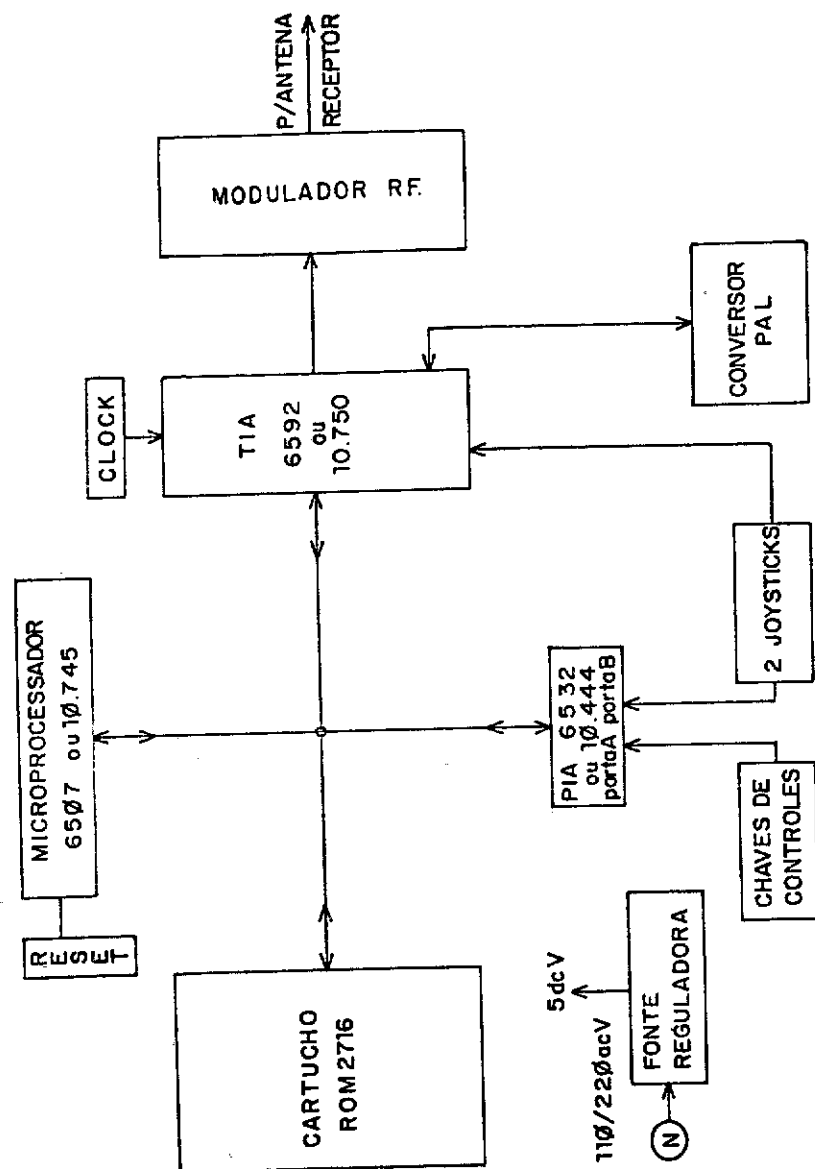


figura 9-1 - Estágios do ATARI.

	ANTIGO (ROCKWELL)	NOVO (SIGNETICS)
--	-------------------	------------------

MICROPROCESSADOR	6.507	10.745
TIA	6.592	10.750
PIA	6.532	10.444

Os dois grupos são totalmente compatíveis e equivalentes, podendo ser feita a substituição total (os três) ou parcial.

Um esboço das posições ocupadas na placa principal pelos principais componentes é apresentado na figura 9-2. Há conectores separados para os dois joysticks, bem como chaves para seleção do nível de dificuldade do jogo (A - experientes, B - novatos). O adaptador de tensão é encaixado no tipo plug-in. Uma blindagem metálica protege o modulador RF e outra os CIs 6507, 6592 e 6532. Esta última tem a fenda onde é encaixado o cartucho, além de material protetor colado aos CIs, sobre os quais é colocada a placa do conversor PAL. Para serviços no console basta distorcer os pinos da blindagem e forçá-la para cima; o material colado aos CIs pode ser retirado com uma faca, sem maiores problemas.

Quando a tampa superior do console é encaixada as chaves da figura 9-2 ficam em ressaltado para o usuário. Na placa fica também uma das partes macho/fêmea dos conectores de joystick e adaptador de 9 Volts. A fenda do conector de cartucho tem uma abertura correspondente na tampa do console.

3 - MICROPROCESSADOR 6507/10745

O microprocessador 6507 da ROCKWELL é da família do famoso 6502 da MOSTEK, usado nos microcomputadores APPLE. Na verdade é um dos 10 microprocessadores da família 65xx, todos eles sendo compatíveis em software (lista de instruções iguais).

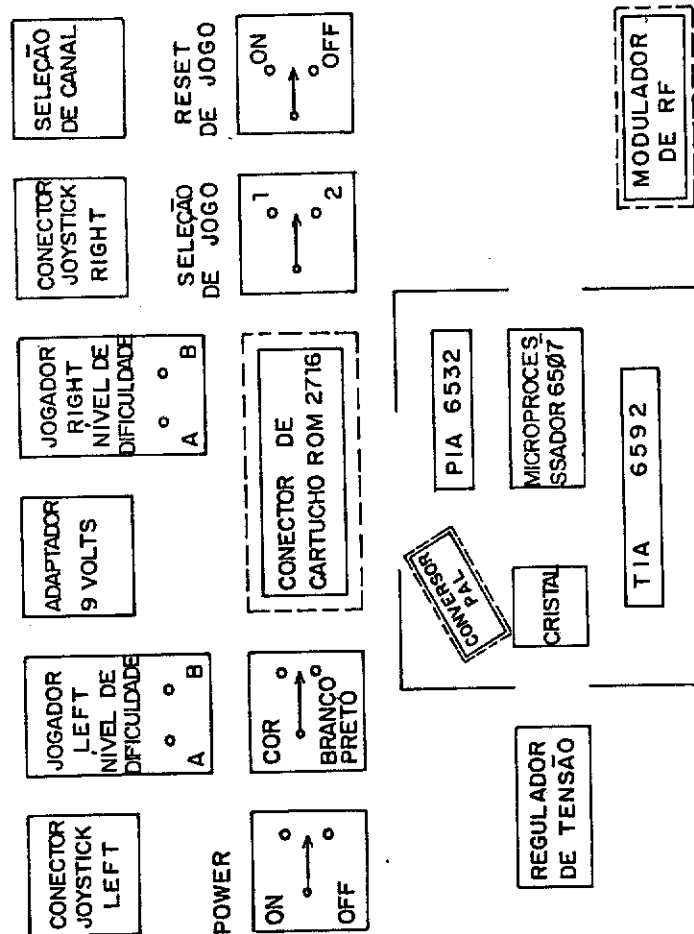


figura 9-2 - Esboço da placa principal.

o 6507 tem 28 pinos (figura 9-3), sendo 13 de endereços (A0 a A12), o que permite endereçamento de $2^{13} = 8$ Kbytes. Inclui-se também na categoria de 8 bits de dados (D0 a D7).

Possui circuito interno de clock, dispensando cristal em operações que não exijam temporização extremamente precisa. Em todo caso é preciso um sinal externo entrando em Ø1 (pino 27) para controlar a frequência do clock. É gerada uma frequência Ø2 (pino 28), como saída, que sincroniza outros dispositivos e periféricos.

$\overline{RES}$ (pino 1) reseta o programa e $R/\overline{W}$ (pino 26) indica leitura (READ) e escrita (WRITE), o primeiro ocorrendo com nível 1 e o segundo com nível 0 (lógica negativa, $\overline{W}$). RDY ("ready", pronto) é sinal de entrada; quando em nível 0 para o microprocessador no ciclo de busca de instrução na memória (fetch), apresentando nos seus pinos o endereço que estava acessando (RDY é bastante útil em consertos, permitindo sequência passo a passo).

A alimentação é 5 Volts regulados (V_{cc} pino 4), com o terra em V_{ss} (pino 2). Os sinais do 6507 são compatíveis com a família TTL.

O 6507 tem uma estrutura de registros bastante simples, que são (figura 9-4):

- A ACUMULADOR - centraliza e registra resultado de operações lógicas e aritméticas
- X REGISTRO DE ÍNDICE X - contém o valor a ser somado ao endereço em instruções com indexação; pode ser usado ainda como contador em loopings e outras tarefas auxiliares
- Y REGISTRO DE ÍNDICE Y - Semelhante a X

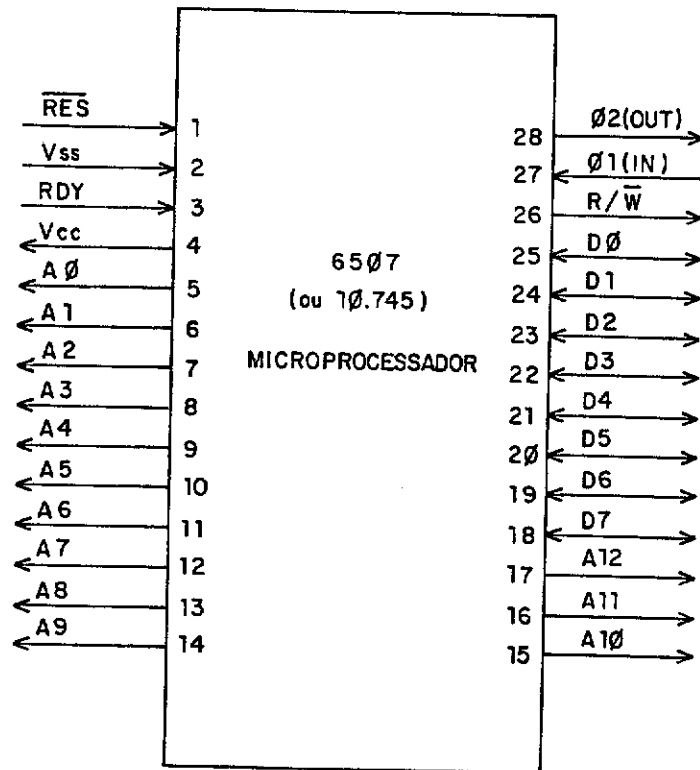


figura 9-3 - Microprocessador 6507

- SP (STACK POINTER (ponteiro de pilha) - aponta para o último endereço na pilha (RAM) onde são salvos os conteúdos de registros.
- PC PROGRAM COUNTER (contador de programa) - começa com o valor 0 e incrementa + 1 a cada byte de instrução, exceto em instruções de desvio que alteram seu conteúdo.
- P REGISTRO DE STATUS - é composto de 6 "flags" que são setados ou resetados conforme o resultado da instrução).
 - N (NEGATIVO): setada para resultado negativo, resetada para positivo
 - V (OVERFLOW): setada para overflow na operação
 - B (BREAK, "quebra") setada quando foi realizada instrução "BREAK"
 - D (DECIMAL) : setada para instrução "SED (set decimal)"; resetada para instrução "CLD (clear decimal)" que impõe operação em binário
 - Z (ZERO) : setada para operações cujo resultado seja 0 e resetada para valores diferentes de 0 (por exemplo, se compararmos o conteúdo de acumulador com outro registro o flag Z será 1 se houver igualdade - resultado zero - e será 0 se ocorrer desigualdade).
 - C (CARRY) : setada quando ocorrer carry na operação.

A lista de instruções do microprocessador 6507 - assembler e código de máquina - é apresentada no apêndice A no fim do livro.

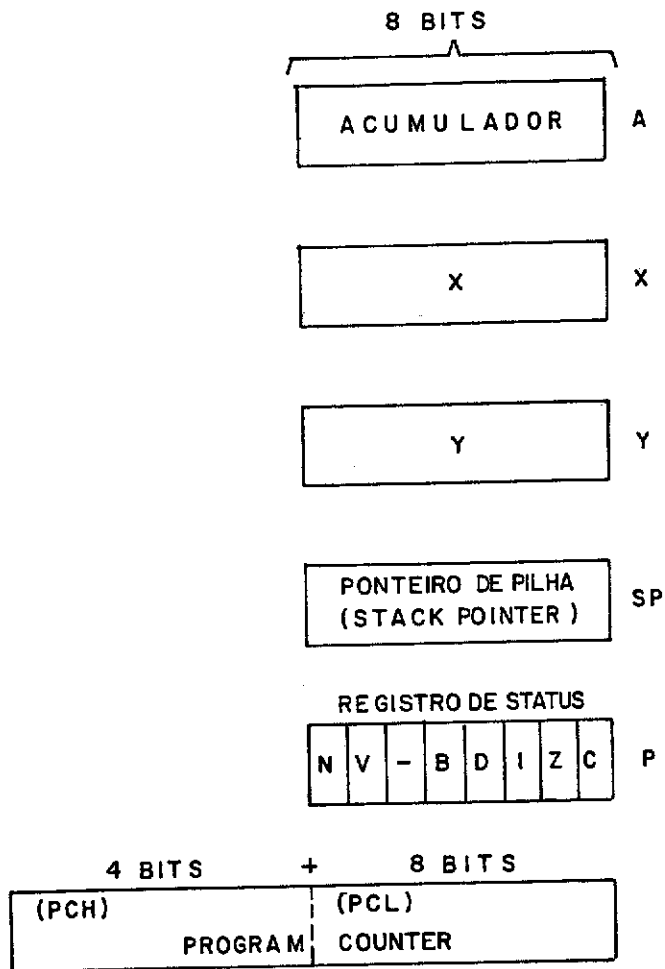


figura 9-4 - Registro do 6502.

O ATARI utiliza um circuito de reset automático comandado pelo temporizador 555 (figura 9-5). Quando a tensão de 5 Volts da fonte regulada é ligada ocorre um pulso nível alto para o pino 1 do microprocessador, resetando-o, isto é, obrigando-o a reiniciar o programa.

Após um certo tempo o 555 restabelece nível 0 no pino reset, possibilitando ao 6507 seguir as instruções em ordem crescente.

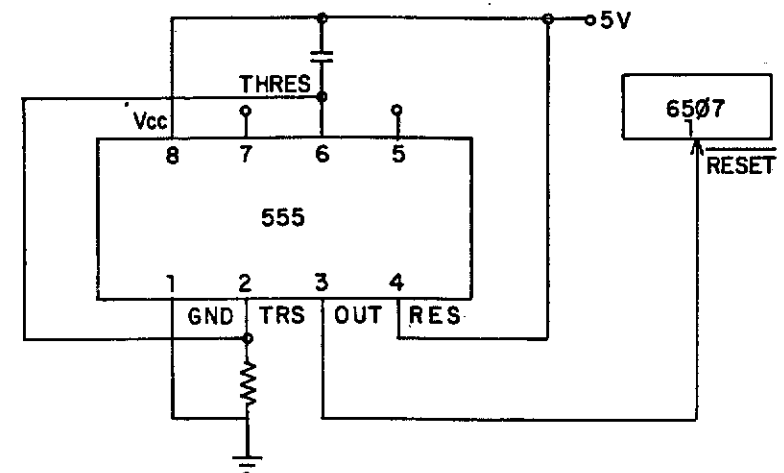


figura 9-5 - Circuito de reset.

5 - FONTE REGULADA

A fonte regulada (figura 9-6) é derivada do regulador de tensão 7805 recebendo no seu pino 1 a alimentação de 9 Volts do adaptador externo, e tendo na saída (pino 3) um filtro de 2 capacitores e um indutor.

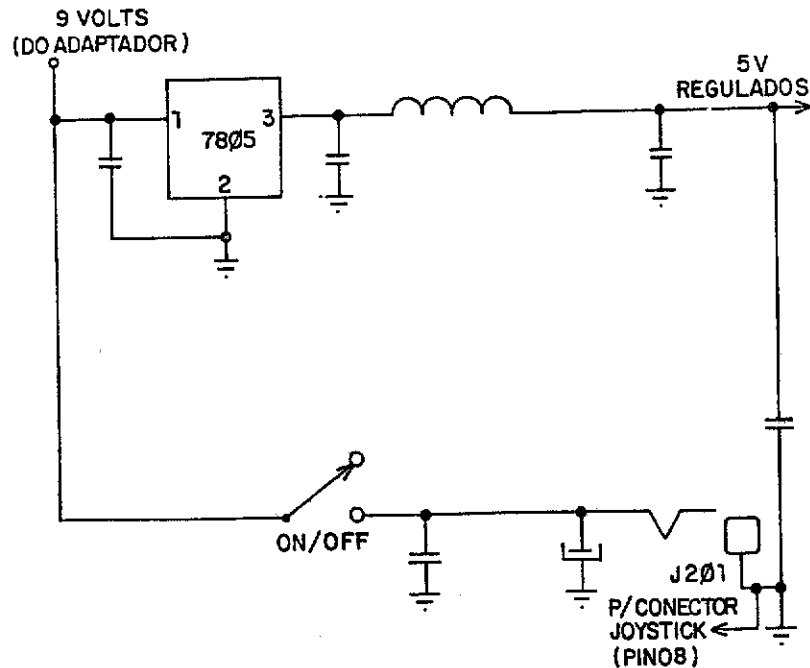


figura 9-6 - Fonte regulada

6 - CARTUCHO

O cartucho consiste no circuito suporte da ROM 2716, já visto no capítulo 4. Sua ligação no ATARI é apresentada na figura 9-7.

O byte de dados da 2716 (D0 a D7) é ligado à barra de dados do microprocessador (DB0 a DB7). Os 11 bits de endereços (A0 a A10), permitindo armazenamento de $2^{11} = 2$ Kbytes, é conectado à barra de endereços ABO a AB10. Os dois terminais de endereço restantes do microprocessador, AB11 e AB12, são ligados a $\overline{OE}$ ("output enable", habilita saída) e $\overline{CS}$ ("chip select", seleção de chip), respectivamente. Quando ambos são levados a nível 0 o byte de dados do endereço acessado fica disponível na respectiva barra. Como alimentação a 2716 utiliza apenas os 5 Volts regulados, através de um indutor.

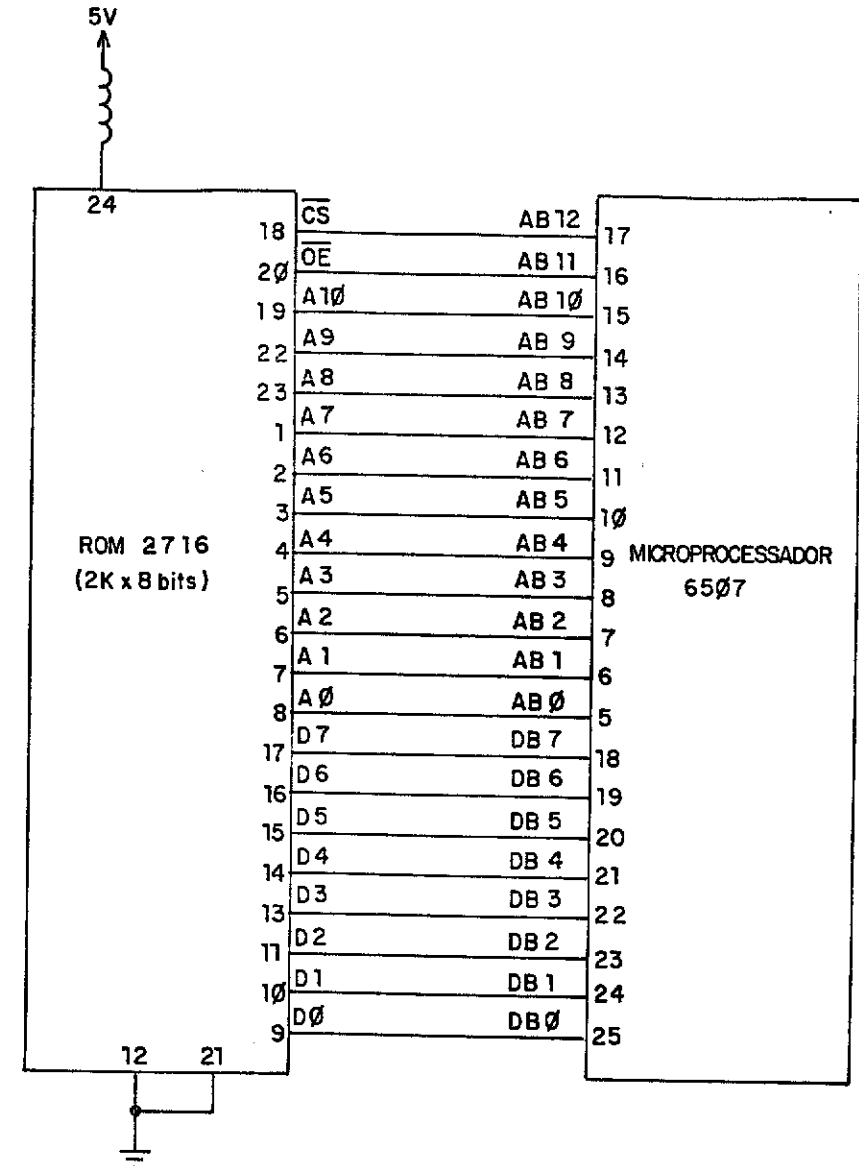


figura 9-7 - ROM - cartucho 2716.

A mesma barra de endereços saindo do microprocessador é ligada aos outros CIs. Para diferenciar o endereçamento entre eles um dos terminais é usado com exclusividade. Assim, AB11 (pino 16 do 6507) é conectado unicamente à ROM 2716, no seu terminal $\overline{OE}$ (pino 20), caracterizando endereços dirigidos a ela.

7 - PIA (INTERFACE ADAPTADORA DE PERIFÉRICOS)

A PIA 6532 faz a interface entre o microprocessador e os joysticks, além das chaves de controle. Possui 6 registros internos (tipo RAM) que armazenam os dados colocados nas suas 2 entradas (portas A e B, com 1 byte cada). Por instrução da ROM - cartucho o microprocessador endereça as linhas ABO-AB6, acessando cada um desses registros. As chaves, porta B, são selecionadas na tampa do console e lidas pelo microprocessador no início do jogo.

O microprocessador endereça a PIA através dos terminais ABO a AB6, portanto 7 linhas. AB7, AB9 e AB12 são usados para seleção, respectivamente CS (pino 38), $\overline{RS}$ (pino 36) e CS0 (pino 37). Além desses sinais a interface precisa ainda receber $R/\overline{W}$ (READ/WRITE, leitura/escrita) no pino 35. Nível 1 indicará entrada de dados por uma das portas em direção ao microprocessador; nível 0 corresponderá a saída de dados recebidos do microprocessador para as portas. Obviamente, as chaves e os joysticks só darão entrada de dados (em direção ao 6507) portanto sinal $R/\overline{W} = 1$ (leitura, input).

$\emptyset 2$ é o sinal do clock gerado no 6507 e usado para sincronizar as operações da PIA.

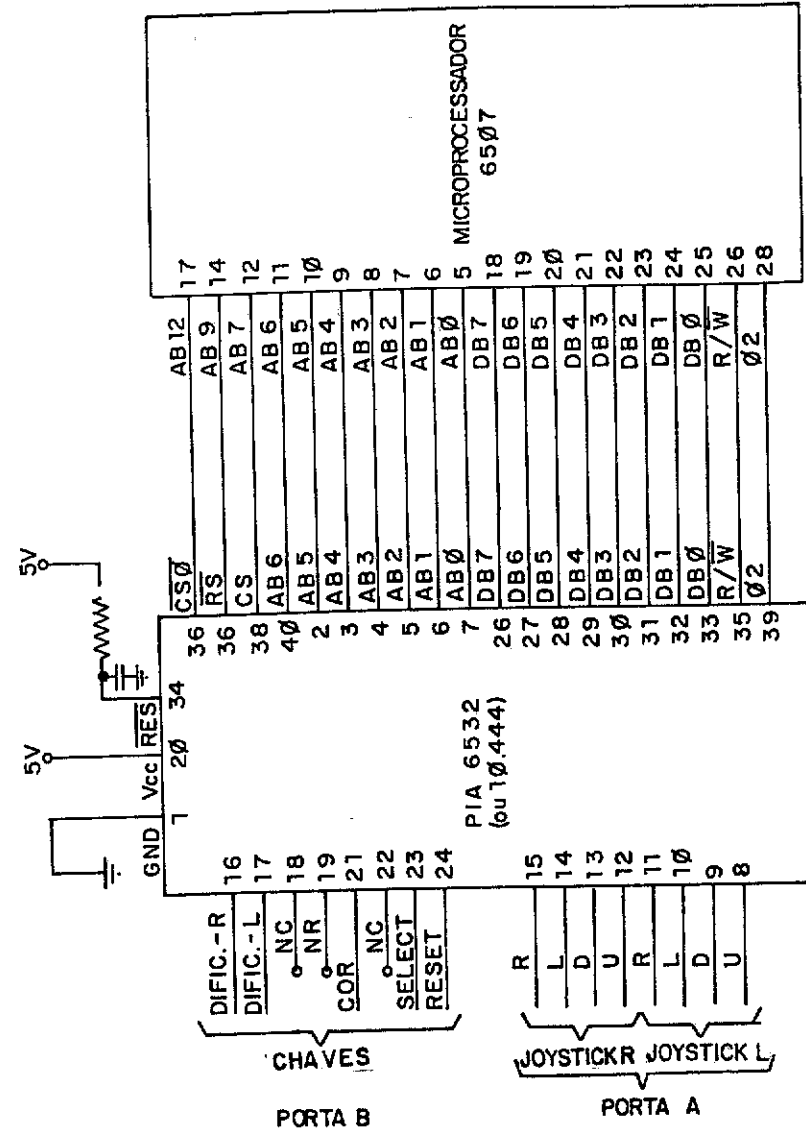


figura 9-8 - PIA 6532

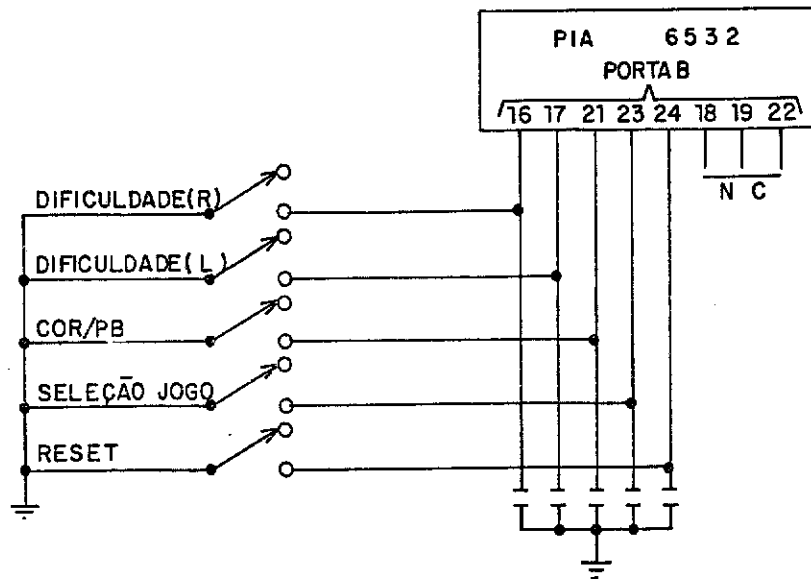


figura 9-9 - Chaves de controle.

8 - CHAVES DE CONTROLE

As chaves DIFICULDADE RIGHT, DIFICULDADE LEFT, COR/BRANCO E PRETO, SELEÇÃO DE JOGO, RESET, dão entrada na porta B, pinos 16, 17, 21, 23, 24, respectivamente. Os pinos 18, 19 e 22 desta porta não são usados, permanecendo flutuando (figura 9-9).

Os capacitores de 001 pF estão normalmente carregados, mantendo os terminais em nível 1. Quando a chave é acionada fecha-se o circuito do capacitor para o terra, descarregando-o e provocando um pulso nível 0 para o pino correspondente da PIA, que armazena esse bit de informação e quando habilitada pelo microprocessador retransmite-o.

9 - JOYSTICKS

Os dois joysticks têm cada um 4 saídas para a PIA (pinos 4,3,2,1), 3 para a TIA (pinos 5,9,6), além da alimentação de 5 Volts (pino 7) e o terra (pino 8), conforme esquema da figura 9-10.

As saídas para a PIA funcionam como a porta B anterior. Os terminais são normalmente mantidos em nível 1. Quando o circuito sensor de contactos internos do joystick (figura 5-7) coloca algum dos 4 bits em nível 0 o capacitor de 001 pF é descarregado, provocando um pulso negativo no respectivo terminal da porta A. O programa-jogo deve providenciar instruções que habilitem a porta A da PIA de tempos em tempos (geralmente 20 milissegundos), fazendo que o sistema processe os dois grupos de 4 bits, e em consequência execute alterações no andamento do jogo (mudança de posição das imagens na tela, resultados, etc).

O terminal 6 - TIRO - é uma ligação direta entre o botão de ação do joystick e a TIA.

10 - TIA (INTERFACE ADAPTADORA DE TELEVISÃO)

A TIA contém todos os estágios analisados na figura 8-1 (ROM, RAM, contador de posição, codificador de cor e controles). É selecionada pelo microprocessador através dos pinos 24 (CS0) e 21 (CS3); CS1 (pino 23) e CS2 (pino 22) ficam permanentemente habilitados, o primeiro ligado à alimentação de 5 Volts e o segundo aterrado (figura 9-11).

A interface recebe o sinal do circuito oscilador (a ser visto adiante) no pino 11, e dele gera a frequência 80 (pino 4) que é encaminhada ao

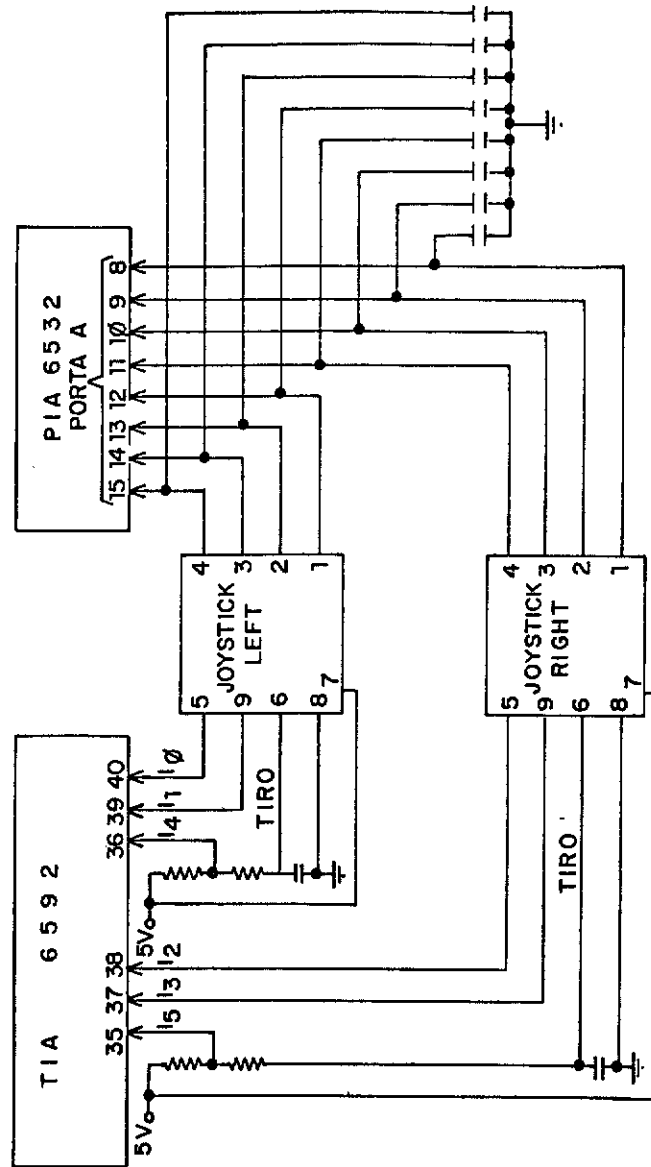


figura 9-10 - Circuito de joysticks.

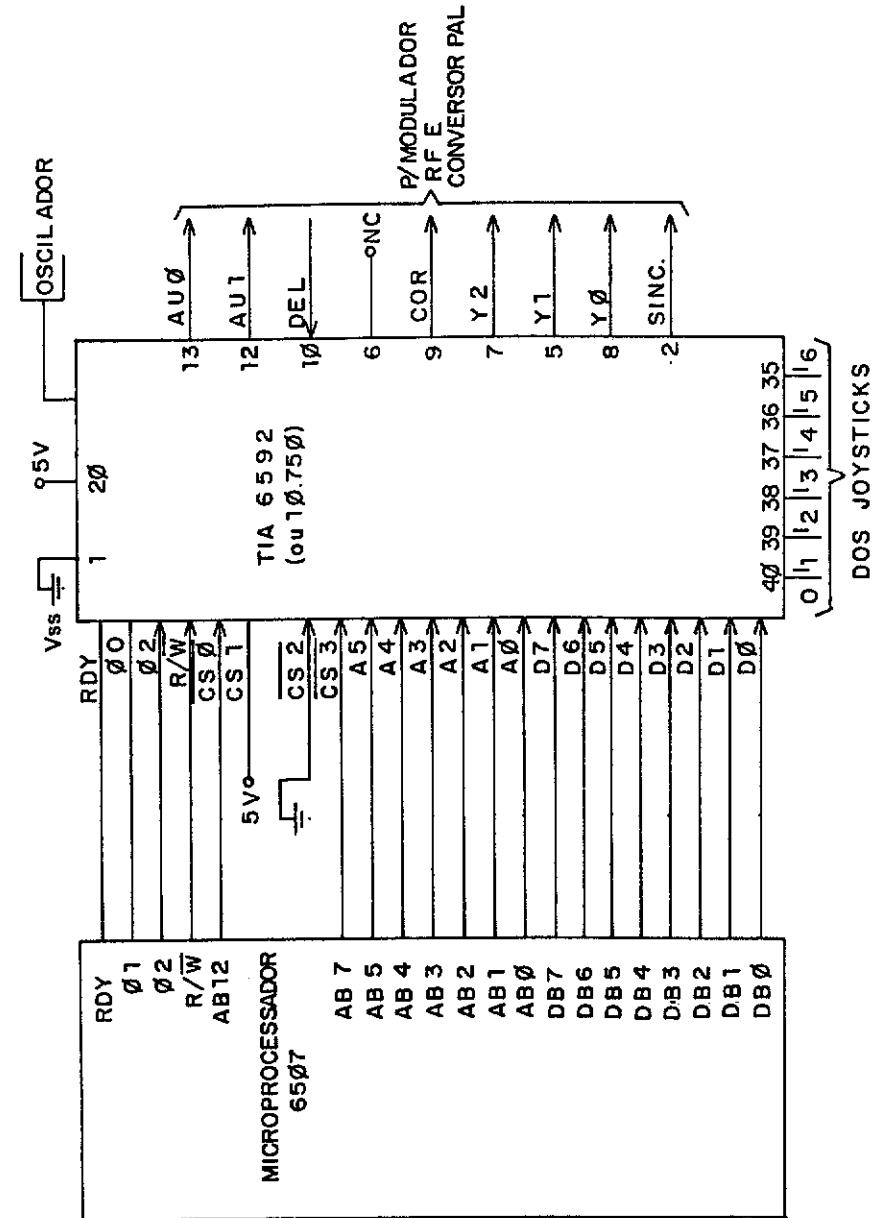


figura 9-11 - TIA 6592

microprocessador. Este a recebe (como $\phi 1$) e processa, emitindo o clock $\phi 2$ que controla toda a temporização do sistema, inclusive voltando à TIA (pino 26).

Existem ainda 6 terminais de endereço (A0 a A5) e 8 de dados (D0 a D7). Com $R/\bar{W}$ (pino 25) em nível 0 será escrito no endereço da RAM interna os bits colocados na barra de dados.

RDY (pino 3) quando levado a nível 0 para o microprocessador, mantendo-o neste estado até voltar ao nível 1.

Como vimos no capítulo 8 a TIA processa as informações recebidas e emite os sinais de luminância Y0 (pino 8), Y1 (pino 5), Y2 (pino 7), cor (pino 9), blanking (pino 6), sincronismo (pino 2), duas saídas de áudio Au0 (pino 13) e Aul (pino 12). No pino 10 encontramos uma entrada delay (retardo) para controle do tempo de atraso entre os sinais de luminância Y e crominância C, ajustável por um trimpot (figura 9-12).

11 - CIRCUITO DE CLOCK

O circuito gerador de clock é visto na figura 9-13. A frequência gerada é enviada para o pino 11 da TIA. O cristal e os 2 transistores são localizados dentro da blindagem maior da placa principal, à esquerda do microprocessador.

12 - CONVERSOR PAL

Na figura 9-14 temos o circuito do conversor PAL, lembrando que é uma placa colocada sobre os CIs 6507, 6532 e 6592, dentro da blindagem.

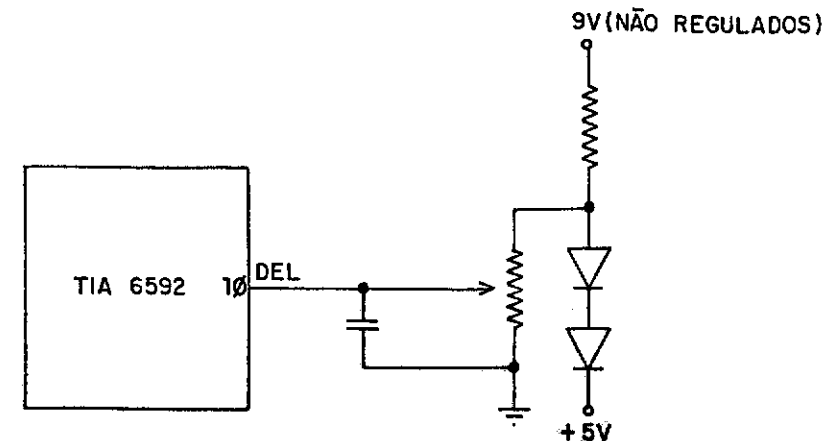


figura 9-12 - Circuito de atraso da luminância.

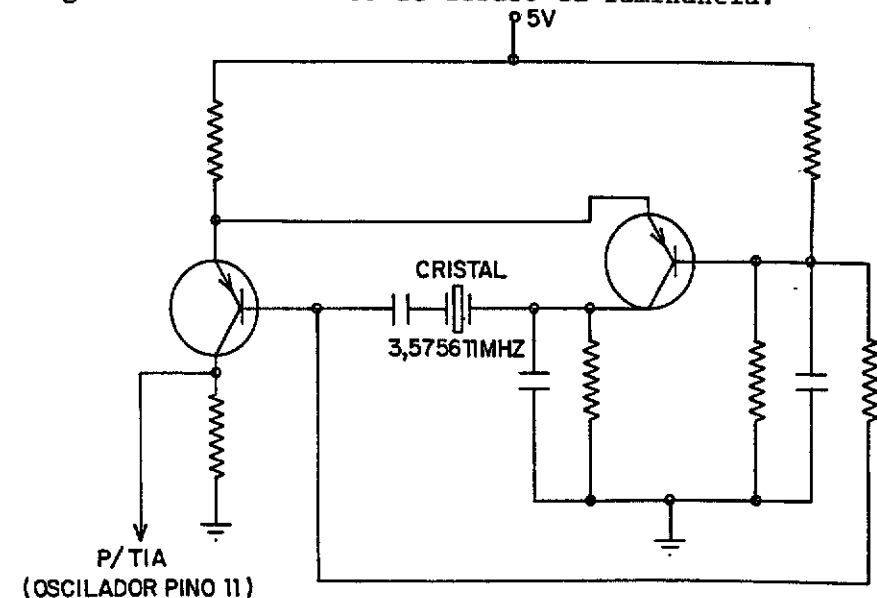


figura 9-13 - Circuito de clock.

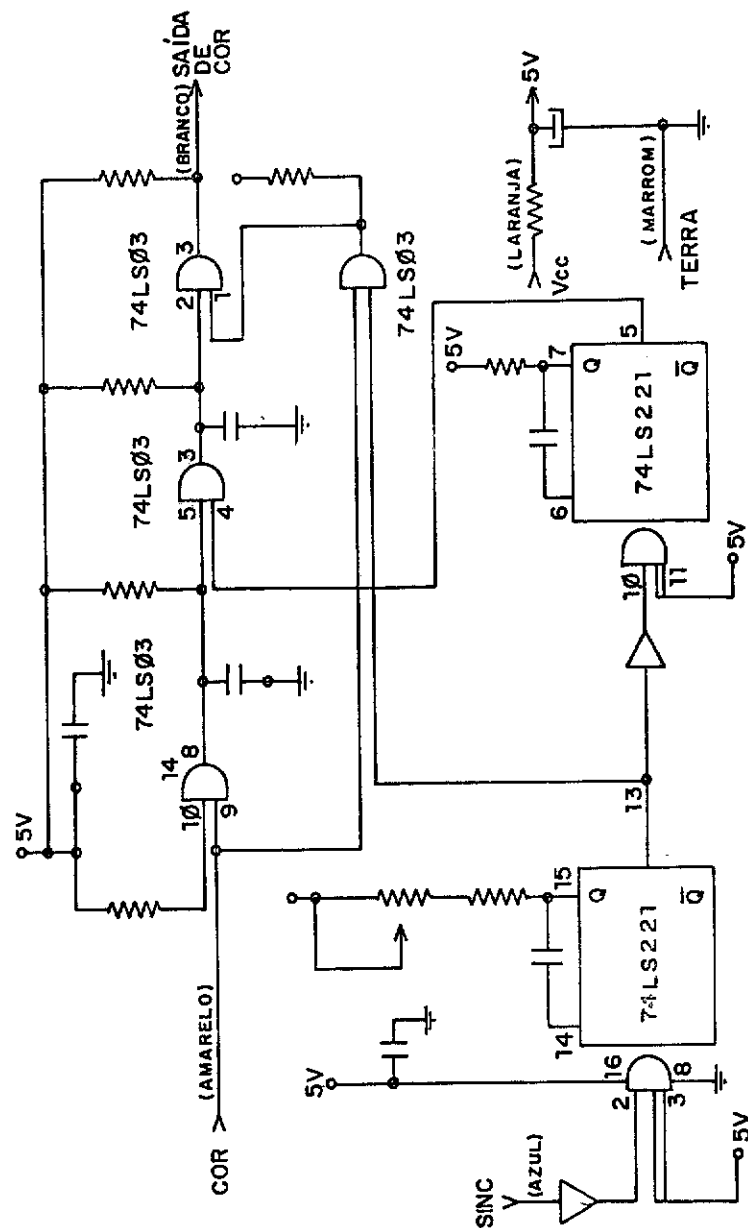


figura 9-14 - Circuito conversor PAL - as indicações entre parêntesis referem-se às cores dos fios.

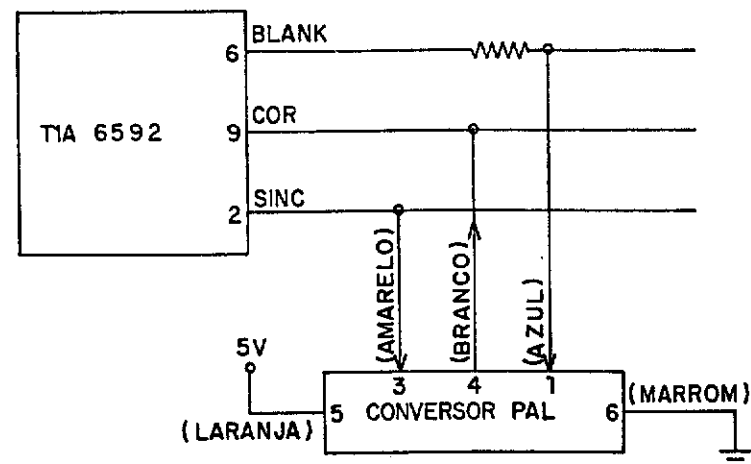


figura 9-15 - Ligação do conversor PAL à TIA - as indicações entre parêntesis referem-se às cores dos fios.

Sua conexão à TIA é apresentada na figura 9-15, com indicação da cor dos fios na versão original da fábrica.

É comum em oficinas clientes querendo fazer conversão de vídeo-game ATARI norte-americano importado (ou seja, contrabandeado) para nosso sistema PAL. Basta adquirir a placa conversora em casa comercial de eletrônica (em São Paulo há várias vendendo) ou revendedor GRADIENTE e fazer as ligações conforme indicado na figura 9-15. Em seguida coloca-se um tecido protetor acima dos CIs dentro da blindagem e sobre ele o conversor.

Outros tipos de circuitos conversores podem ser encontrados ou projetados.

13 - OSCILADOR DE ÁUDIO

No capítulo sobre televisão vimos que o som é uma transmissão FM cuja portadora é 4,5 MHz acima da frequência de vídeo. No ATARI as duas saídas de áudio são moduladas pelo circuito oscilador de 4,5 MHz apresentado na figura 9-16, sendo o resultado posteriormente somado ao vídeo no modulador de RF.

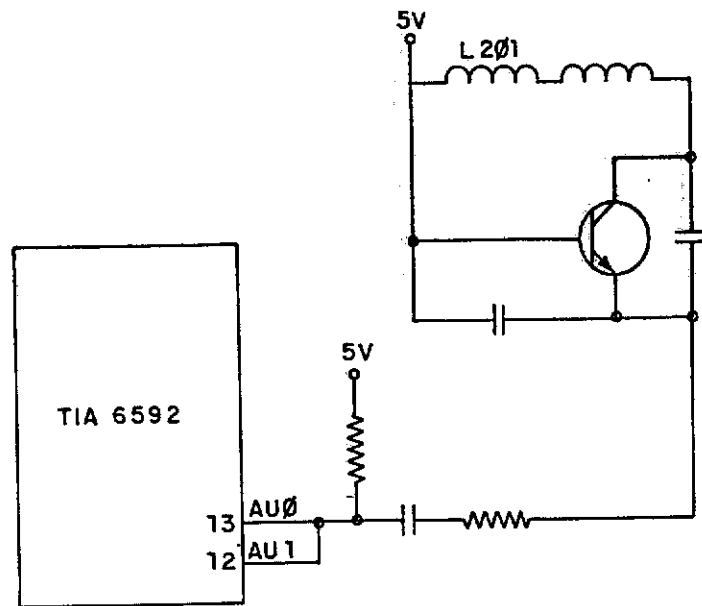


figura 9-16 - Circuito oscilador de áudio.

A frequência pode ser ajustada para 4.5 MHz no emissor do transistor Q 202 pela variação do ferrite da bobina L 201.

14 - MODULADOR DE RF

Na figura 9-17 temos o estágio final do ATARI, terminando no modulador de RF que modula

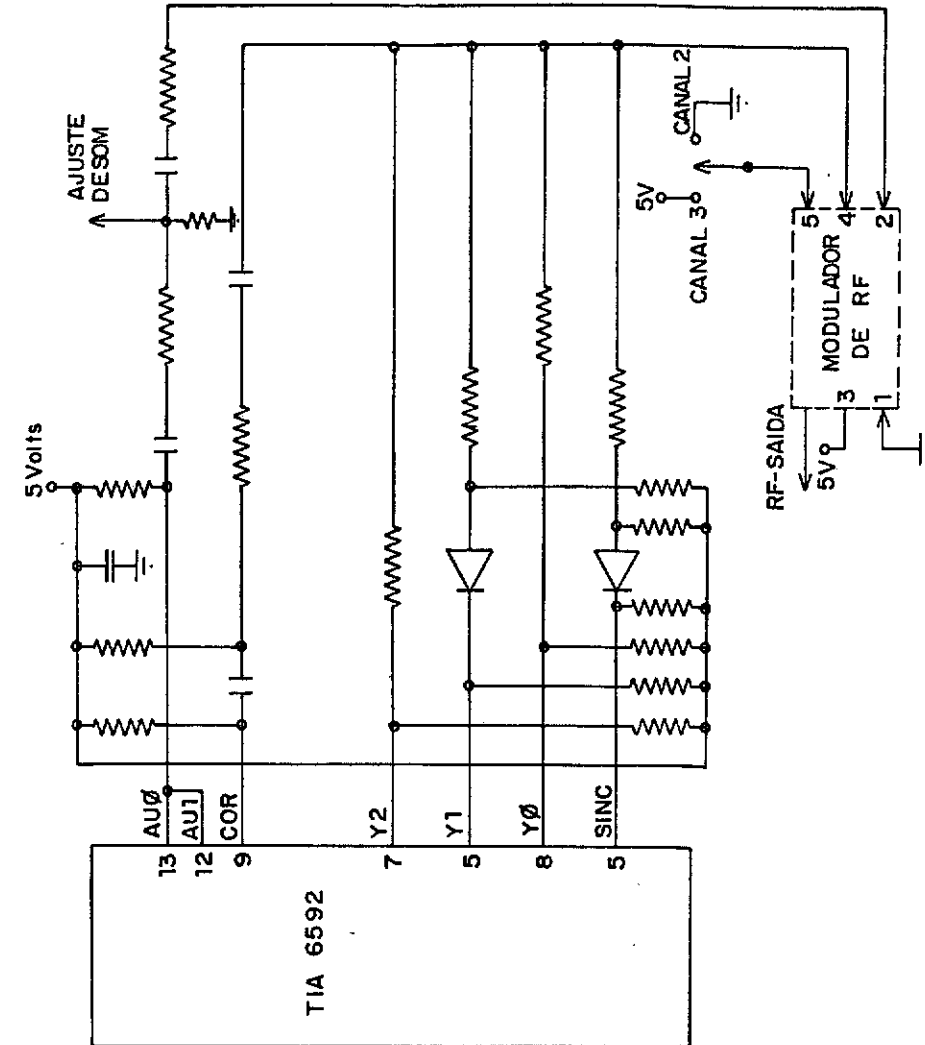


figura 9-17 - Circuito do modulador de RF.

o sinal de vídeo composto na frequência portadora do canal 2 ou 3, conforme a posição da chave ligada a seu pino 5.

O modulador de RF é mantido dentro da blindagem e tem várias características especificadas por lei (no Brasil sob a fiscalização do DENTEL). Geralmente a blindagem completa a ligação para o terra, de modo que ao retirá-la a transmissão é totalmente cortada.

CAPÍTULO 10 – ODISSEY

O ODISSEY, fabricado pela PHILIPS, é um vídeo-game relativamente complexo (se comparado com o ATARI por exemplo), aproximando-se muito da estrutura de microcomputadores. Além de teclado alfanumérico o uso de um microprocessador tipo "one-single chip" confere a esse vídeo-game características muito especiais.

1 - CONSOLE

No console do ODISSEY destaca-se o teclado alfanumérico, com entradas separadas para números, funções e letras. As teclas são acionadas pelo sistema de "soft touch", ou seja, há apenas indicação dos símbolos numa placa lisa.

O cartucho é encaixado numa fenda na parte central superior. Uma boa indicação é verificar se o encaixe foi realizado corretamente, sendo este ponto motivo de frequentes falhas.

Um par de joysticks fica disponível para dois jogadores, geralmente confrontando-se durante o jogo. Através de comandos é possível cancelar um deles, restando um só participante. A alteração nas regras do jogo para adaptá-lo ao número de jogadores é de competência do programa gravado no cartucho.

2 - ESQUEMA GERAL

A tensão interna dc é obtida de um ADAPTADOR externo, que transforma 110/220 Volts ac da rede em 9 Volts ac, passando-a a um sistema REGULADOR DE TENSÃO, interno ao console, que retifica-a e regula-a para 5 Volts dc.

O microprocessador é o 8048 da INTEL - tipo "one single chip". Na sua estrutura interna ele contém ROM, RAM e demais funções. A gravação da ROM é feita pela própria fábrica, não sendo possível alterá-la. Ele possui 40 pinos (DIP), é alimentado por 5 Volts dc e multiplexa endereços nas linhas de dados.

A RAM 6810 (MOTOROLA) é estática (SRAM) e tem capacidade de 128×8 , ou seja, 128 bytes = 128×8 bits. É alimentada por 5 Volts dc e precisa do latch para desmultiplexar dados e endereços na sua interação com o microprocessador. Faz a função de RAM de vídeo, armazenando nos 128 endereços os bits das células e de cor.

O cartucho contém uma ROM de 2 Kbytes também precisando do latch para interagir com o microprocessador.

O teclado usa o sistema de varredura, interfaceando com o microprocessador através de um CI codificador e outro decodificador. A varredura é feita por uma porta IO, que divide-se em 3 bits de saída (output, decodificação) e 4 de entrada (input, codificação). Ainda no teclado encontramos um botão ligado diretamente ao reset do microprocessador.

Os joysticks têm 5 bits de dados cada

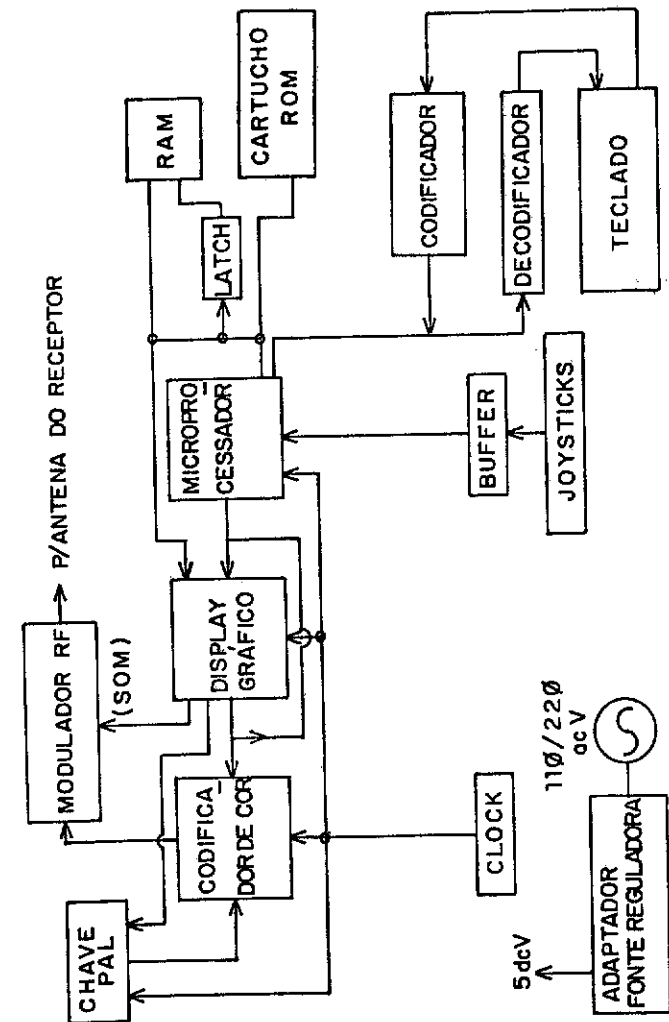


figura 10-1 - Estágios do ODISSEY.

um, sendo 4 de controle de posições e 1 de ação. A interface com o microprocessador é realizada por dois buffers.

O display gráfico 8244 contém a ROM gerador gráfico além das funções de controles e posições. Gera todos sinais indicativos de crominância, luminância, sincronismos, além do som.

O codificador de cor recebe o sinal de 3,58 MHz do clock, processando sua fase conforme as informações do display gráfico e do conversor PAL, entregando ao modulador de RF os sinais de crominância C, luminância Y e sincronismo.

O modulador de RF recebe a saída do codificador de cor, somando-a ao sinal de som recebido diretamente do display gráfico, modulando-os e formando o sinal de vídeo composto que é acoplado à antena do receptor.

3 - FONTE REGULADA

A fonte regulada recebe 9 Volts ac do adaptador (figura 10-2), retificando-os em uma ponte de 4 diodos. A chave liga/desliga do vídeo-game fica neste estágio, entre a ponte e o adaptador.

A filtragem é obtida por dois capacitores, vindo por último a regulagem pelo regulador de tensão 7805. Em operações de manutenção os pinos 2 e 3 podem ser tomados como terra e alimentação, respectivamente.

4 - CLOCK

O circuito de clock usa um cristal de 7,151222MHz (figura 10-3), cuja frequência é di-

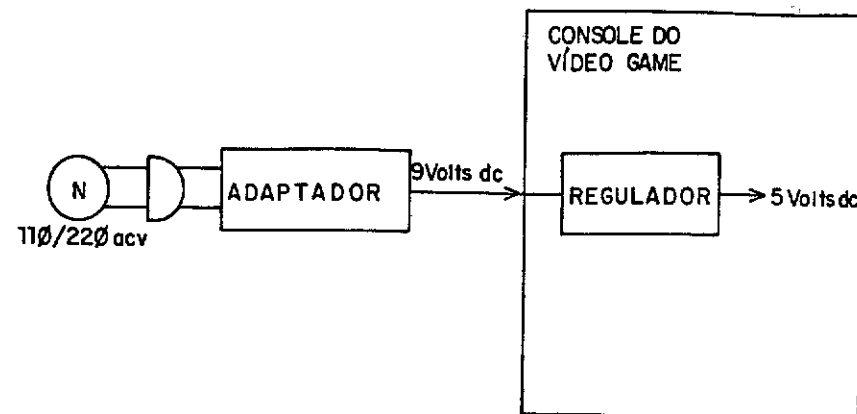


figura 10-2 - Fonte regulada.

vidida por dois no CI 74LS74, gerando sinal de 3,58 MHz no pino 9, que é enviado para o display gráfico. No pino 8 temos outra saída com a mesma frequência que, entrando no pino 3 sai no 5 como sinal de clock para o microprocessador.

Na figura 10-4 vemos o CI 74LS74, que contém 2 flip-flops tipo D. Mantendo a saída Q conectada a entrada de dados D, a cada transição do clock (pinos 3 ou 11) de negativo para positivo acarreta uma alteração no nível das saídas Q e $\bar{Q}$; como na transição positiva não há essa mudança, a frequência fica dividida por 2. A figura 10-5 ilustra esse processo.

O trimmer C006 ajusta a frequência de

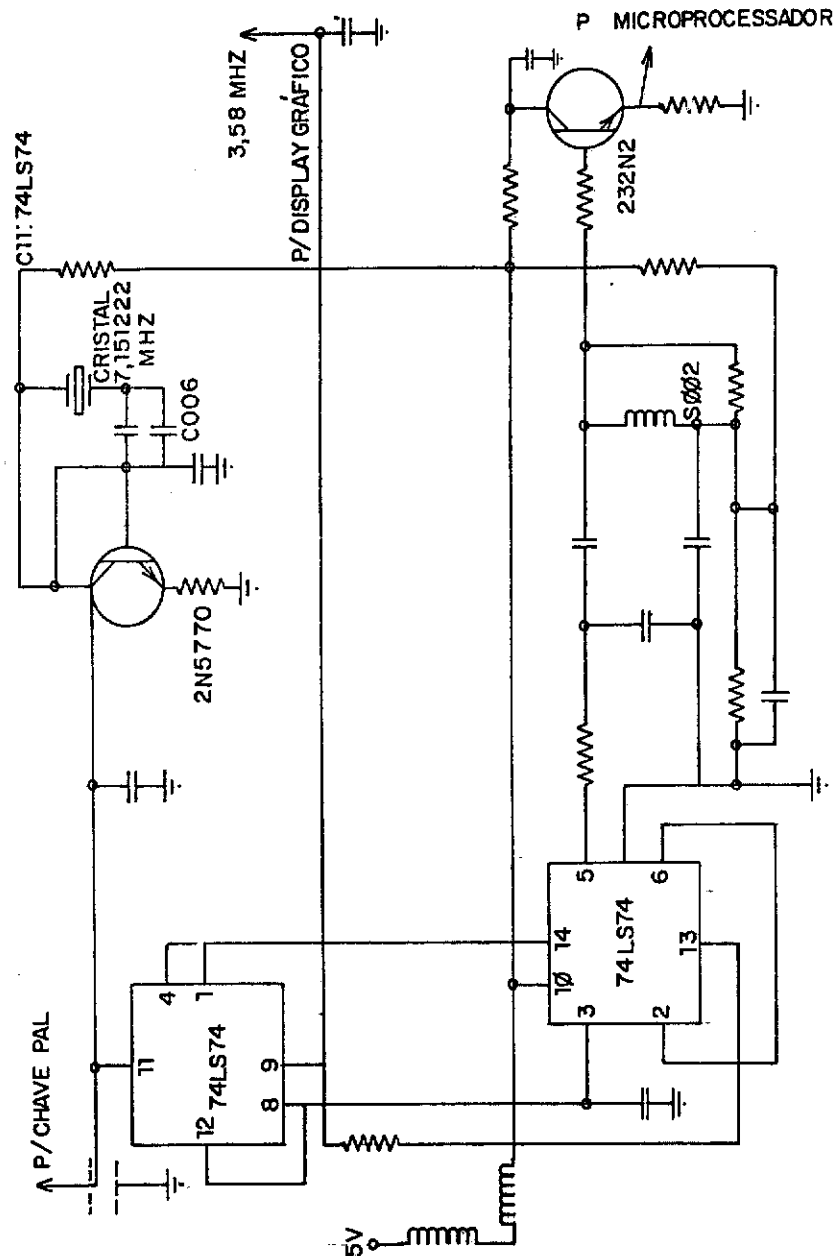


figura 10-3 - Circuito de clock.

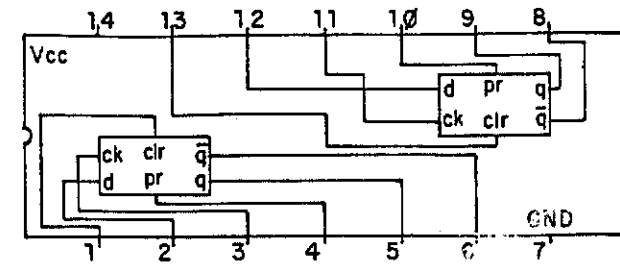


figura 10-4 - CI 74LS74.

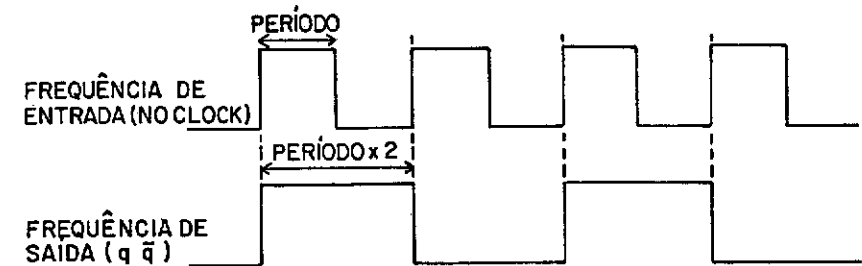


figura 10-5 - Divisão da frequência por 2.

7,151222MHZ. A bobina S002 é variável para a maior simetria e amplitude de sinal de clock para o microprocessador.

5 - MICROPROCESSADOR 8048

O 8048 - da INTEL - foi projetado em 1977 como o primeiro microprocessador a possuir ROM e RAM internas, disponíveis ao usuário. Tem exatamente a mesma estrutura do 8748, sendo totalmente compatível com ele, sendo a diferença entre os dois que o último contém uma EPROM, ao invés de ROM. O 8048 é largamente usado como controlador em instrumentos industriais.

Na arquitetura do 8048 encontramos (figura 10-6):

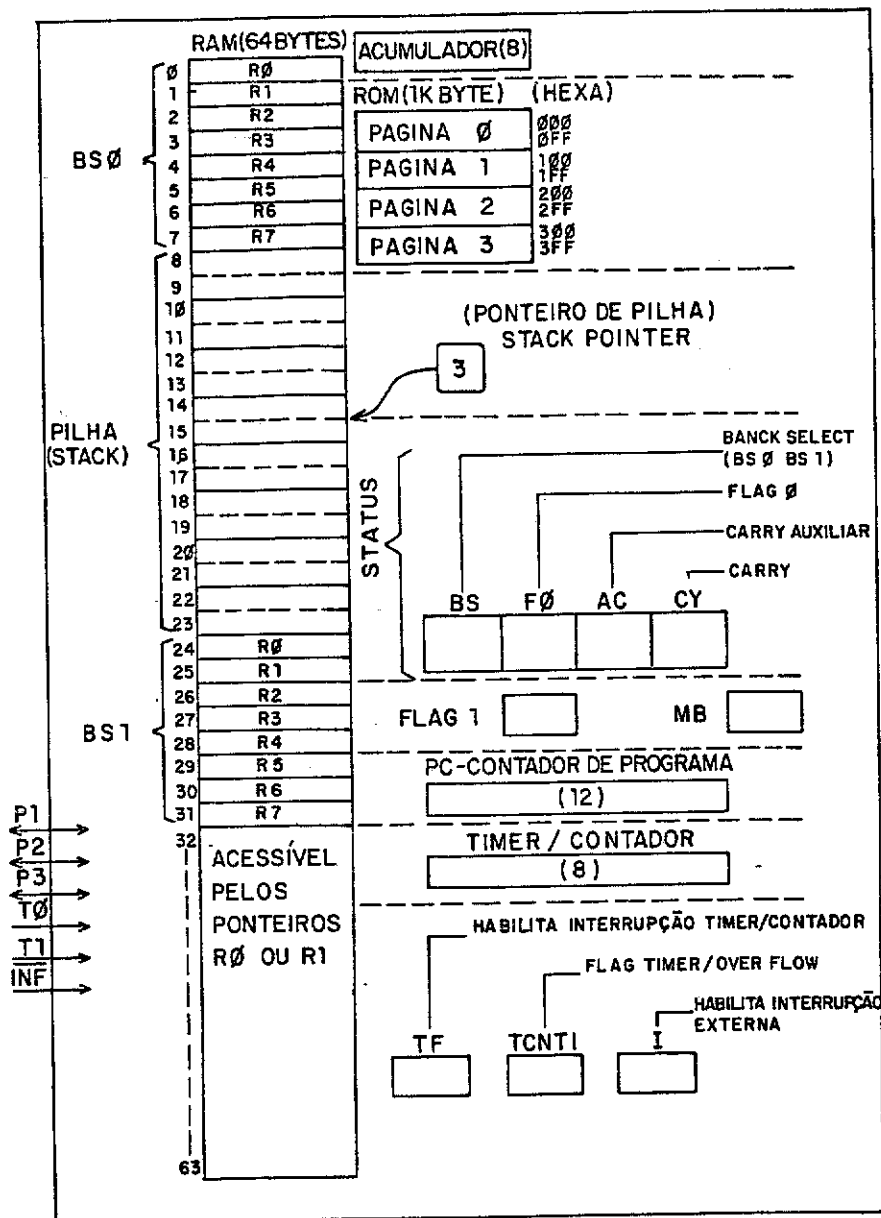


figura 10-6 - Arquitetura do 8048.

* ROM - 1 Kbyte dividida em 4 páginas de 256 bytes cada uma. Quando o microprocessador é ligado ou resetado ele busca a primeira instrução a executar no endereço 000H da página 0. Nas interrupções a instrução é buscada no endereço 003H.

* RAM - 64 bytes - dividida em 4 blocos. Os bytes de 8 a 23 são usados como pilha (stack). Os 16 bytes da pilha são tratados como 8 posições de 2 bytes cada. Quando o 8048 atende uma interrupção os 12 bits do PC mais os 4 de status são copiados numa dessas posições cujo endereço passa a ser indicado pelos 3 bits do ponteiro de pilha-SP. Terminada a interrupção uma instrução faz com que o conteúdo do endereço indicado pelo SP seja recarregado no PC e no registro de status.

Há 8 registros de uso geral (R0 a R7) cujos conteúdos e operações são controlados por instruções (exemplo: MOVR2,A, copie o conteúdo do acumulador no registro R2). Quando o bit de status BS for 0 as instruções atingirão os registros dos bytes 0 a 7; se BS for 1 elas acionarão os registros dos bytes 24 a 32. O último bloco de bytes, de 32 a 63, fica disponível para endereçamento pelos registros R0 a R7 (exemplo: ADD A, @ R4, adicione ao acumulador o conteúdo do endereço indicado pelo registrador R4).

Além dessa RAM interna de 64 bytes o 8048 pode também trabalhar com uma RAM externa de até 256 bytes, usando R0 e R1 como ponteiros. A instrução "MOVX A, @ R1-2" copia no acumulador o conteúdo do endereço externo indicado por R1 ou R2. Já "MOVX @ R1-2,A" escreve o conteúdo do acumulador no endereço externo indicado por R1 ou R2. As

sim, para usar uma RAM externa basta carregar num dos registros R1 ou R2 seu endereço e em seguida programar uma das duas instruções "MOVX".

- * PONTEIRO DE PILHA (STACK POINTER) - 3 bits - indica a posição onde foi empilhado o último registro, sendo incrementado + 1 a cada novo empilhamento e decrementado - 1 quando algum registro é retirado da pilha. Observe que cada posição na pilha ocupa 2 bytes, indo de 8 até 23 na RAM interna.
- * FLAGS DE STATUS - 4 bits - são bits setados ou resetados (1 ou 0) conforme as instruções do programa, podendo ser testados. BS é resetado pela instrução SEL RBO, que seleciona o BS0 da RAM e resetado por SEL RB1, selecionando BS1. FO pode ser resetada ou complementada pelas instruções CLR FO e CPL FO respectivamente. AC ("auxiliary carry") e ("carry") são bits decorrentes de operações aritméticas, havendo diversas instruções que os testam (exemplo: JC "endereço", salte para o endereço se o carry = 1).
- * FLAG 1 - 1 bit - segue às mesmas instruções de FO, sendo também denominado F1.
- * MB - 1 bit - resetada seleciona endereços de 0 a 2K na memória; setada seleciona endereços de 2K a 4K.
- * CONTADOR DE PROGRAMA (PC) - 12 bits - seu conteúdo indica o endereço da instrução a ser buscada para execução. É incrementado + 1 a cada nova instrução. Em instruções de salto ("jump") seu conteúdo é carregado com o endereço do salto. Tendo 12 bits permite o en-

dereçamento de $2^{12} = 4.096$ posições.

Ao ser ligado e/ou resetado o 8048 executa os primeiros 1.024 bytes de instruções gravados na ROM interna, após o que ele passa a seguir o programa gravado na ROM externa. No ODISSEY a ROM interna é gravada pela própria fábrica, contendo inicialização e rotinas do vídeo-game, enquanto a ROM externa é o "cartucho". Este é um detalhe importante, pois em caso de substituição do microprocessador deve-se trocá-lo por outro que contenha o mesmo programa monitor "ODISSEY" gravado pela PHI - LIPS. Um microprocessador 8048 que não possua tal programa não funcionará no ODISSEY, embora possa estar perfeito.

Na posição 0 da ROM interna são gravados 8 bits indicando um endereço (nas primeiras 256 posições) onde o 8048 buscará a 1ª instrução ao ser ligado e/ou resetado.

- * TIMER/CONTADOR - 8 bits - quando usado como timer (temporizador) é incrementado + 1 a cada pulso do clock até chegar a 255, quando então seta o flag TF - 1 bit (flag de overflow do timer) e este provoca uma interrupção no programa, fazendo o microprocessador executar a instrução gravada no endereço 07 da ROM interna se o flag TCNTI - 1 bit (habilitação de interrupção do timer/contador) estiver setado pela instrução EN TCNTI. A contagem começa do valor carregado no TIMER/CONTADOR pela instrução MOV T, A, onde A é o conteúdo do acumulador (exemplo: com o conteúdo do acumulador = 150, o TIMER/CONTADOR após ser carregado e acionado contará 1 bit a partir de 150 até 255 a cada clock). Como contador (e é assim que ele é usado no ODISSEY) as operações são seme-

lhantes ao visto para timer, porém a contagem é feita a cada sinal de interrupção captado no pino 39-T1. No ODISSEY T1 é ligado ao sinal de blanking (retraço durante mudança de linha na tela), de maneira que o TIMER/CONTADOR marca com precisão a linha que está sendo varrida pelo feixe eletrônico. As instruções STRT T e STRT CNT inicializam esse registro como timer ou contador, respectivamente.

* I - 1 bit - é um flag que habilita interrupções externas no pino 6 - INT - quando está setado (nível 1) ou desabilita-as (o microprocessador não as considera) quando resetado. As instruções EN I (habilita interrupção) e DIS I (desabilita interrupção) colocam o flag no estado desejado.

* ACUMULADOR - 8 bits - registro especial, onde são centrados quase todos os resultados. Diversas instruções permitem manipular seu controle.

Na figura 10-7 temos a pinagem do 8048. A alimentação de 5 Volts dc, regulada, é ligada aos pinos 40 (Vcc) e VDD(26). O terra é conectado ao pino 20 (VSS) (uma bateria de baixa potência, ligada ao pino 26 manteria o conteúdo da RAM em caso de falha na alimentação-back up).

O microprocessador possui circuito interno de clock, bastando conectar um cristal do pino 2 (XTAL 1) ao pino 3 (XTAL 2). A frequência máxima do clock é 6 MHz, resultando em execução de instruções em 2,5 microsegundos (1 ciclo) ou 5 microsegundos (2 ciclos). Querendo economizar o cristal o usuário pode omiti-lo, ligando um resistor de

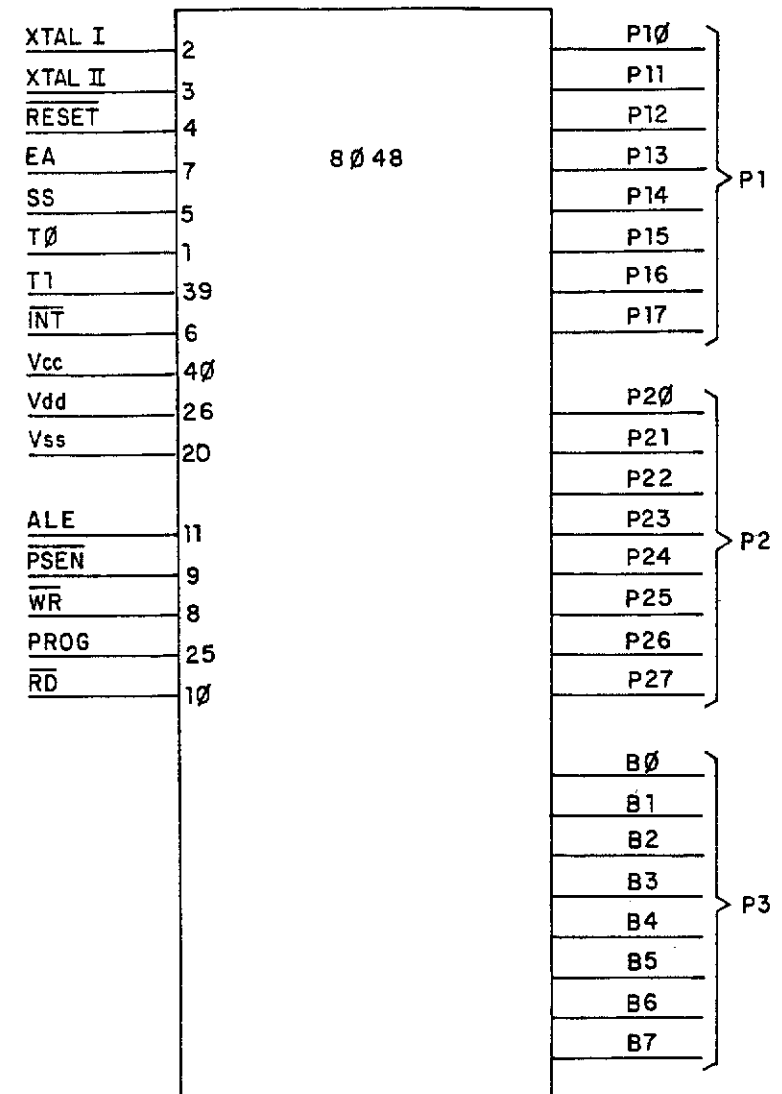


figura 10-7 - Microprocessador 8048.

20 OHMs entre XTAL 1 e XTAL 2 e acoplado este último terminal ao terra através de um capacitor de 20 pFarad. A frequência obtida será de 3MHZ, mas com precisão menor. O ODISSEY usa uma terceira alternativa, acoplado o sinal recebido do clock ao pino 2 XTAL(1), e deixando o pino 3 XTAL(2) flutuando.

Quando o 8048 é ligado ou ao se colocar nível 0 no pino 4 - RESET - ele executa a instrução cujo endereço está gravado na posição 0 de sua ROM (endereço 000). Enquanto RESET = 0 o endereço 00 continuará a ser acessado. Portanto, durante a execução normal do programa o pino 4 deve ser mantido em nível alto (RESET = 1). Para resetar (reiniciar) um jogo em andamento o botão RESET do teclado é acionado, descarregando o capacitor de 1 microFarad que mantém a tensão de nível 1 (figura 10-8).

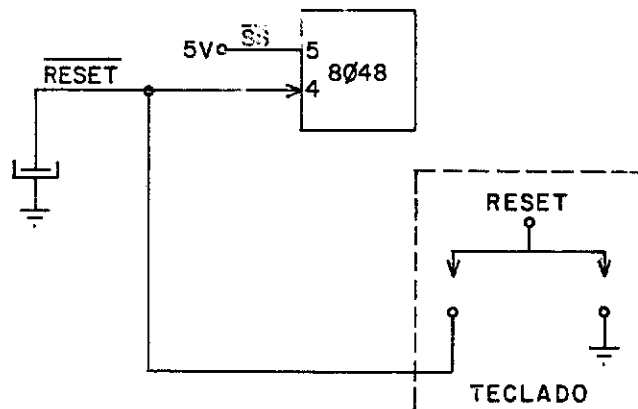


figura 10-8 - RESET do ODISSEY.

Se levado a nível 0 o pino 5 ($\overline{SS}$) pára a execução do programa, aguardando uma mudança para nível 1, quando então retornará a continuidade da programação (no mesmo ponto em que o deixou, ao contrário do RESET que volta ao início do programa). No ODISSEY ele é mantido constantemente em 5 Volts, portanto desativado (figura 10-8). Este terminal é extremamente útil em consertos, permitindo parar o jogo num determinado instante (aterrando-o) e com a ponta de prova lógica conferir todos os estágios, à procura de falhas.

TO e T1 são terminais de testes externo (figura 10-9), existindo diversas instruções que os empregam (exemplo; JTO 'ENDEREÇO', salte para o endereço se TO = 1; JNT0 'ENDEREÇO', salte para o endereço indicado se TO = 0). T1 também pode exercer outras funções de contagem de eventos, como veremos a seguir. TO é ligado ao pino 1 do conector de cartucho, enquanto T1 recebe sinais de blanking do display gráfico, indicando ao processador qual linha da tela está sendo varrida e o momento exato do retraço.

INT é outro pino para teste externo, causando uma interrupção no programa quando é levado ao nível 0. Nesta condição ele desvia a programação para o endereço 03H (na ROM), onde deve estar gravado o endereço de uma rotina de interrupção. No ODISSEY ele é ligado ao pino 2 do display gráfico (figura 10-10).

PSEN ("program store enable", habilitação de armazenamento de programa) é levado a 0 toda vez que há uma busca de instrução na memória externa. Ou seja, após executar as 1.024 bytes da ROM interna o 8048 passa a buscar instruções na ROM externa, emitindo o sinal PSEN em cada ciclo de busca.

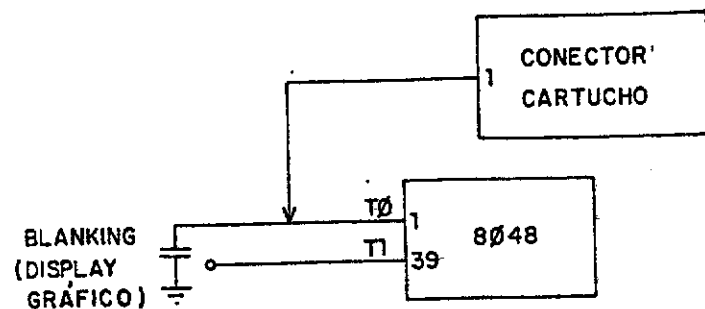


figura 10-9 - Terminais de testes externos T0 e T1.

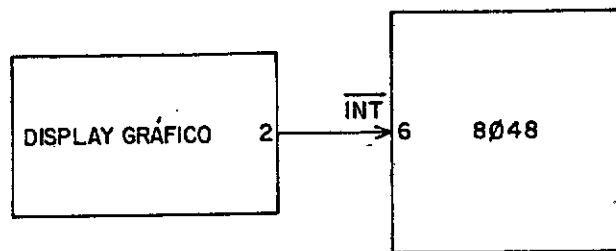


figura 10-10 - Terminal de interrupção $\overline{INT}$.

No ODISSEY o $\overline{PSEN}$ é ligado ao pino F da ROM cartucho (figura 10-11), funcionando como $\overline{OE}$ ("output enable", habilitação de saída)

PROG, pino 25, é usado em expansões do 8048, não tendo utilidade no ODISSEY e sendo mantido flutuando.

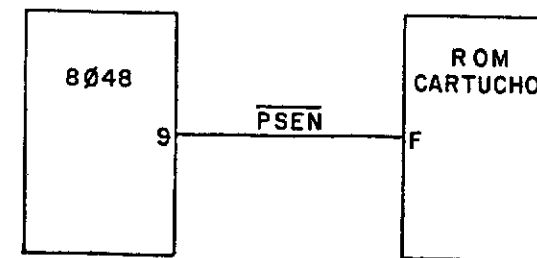


figura 10-11 - Terminal de busca ("fetch") externa $\overline{PSEN}$.

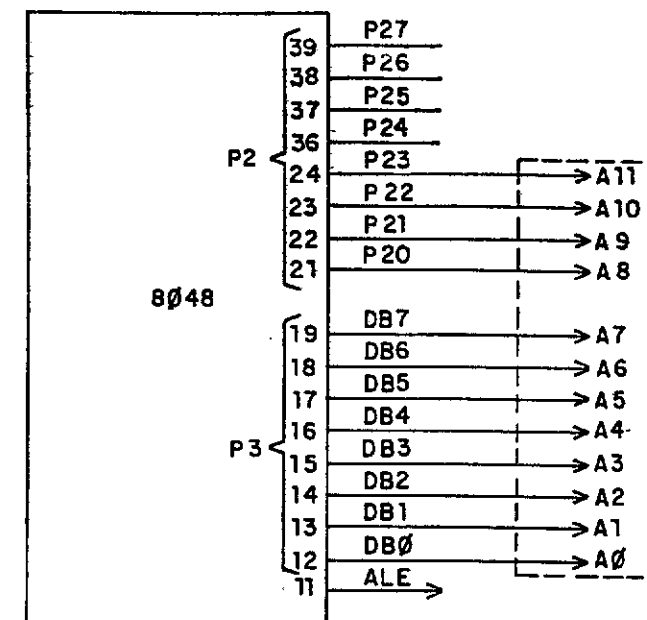


figura 10-12 - Formação de endereços externos no 8048.

O 8048 possui 3 grupos de portas. A porta 1 engloba os pinos 27 (P10), 28 (P11), 29 (P12), 30 (P13), 31 (P14), 32 (P15), 33 (P16), 34 (P17). A porta 2 composta pelos pinos 21 (P20), 22 (P21), 23 (P22), 24 (P23), 35 (P24), 36 (P25), 37 (P26), 38 (P27). A "porta" 3 contém pinos 12 (DB0), 13 (DB1),

14 (DB2), 15 (DB3), 16 (DB4), 17 (DB5), 18 (DB6), 19 (DB7). Instruções específicas permitem a entrada ou saída de 1 byte através de uma das portas (exemplos IN A, P1 coloca os 8 bits existentes nos pinos da porta 1 no acumulador; OUTL P2, A faz o inverso com a porta 2, colocando nos seus pinos o byte do acumulador).

Embora tratados como "porta 3" os pinos 12 a 19 (DB0 a DB7) formam na verdade uma barra de dados, encarregada da entrada e saída de dados em instruções que envolvam acesso a memória externa (desviando-se da ROM e RAM interna). Nessas instruções o pino 11 - ALE ("address latch enable", habilitação de latch de endereços) fica em nível 1 e o endereço externo a ser acessado é emitido nos 8 pinos de P3 mais os pinos P20, P21, P22 e P23 de P2. Quando ALE volta a nível 0 P3 apresenta-se novamente como barra de dados. Ou seja, o microprocessador multiplexa endereços e dados em P3 e parcialmente em P2. Os pinos disponíveis para endereços externos perfazem $8 + 4 = 12$, o que possibilita endereçamento externo de até $2^{12} = 4.096 = 4K$.

A figura 10-12 ilustra a formação dos 12 bits de endereços externos.

No ODISSEY são usados dois CIs 74LS175 (figura 10-13), como latch da multiplexação. Cada um deles possui 4 flip-flops que armazenam o bit aparecendo no terminal D (1D, 2D, 3D, 4D), mas somente quando recebem sinal de clock no pino 9. Os 8 bits de P3 são conectados aos terminais de dados dos flip-flops (figura 10-14); ALE é invertido numa porta NAND (74LS00, figura 10-13), aparecendo nos pinos 9 dos dois latches como clock. Assim, quando o 8048 quer receber ou transmitir dados para o exterior sem usar as instruções de portas ele coloca o endereço a ser acessados nos 8 terminais de P3 mais

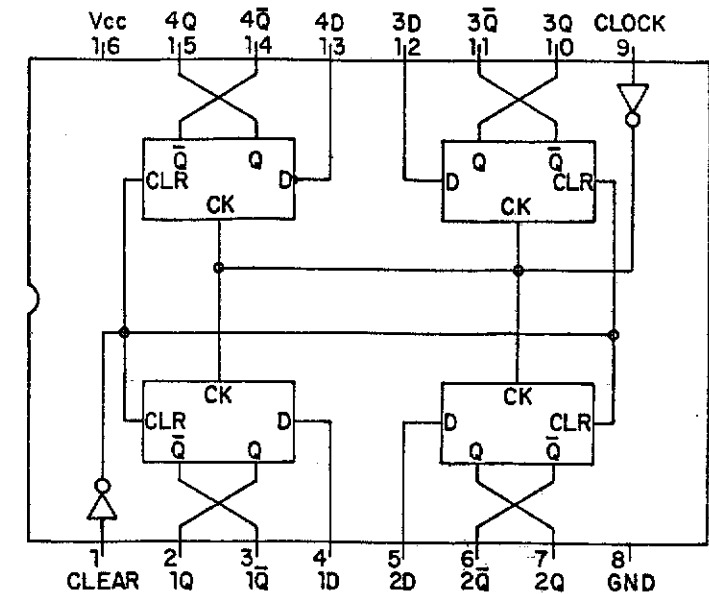


figura 10-13 - CI 74LS175.

4 em P2 e emite o sinal ALE; este faz com que os bits B0 a B7 entrando nos latch fiquem armazenados na saída (bits A0 a A7) como endereços. A seguir o microprocessador emite 8 bits de dados nos mesmos terminais (pinos 12 a 19), mas sem fazer ALE=1. Na falta de sinal de clock adequado os latches não armazenam esse byte, que fica disponível para a barra de dados simultaneamente com os endereços A0 a A7.

Para saída de dados (incluindo output nas portas IO e escrita em P3) o microprocessador leva a nível 0 o pino 10 - $\overline{WR}$ ("write", escrita). Para entrada de dados (incluindo input nas portas IO e leitura em P3) ele coloca em 0 o pino 8 - $\overline{RD}$ ("read", leitura).

Finalmente temos o pino 7 - EA ("ex-

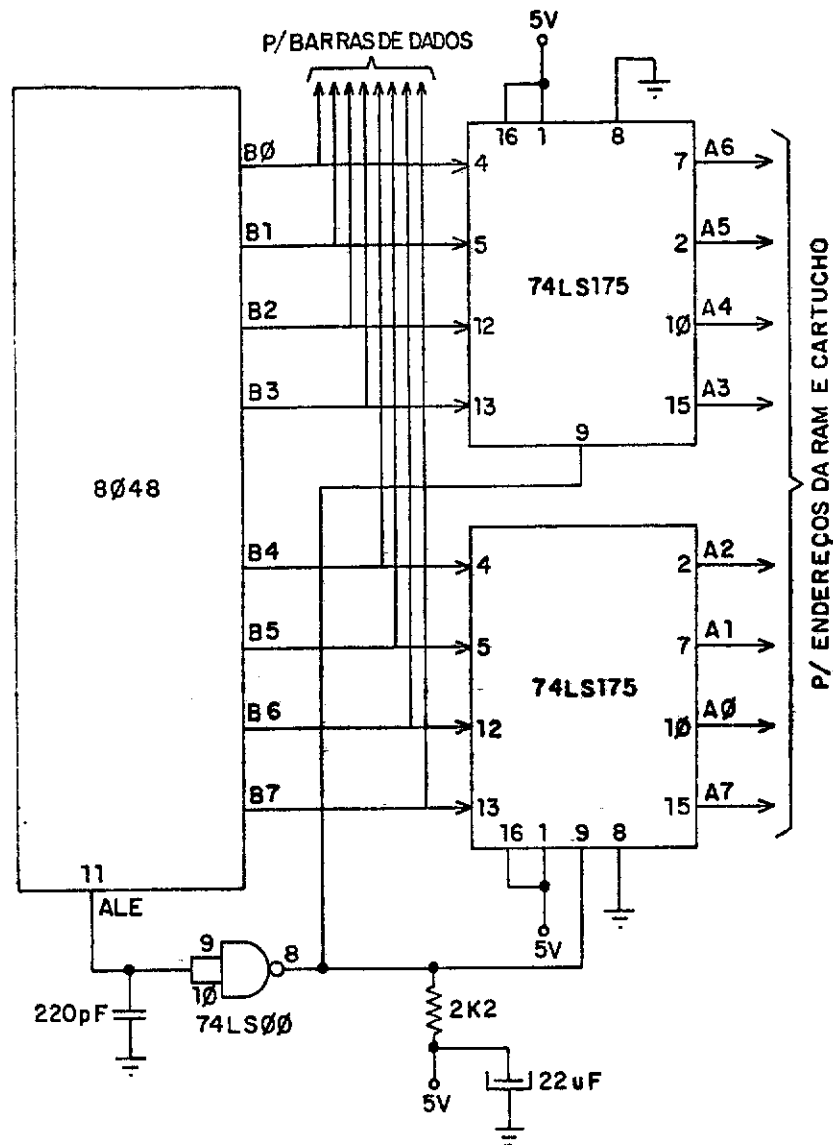


figura 10-14 - Multiplexação de dados e endereços no ODISSEY.

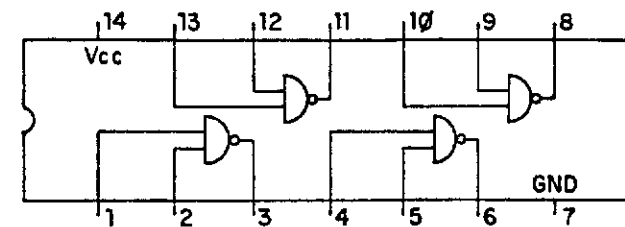


figura 10-15 - CI 74LS00.

ternal acess", acesso externo), que em nível 1 des - via o microprocessador da memória interna e o faz trabalhar exclusivamente com dados externos. Esse terminal é útil em debulhamento e teste do sistema, mas não tem utilidade no ODISSEY, sendo mantido aterrado.

A lista de instruções do 8048 - ASSEMBLER e Código de máquina - consta no apêndice B.

6 - RAM 6810

A RAM estática 6810 (figura 10-16) - MOTOROLA - tem capacidade de 128 bytes, endereçados por 7 bits de endereços (A0 = pino 17, A1 = pino 18, A2 = pino 19, A3 = pino 20, A4 = pino 21, A5 = pino 22, A6 = pino 23). A barra bidirecional de dados é assim distribuída: B0 = pino 8, B1 = pino 7, B2 = pino 6, B3 = pino 5, B4 = pino 4, B5 = pino 3, B6 = pino 2. Os terminais $\overline{CS}$ (seleção de chip) 1 e 12 são aterrados junto com o pino 14, e $\overline{CS}$ - 10 - junto com 5 Volts do pino 24. Há mais 3 terminais de habilitação $\overline{CS1}$ (pino 11), $\overline{CS3}$ (pino 16) e $\overline{CS5}$ (pino 15). No pino 13 encontramos $\overline{W/R}$, que em nível 0 indica WRITE (escrita) e em 1 sinaliza READ (leitura).

Os 8 pinos de dados são ligados diretamen-

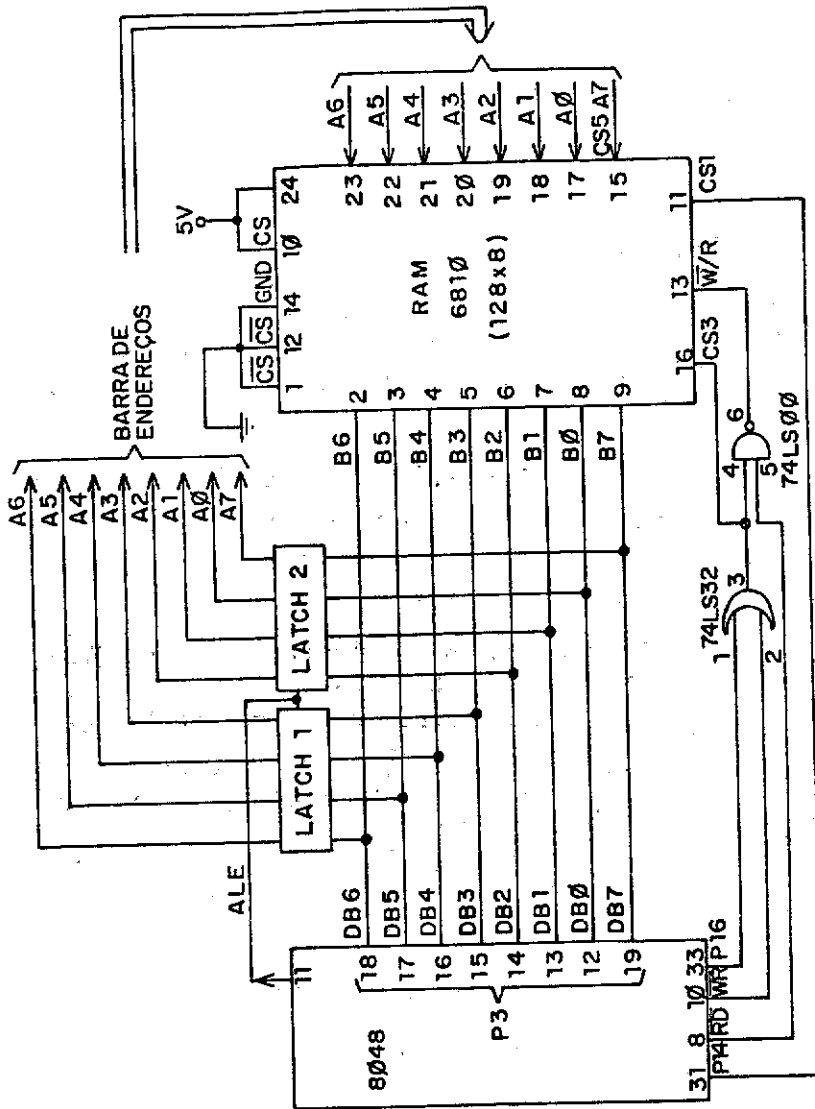


figura 10-16 - Circuito RAM de vídeo.

te a P3 do microprocessador, enquanto os 7 bits de endereços mais CS5 são retirados dos dois latches, permitindo multiplexação de endereços e dados. Como CS5 usa lógica negativa os endereços da RAM estarão em 0XXX XXXX. Lembre-se que esses endereços são obtidos dos registros R1 e R2 da RAM interna do 8048.

Para que a RAM seja habilitada é preciso que P14 (pino 14) seja levado a nível 0, ativando $\overline{CS1}$. Além disso é necessário que CS3 esteja em nível 1, o que é conseguido durante operações de leitura e escrita na RAM mantendo P16 ou $\overline{WR}$ em nível 1 no pino 1 da porta OR (74LS32, figura 10-17).

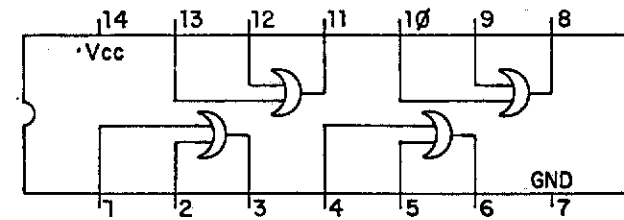


figura 10-17 - CI 74LS32.

Na leitura $\overline{RD}$ vai a nível 0 na entrada do pino 5 da porta NAND, resultando 1 na saída pino 6 (independente da outra entrada, pois 0 NAND 1 = 1 e 0 NAND 0 = 1), o que seleciona a operação R ("read", lógica positiva da dupla $\overline{W}/R$). Na escrita $\overline{WR}$ vai a zero no pino 2 da porta OR mas a saída desta continua sendo 1 devido a P16; como $\overline{RD}$ é mantido em 1 as duas entradas da NAND estarão em 1 e sua saída será 0 (1 NAND 1 = 0), selecionando operações $\overline{W}$ ("write", lógica negativa na dupla $\overline{W}/R$).

7 - ROM - CARTUCHO

Quando o cartucho é encaixado no con-

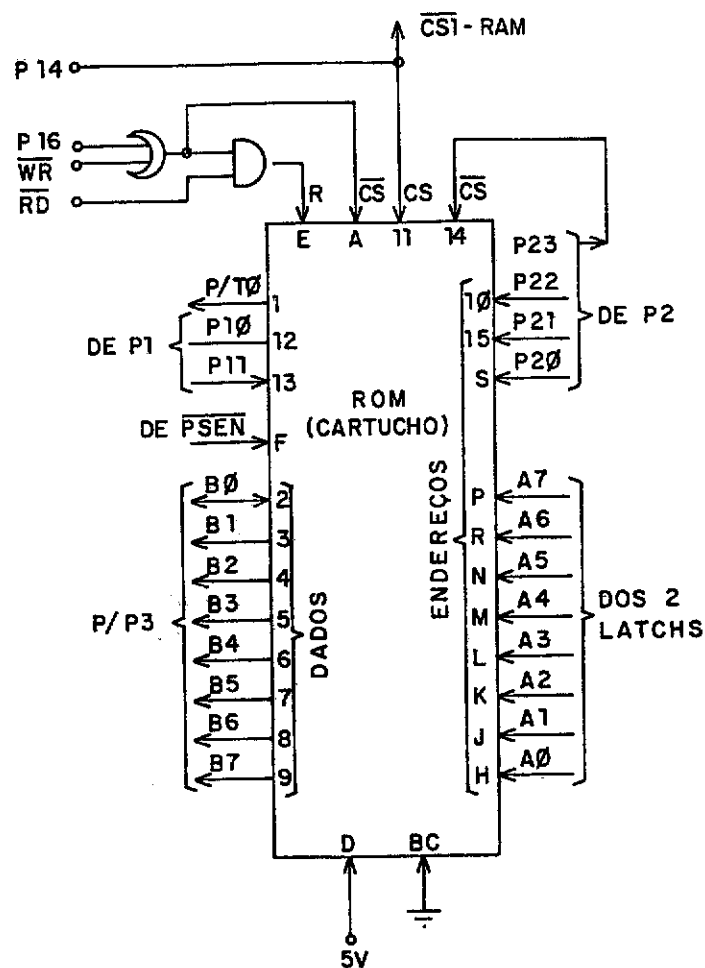


figura 10-18 - ROM - cartucho.

sole do ODISSEY (conector macho do cartucho x conector fêmea do console) sua ROM é incluída no circuito como visto na figura 10-18.

Durante a busca de uma instrução na ROM - cartucho (após a execução dos 1.024 bytes da ROM interna) o 8048 emite o sinal $\overline{\text{PSEN}}$, que exerce o papel de "LEITURA DE MEMÓRIA", habilitando a transferência de dados.

Os 8 bits inferiores do endereço a ser buscado são postos na porta P3 mais 4 bits na porta P2. Destes últimos P23 é usado como seleção de chip $\overline{\text{CS}}$. Temos portanto 11 bits de endereços (A0-A7, P20, P21, P22), caracterizando a ROM de $2^{11} = 2k$ bytes. Quando ALE é levado a 1 esse endereço fica "preso" nos dois latches e dão entrada na ROM. ALE volta a 0 e o endereço retirado da porta P3, na qual a ROM assenta o byte conteúdo da posição endereçada.

8 - TECLADO (KEYBOARD)

O teclado funciona no sistema de varredura (figura 10-19), com um CI decodificador (74 LS156, figura 10-20) e outro codificador (74 LS148, figura 10-21).

O decodificador, ao receber o sinal de strobe de P12 vai sucessivamente baixando o nível das fileiras de matriz 6 x 8 do Keyboard (de 1 a 6). Cada fileira tem 1 byte (de 7 a 14 no Keyboard) conectado à entrada no codificador (pinos 4, 3, 2, 1, 13, 12, 11, 10). Estes normalmente estão em nível 1, devido à tensão de 5 Volts. Uma tecla apertada em fileira com nível 0 aterra o bit de entrada no codificador, que traduz a posição do bit (de 0 a 7) em número binário de 3 bits (1/0 1/0

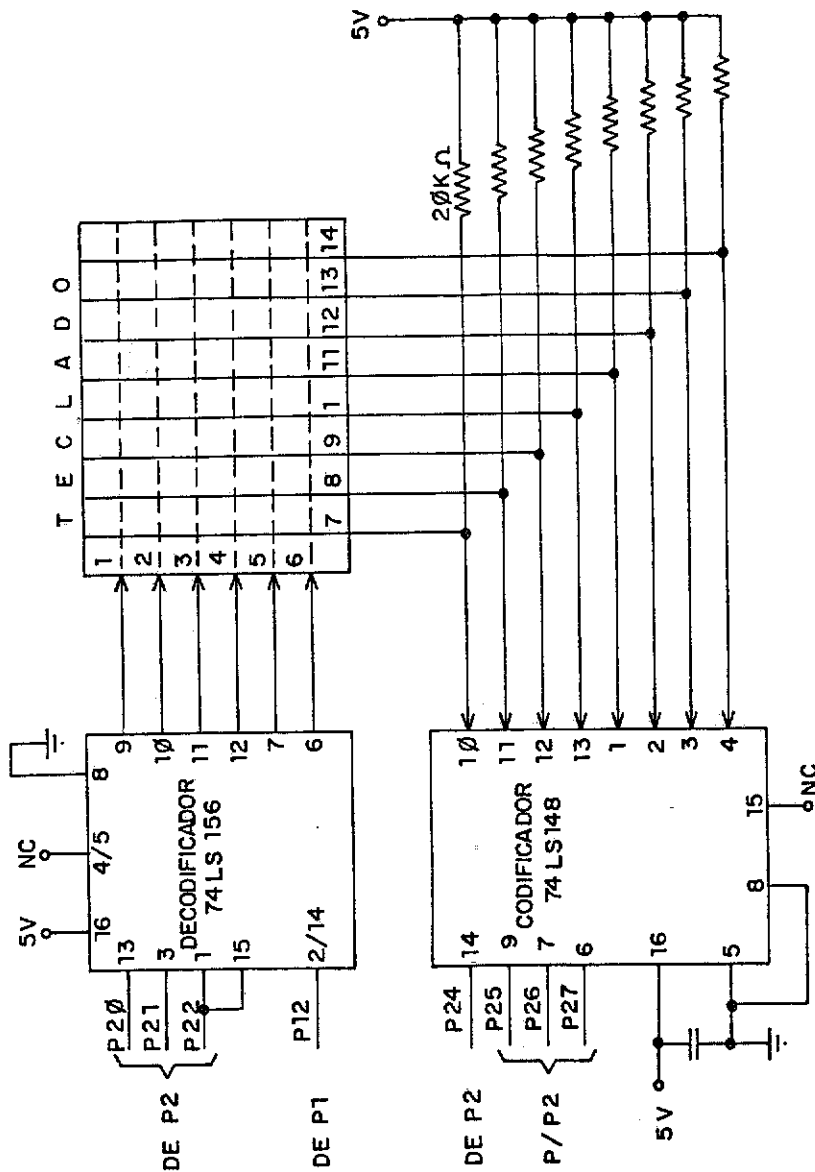


figura 10-19 - Circuito do teclado.

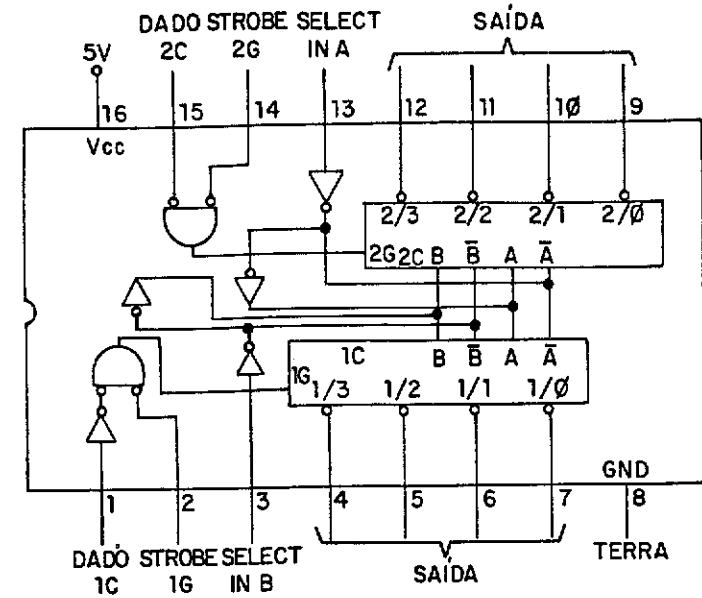


figura 10-20 - CI 74LS156 - decodificador.

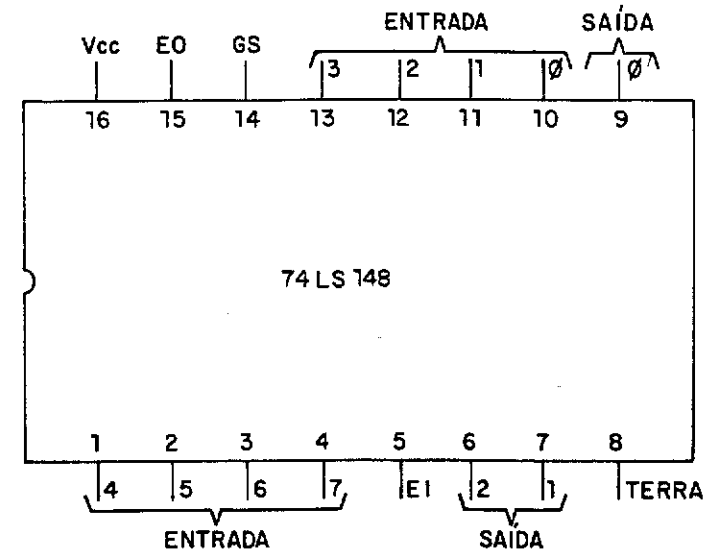


figura 10-21 - CI 74LS148 - codificador.

1/0), apresentando-os nos pinos 9, 7, 6 (na ordem : B2 = pino 6, B1 = pino 7, B0 = pino 9), que são encaminhados para P2 através de P25, P26 e P27. O microprocessador reconhece os bits recebidos, processando-os adequadamente. P24, pino 14, apenas seleciona o chip.

A tecla Reset ao ser pressionada altera a linha para pino 5 do microprocessador, ativando seu Reset.

9 - JOYSTICK

Os dois joysticks operam digitalmente fechando o contacto correspondente à direção em que são manipulados e dispondo também de um botão de ação. Eles são encaixados ao circuito por meio de 2 conectores existentes no console (figura 10-22).

Inicialmente as entradas do buffer 74LS365 (pinos 14, 12, 10, 4, 6) são mantidas em nível 1 pela voltagem de 5 Volts recebida através dos resistores de 20 KOhms. Quando o joystick é manipulado fecha-se um dos seus contactos, ligando a entrada no buffer com a pequena resistência de 100 Ohms, o que caracteriza quase atterramente, sinal 0. O mesmo esquema é válido para o botão de ação.

O buffer 74LS365 (figura 10-23) só permite ser atravessado pelos sinais do joystick quando seus pinos 1 e 15 recebem pulsos em nível 0. Os pinos dos dois buffers são ligados ao terminal RD do microprocessador. Os pinos 1 de buffer dos jogadores LEFT e RIGHT são conectados aos terminais 10 e 9, respectivamente, do decodificador do teclado. A cada 20 milisegundos, aproximadamente, o 8048 emite o sinal RD e aciona o decodificador. Quando o

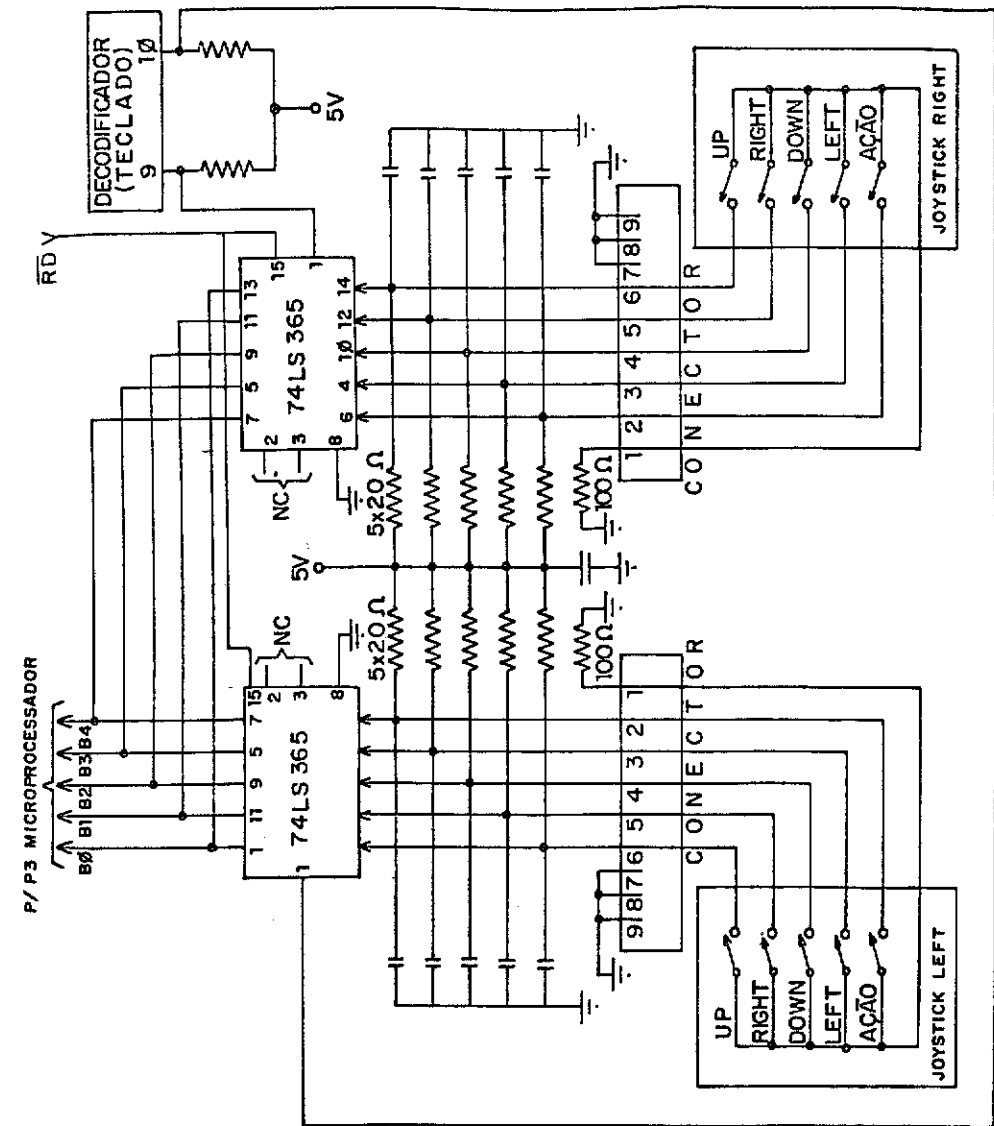


figura 10-22 - Circuito dos joysticks.

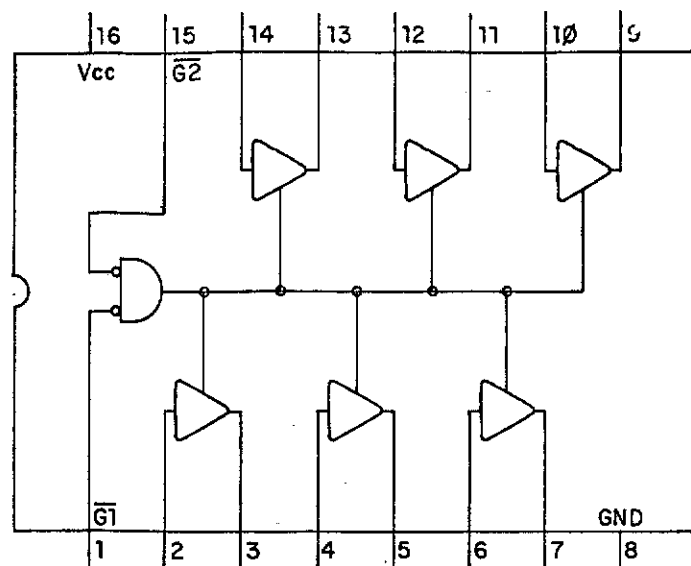


figura 10-23 - CI 74LS365 - buffer

terminal 9 deste é levado a 0 as duas entradas do buffer RIGHT estão em 0, habilitando-o e fazendo os sinais do joystick atravessá-lo. Em seguida o terminal 10 do decodificador vai a 0, juntamente com $\overline{RD}$ habilitando o buffer LEFT. Observe que quando o buffer não está habilitado sua saída será tristate.

Os dados recebidos dos joysticks através dos buffers dão entrada no microprocessador pela porta P3 (B0, B1, B2, B3, B4), sendo devidamente processados e em consequência alterada a imagem na tela.

10 - DISPLAY GRÁFICO

O microprocessador envia endereços e dados ao display gráfico através dos 8 terminais de

P3 mais P13 (figura 10-24). Não é necessária interface para multiplexação de dados e endereços, pois o display dispõe de latch interno, bastando conectar ALE para a devida sincronização. Os sinais de blanking são conectados a T1 (pino 39), servindo como contador de registro TIMER/CONTADOR do microprocessador, de maneira que este sempre sabe a linha da tela atualmente sendo varrida pelo feixe eletrônico. $\overline{WR}$ é usado em nível baixo para endereçar células na ROM, enquanto $\overline{RD}$ é combinado numa porta OR com P16.

O display gráfico recebe o sinal do circuito do clock no pino 1 e segundo ele gera todos os sinais de cor (RED/vermelho - pino 15, GREEN/verde - pino 19, BLUE/azul - pino 20), sincronismo vertical e horizontal (pino 5), burst (pino 4), blanking vertical (pino 8) e horizontal (7), sendo os dois últimos combinados numa porta OR e deles tirada uma saída para T1 (que incrementa a contagem do registro TIMER/CONTADOR do microprocessador a cada blanking). O sinal de luminância é gerado no pino 10 e invertido na porta NAND, mas só aparece na saída da porta NAND seguinte quando há uma habilitação (nível 1) de P17. Todos os sinais acima são encaminhados ao codificador de cor, enquanto o sinal de som saindo do pino 27 é ligado diretamente ao modulador de RF.

11 - CODIFICADOR DE COR

O codificador de cor recebe os sinais B-Y, R-Y e burst de 3,58 MHz e devidamente defasados - da chave PAL e seleciona-os conforme as entradas RED (vermelho), GREEN (verde), BLUE (azul), BURST, BLANK, Y (luminância) e SINC originadas no display gráfico.

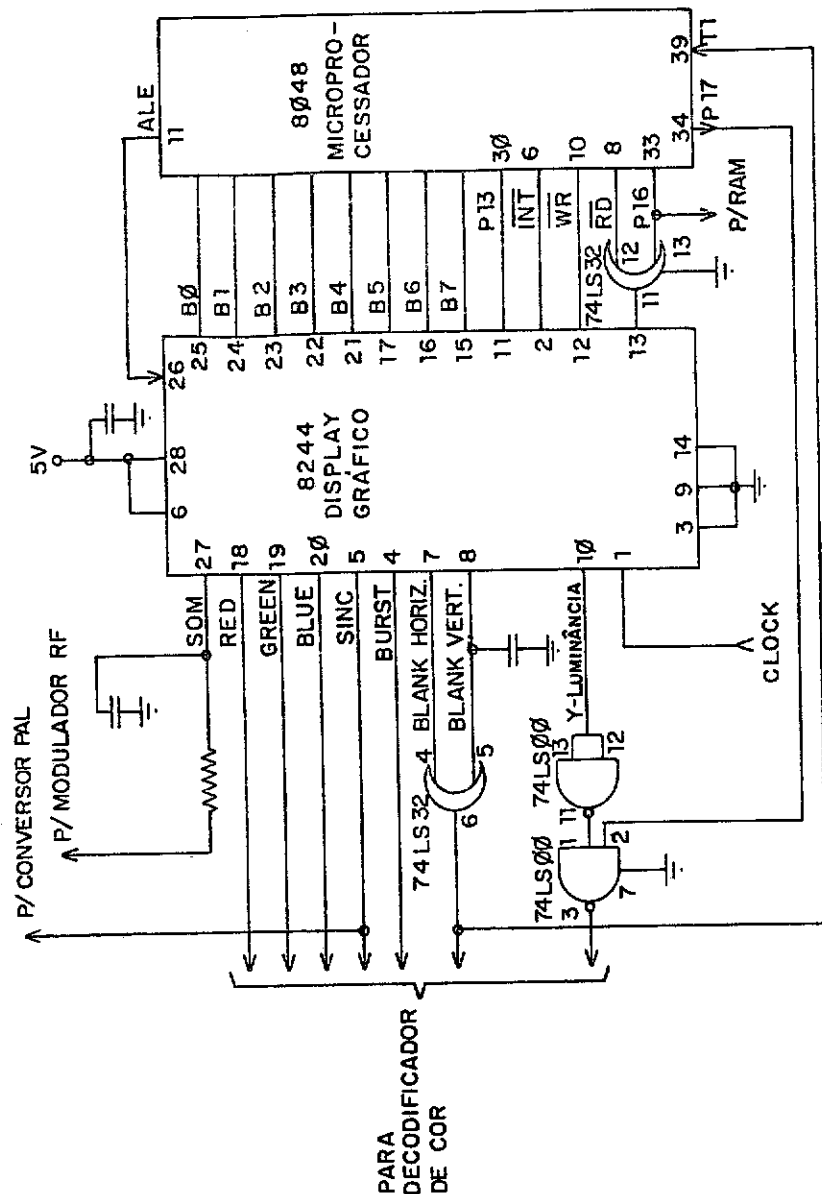


figura 10-24 - Circuito do display gráfico.

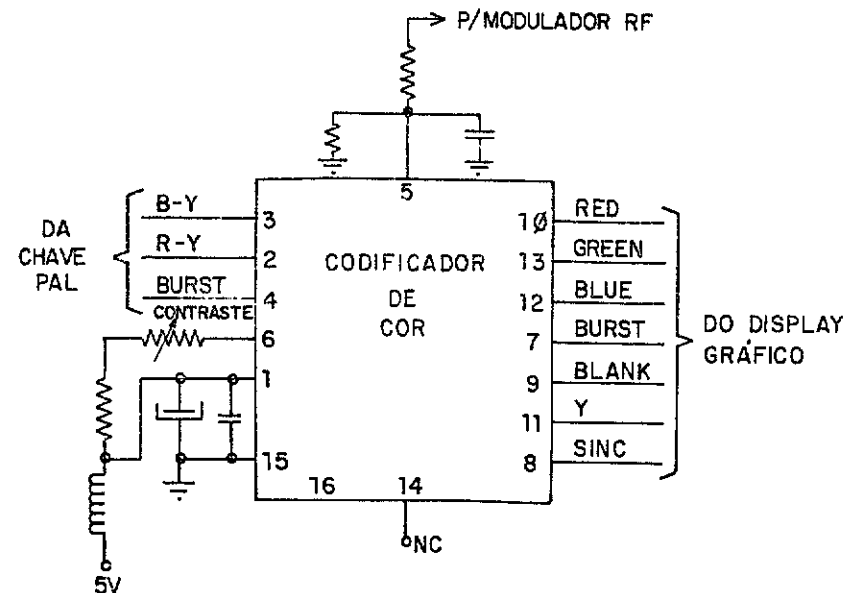


figura 10-25 - Codificador de cor.

Desta forma a saída do codificador de cor (pino 5) é o sinal de vídeo composto, incluindo Y + C + SINC + BURST, que é enviado ao modulador de RF para modular a portadora de um canal.

O potenciômetro ligado ao pino 6 permite o controle do contraste (diferença entre branco e preto na tela).

12 - CHAVE PAL

A chave PAL é uma placa separada acrescentada para conversão do sistema NTSC ao PAL. Ela pode ser adquirida também separadamente, pois não faz parte do circuito original.

Como entradas (figura 10-26) a chave PAL recebe o sinal de sincronismo composto existente

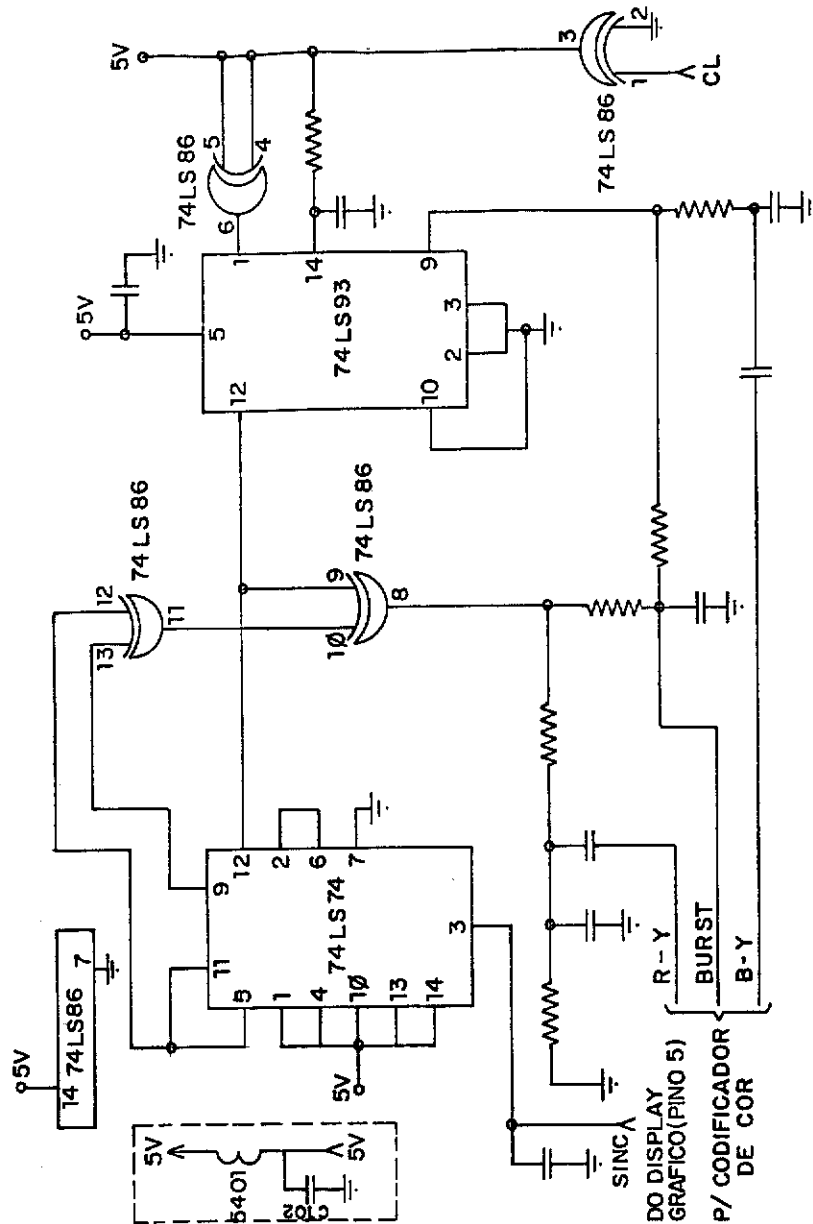


figura 10-26 - Circuito da chave PAL.

no pino 5 do display gráfico e o clock de 7,151222 MHz gerado no coletor do transistor do circuito de clock.

Como saídas ela fornece para o codificador de cor os sinais B-Y, R-Y e burst, todos de 3,58MHz e já na devida fase.

13 - MODULADOR DE RF

O modulador de RF (figura 10-27) recebe os sinais de Y, C, SINC e BURST do codificador de cor, bem como o som de display gráfico, modulando-os na frequência portadora dos canais 3 ou 4. Sua saída é ligada diretamente à antena do receptor.

O aparelho é originalmente chaveado para o canal 3. Caso o usuário prefira o canal 4 basta comutar a chave S-3 existente no modulador de RF.

A impedância na saída é 300 Ohms (compatível com antena interna do receptor; a externa é 75 Ohms).

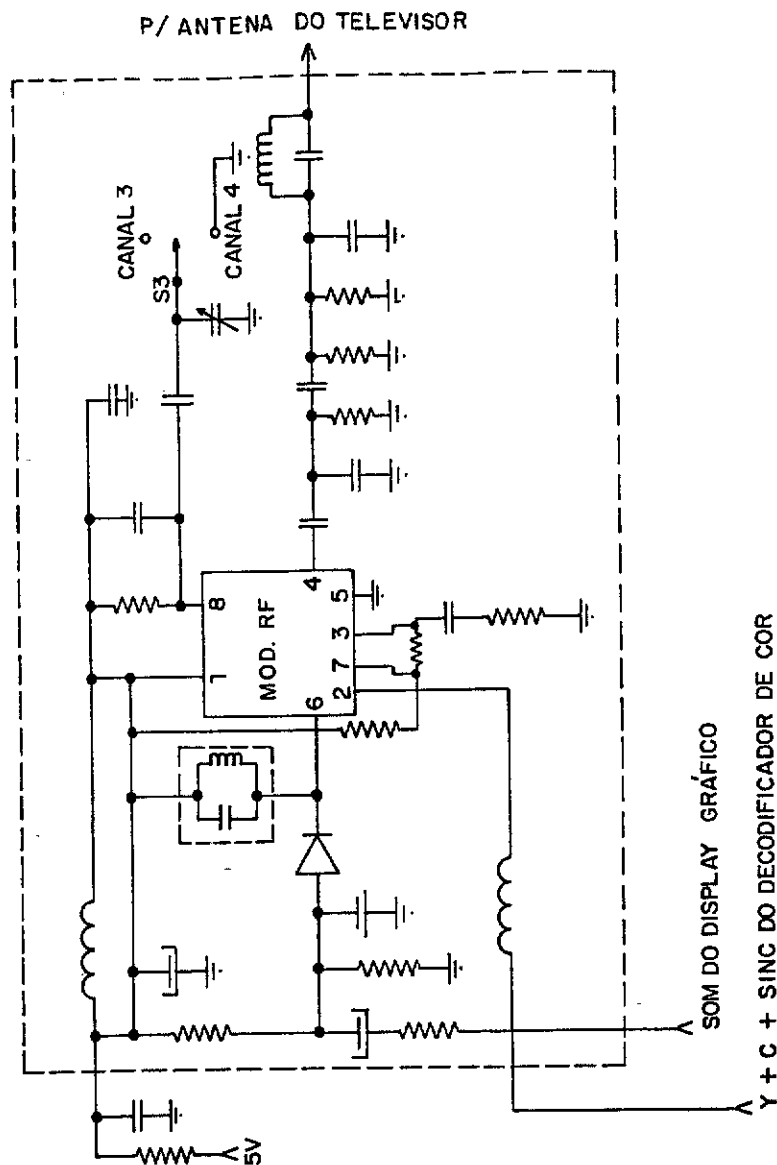


figura 10-27 - Circuito do modulador de RF.

CAPÍTULO 11 – TÉCNICAS E INSTRUMENTOS PARA MANUTENÇÃO DE VÍDEO-GAMES

1 - ABORDAGEM DO DEFEITO

A abordagem do defeito é a parte mais crítica do conserto e dela depende o sucesso do serviço. Ao contrário de aparelhos analógicos onde as operações chegam a ser "visíveis" (no sentido que os sinais são facilmente percebidos) encontramos em equipamentos digitais um fluxo de sinais que são registráveis se nós sabemos com antecedência todas suas características. Isto é, na área digital a habilidade (embora importante) não é suficiente para consertos, sendo indispensável o conhecimento teórico.

O roteiro de serviço mais recomendável é:

- 1 - Análise funcional: a partir da observação do desempenho defeituoso determinar qual estágio poderia provocar esse defeito; aqui não se usa instrumentos, mas simplesmente aplica-se deduções teóricas. Como vídeo-games são adaptados para televisão a tela será a melhor referência para essa análise.
- 2 - Teste de estágio: qualquer estágio apresenta na saída um sinal que depende dos sinais colocados na entrada. Os instrumentos de conserto devem simular essa situação, colocando sinais conhecidos na entrada do estágio e registrando sua saída; se ela não apresentar o resultado esperado

configura-se estágio defeituoso, que é então se parado. Note que a análise funcional serve para determinar qual estágio deve ser testado, caso contrário teríamos de testar a todos e sempre restaria a dúvida se foi feito o teste adequado

- 3 - Teste local: o estágio separado é agora examinado detalhadamente, examinando-se não só se ele está defeituoso como também se opera em condições adequadas.

2 - INSTRUMENTAÇÃO PARA CONCERTO

Vídeo-games são microcomputadores, e em princípio os mesmos instrumentos e técnicas seriam aplicáveis a ambos. Entretanto, o baixo custo do vídeo-game e a característica "de balcão" do seu concerto tornam inadequados para ele os instrumentos e técnicas usadas para micros, que exigem pesados investimentos e conhecimentos altamente especializados. A título de orientação geral apresentaremos a seguir quais são esses instrumentos, mas na realidade só adotaremos três deles no estudo de manutenção do ATARI e ODISSEY (além, é claro, do osciloscópio e multímetro, indispensáveis em qualquer serviço de Eletrônica).

- ROM DE DEBUGGING (cartucho teste): é uma ROM onde está gravado um programa que testa os estágios e indica na tela as prováveis causas do defeito. Entretanto, seu uso só é possível se o aparelho tem seus principais estágios funcionando regularmente - o que é muito improvável quando ele é levado para concerto. As fábricas de vídeo games dispõem de ROM de debugging (cartucho teste) com a respectiva documentação, como mostraremos para o ATARI e o ODISSEY.

- PONTA DE PROVA LÓGICA: é a mais útil e indispensável ferramenta em Eletrônica Digital. Quando encostada num ponto do circuito indica qual o nível lógico (1,0 ou tristate).

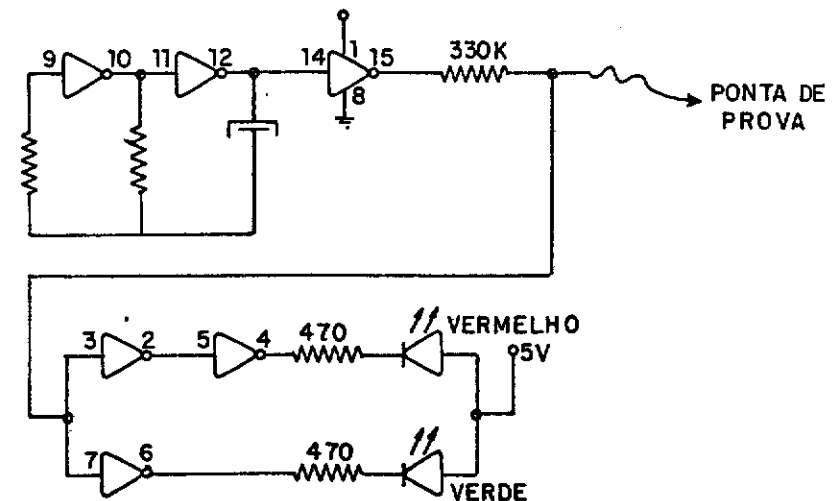


figura 11-1 - Circuito da ponta de prova lógica.

O circuito de uma ponta de prova simples (e boa) é visto na figura 11-1, com o respectivo lay-out de circuito impresso em 11-2. Com o LED verde aceso temos nível 1, LED vermelho aceso nível 0; os dois piscando alternadamente será tristate.

- TESTER ESTATICO/COMPUTESTER: a maior dificuldade no concerto de aparelhos digitais é conseguir seguir o grande número de sinais, de frequência alta e irregular. Uma solução possível é substituir o microprocessador - fonte desses sinais - por um aparelho - o COMPUTESTER (figura 11-3) - que passa a emití-los sob controle manual do operador.

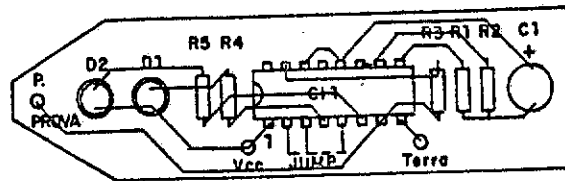


figura 11-2 - Lay-out

O microprocessador é retirado e seu soquete preenchido com os terminais do COMPUTESTER (16 de endereços - A0/A15, 8 de dados - D0/D7, 3 de controles - C1/C3). POWER e GND são ligados diretamente à fonte de alimentação (+5 Volts de regulados e terra). O operador faz no COMPUTESTER o chaveamento correspondente a endereços e dados desejados, a seguir assenta as chaves C1/C3 conforme a operação (leitura, escrita, input, etc). Com a ponta de prova lógica pesquisa-se o resultado, que é registrado por dois LEDs (R-red e G-green), indicando 1,0 ou tristate. Em casos de entrada de dados (leitura ou input) a 9a. chave desabilita todo o conjunto de chaves de dados, evitando conflito de sinais.

3 - MANUTENÇÃO DO ATARI

O ATARI é baseado num número reduzido de CIs. Nesta condição o processo "troca-peça" é vantajoso. Iniciaremos por ele, apresentado em seguida outras técnicas. Pressupomos que a fonte de alimentação já foi checada e está correta.

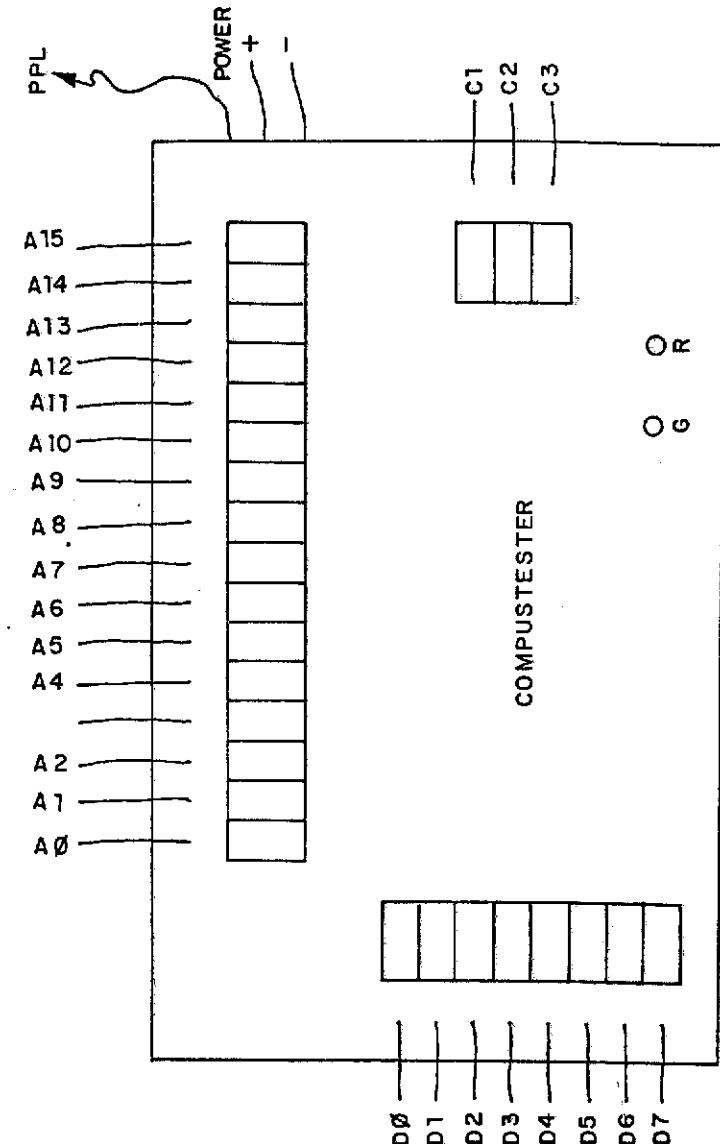


figura 11-3 - Tester estático - COMPUTESTER

SUBSTITUIÇÃO DE PEÇAS

Um ATARI em perfeito estado de funcionamento é mantido de reserva na bancada. O microprocessador do ATARI defeituoso é retirado e colocado no ATARI reserva. Se este último continuar funcionando normalmente é sinal que o microprocessador está bom. Caso contrário encaixe o microprocessador reserva no ATARI defeituoso e confirme se o defeito desaparece. A operação é repetida com a TIA e PIA, um CI de cada vez. Na maioria dos casos isso será suficiente para detectar e corrigir a falha. A essa altura o leitor se perguntará porque não substituir CIs nos ATARI diretamente no aparelho defeituoso. A questão é que problemas de mau contacto podem levar a falsos resultados, trocando-se desnecessariamente CIs (que são bastante caros) e deixando dúvida se o defeito voltará após certo tempo. A substituição noutra consola é, assim, muito mais segura (além de que o valor de compra dos três CIs é quase o mesmo de um ATARI usado para reserva).

ACOMPANHAMENTO DE SINAL COM OSCILOSCÓPIO

A figura 11-4 mostra os principais pontos para teste com osciloscópio, que deve ser feito do fim para o começo (da antena em direção ao microprocessador). Do cabo coaxial RF-OUT deve sair o sinal de vídeo composto modulado pela portadora, ou seja, 54-60MHz (canal 2) ou 60-66MHz (canal 3) com blanking, sincronismo, 8 a 11 pulsos de burst da cor e a informação de vídeo propriamente dita. Caso se verifique sinal irregular testa-se a entrada do modulador RF. No pino 2 deve ser encontrado o sinal de 4,5MHz de som; se o resultado for negativo volta-se aos pinos 12 e 13 da TIA onde é originado a frequência de áudio, pesquisando-se os componentes intermediários

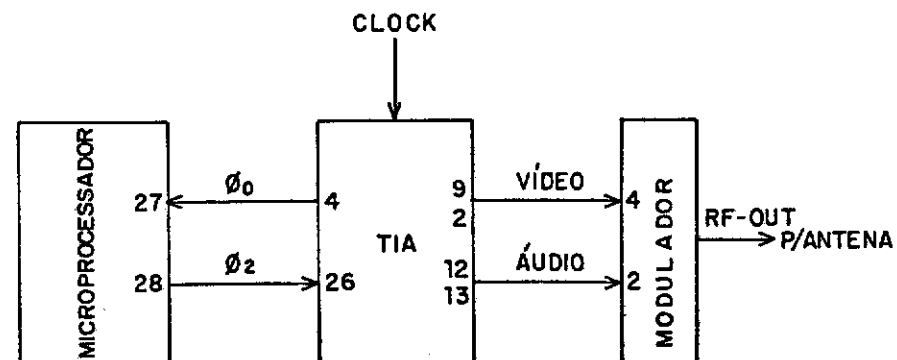


figura 11-4 - Pontos para teste de sinal no ATARI

(capacitores e resistores). No pino 4 teremos um sinal idêntico à saída de RF, porém sem áudio e na frequência de 0 a 6 MHz; ocorrendo irregularidade é feito a pesquisa na origem na TIA - pino 9, cromagem, onde deve ser encontrado o sinal de 3,58MHz e pino 2, sincronismo, com a frequência horizontal de 15.750 HZ e vertical de 60 HZ. Ainda na TIA, pino 11, deve entrar o clock de 3,575611 MHz, que deverá sair dividido por 3 (aproximadamente 1,2 MHz) no pino 4, dando entrada no pino 27 do microprocessador e saindo dele, pino 28, até chegar ao pino 26 da TIA. Uma boa indicação é manter um cartucho-teste (pode ser qualquer jogo) e usá-lo num ATARI em bom estado para testar os pontos acima, acostumando-se com as formas de onda e comparando-as com aquelas obtidas no ATARI defeituoso.

TESTE DE PIA/JOYSTICK

Suporemos a disponibilidade de uma ponta de prova lógica ou um COMPUTESTER. Na falta desses o nível do sinal pode ser medido com o voltímetro (nível 0: até cerca de 1 Volt; nível 1: acima de 2,5 Volts), embora possa ocorrer resultados falsos (por problemas ligados à amperagem e ao tristate); para gerar sinal 1 use uma ligação à fonte + 5 Volts através de um resistor de 4K7 Ohms, para sinal 0 aterre o ponto em questão. A PIA é o estágio de maior aplicação da ponta de prova ou COMPUTESTER. Inicialmente, teste as 5 chaves de controles (DIFFICULTY R e T, COLOR, SELECT, RESET). Na posição OFF suas entradas na PIA (pinos 16, 17, 21, 23, 24, respectivamente) devem apresentar nível 1; acione cada uma das chaves e verifique se as entradas alteram para nível 0. Com os joysticks não acionados suas entradas na PIA (pinos 8 a 15) devem estar em nível 1; mova-os nas 4 direções (R, L, U, D) e verifique se para cada uma delas o respectivo pino na PIA muda para nível 0. Veja se o pino 35 da TIA está em nível 1; aperte o botão TIRO do joystick esquerdo e teste se há alteração para nível 0. Repita a operação para pino 36 da TIA e joystick direito. Se houver irregularidade nesses testes de joysticks, retire-os; coloque sinal 1 numa das entradas do conector de joystick (terminais 1, 2, 3, 4, 6) e teste se ele chega ao pino correspondente na PIA ou TIA; refaça o teste com sinal 0; repita para as 5 entradas. Se todas as entradas passam no teste a falha está no próprio joystick.

TESTE DO CONECTOR DE CARTUCHO

O conector de cartucho é provavelmente a parte que mais apresenta defeitos, devido ao fato de ficar em contato direto com o usuário. Para decidir se a falha é do conector ou do próprio cartucho basta substituir este último, usando outro em reconhecido bom

estado. Sendo do conector abra um cartucho (descole o papel propaganda e desaperte o parafuso debaixo dele) e encaixe no console somente a placa de circuito impresso com a ROM, sem as duas tampas. Retire o microprocessador e ligue o vídeo-game. Veja se a ROM no circuito impresso está recebendo alimentação normal (+ 5 Volts no pino 24 e terra nos pinos 12 e 21). Coloque sinal 1 no terminal do soquete do microprocessador correspondente a A0 (terminal 5) e teste se este nível chega até A0 da ROM (pino 8); refaça para sinal 0; repita para todos os terminais de endereço e de dados (para este teste retire o processador).

SINTOMAS DE DEFEITOS

As seguintes indicações de sintomas de defeitos é retirada da documentação fornecida pela POLYVOX (GRADIENTE).

<u>DEFEITO</u>	<u>SINTOMAS</u>	<u>CAUSAS POSSÍVEIS</u>
LÓGICA	mã visualização na tela, vídeo com uma só cor ou as seguintes figuras	microprocessador, TIA,PIA,cristal, transistores do oscilador de áudio
	 	
	 	
	linhas verticais com cores	microprocessador, TIA,PIA, conector de cartucho (ver trilhas abertas ou em curto da conexão até o cartucho)

DEFEITO	SINTOMAS	CAUSAS POSSÍVEIS
VÍDEO	vídeo com aparência de neve	regulador de tensão 7805, chave ON/OFF da fonte, modulador de RF.
	vídeo fraco	modulador de RF e cabo de antena
COR	não há cor	cristal, TIA, saída de crominância da TIA, cabo de antena, ajuste do conversor PAL.
	aparece só uma barra de referência no padrão de cor	TIA, trimpot de delay (pino 10 da TIA) capacitor 01 MicroFarad na saída do delay
	não tem a mesma cor nos dois lados da linha de referência do padrão de cor	trimpot de delay, capacitor. 01 microFarad na saída do delay, diodos ligados ao trim-pot de delay.
	cor fraca	modulador de RF, saída de crominância da TIA, cabo de antena
ÁUDIO	não há som	saídas de áudio da TIA, circuito oscilador de áudio (4,5MHZ) ajuste do modulador de RF

DEFEITO	SINTOMAS	CAUSAS POSSÍVEIS
	áudio fraco	TIA, circuito oscilador de áudio, ajuste do modulador de RF.
ATUADOR	botão de TIRO não funciona	conector do joystick ; joystick.
OUTROS	chaves inoperantes, exceto POWER	PIA, capacitores de .001 microFarad ligados às chaves; chaves de controles

CARTUCHO TESTE - a POLIVOX (GRADIENTE) fornece também um cartucho-teste com instruções para uso, destinado tanto a serviços de conserto quanto a ajustes e calibragem.

4 - MANUTENÇÃO DE ODISSEY

O ODISSEY utiliza um grande número de CIs, o que torna impraticável o processo "troca-peça". A maior complexidade do ODISSEY também leva ao uso de instrumentos digitais especializados, como o COMPUTESTER. Suporemos que a fonte e todo o sistema de alimentação foi devidamente checada.

TESTES INICIAIS/SUBSTITUIÇÃO DO MICROPROCESSADOR

Inicialmente é necessário que no microprocessador estejam em nível 0 os pinos 7 (EA), 20 (Vss); em nível 1 os pinos 4 (RESET), 5 (SS), 6 (INT), 40 (Vcc), 26 (Vdd). Se tudo estiver OK substitua o microprocessador. Lembre-se que é necessário um 8048 gravado com

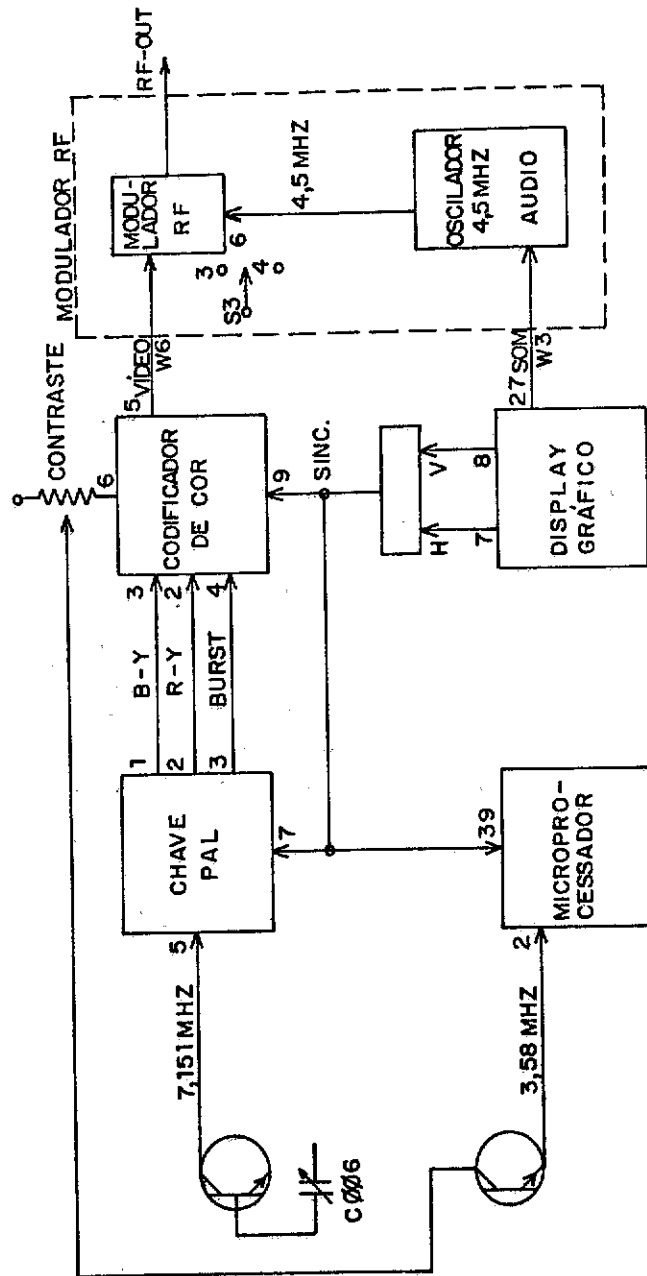


figura 11-5 - Pontos para teste de sinal no ODISSEY

o programa ODISSEY na sua ROM interna.

ACOMPANHAMENTO DE SINAL COM O OSCILOSCÓPIO

Na figura 11-5 vemos os principais pontos para teste com osciloscópio. Na saída - RFOUT - deve haver o sinal composto de vídeo já modulado no canal 3 (60-66 MHz) ou canal 4 (66-72 MHz); a chave S3 dentro da blindagem do modulador permite selecionar um desses canais. A primeira entrada do modulador de RF é o sinal de 0-6 MHz de vídeo composto (com sincronismo e croma) no terminal W2; a segunda é o som no terminal W3. Este último entra com menos de 1 MHz, dependendo do cartucho, e modula os 4,5 MHz produzidos no oscilador de áudio dentro da blindagem, cuja saída deve aparecer no pino 6 do modulador de RF propriamente dito. Os mesmos sinais de vídeo e som serão encontrados nos pinos 5 do codificador de cor e 27 do display gráfico, respectivamente. Os sincronismos horizontal H (15.750 Hz) e vertical (60 Hz) são emitidos nos pinos 7 e 8 do display gráfico e, após combinados, formam o sinc de mesmas frequências (15.750 Hz e 60 Hz) que entram nos pinos 9 do codificador de cor, 39 do microprocessador e 7 da chave PAL. Esta emite nos pinos 1, 2 e 3 os sinais B-Y, R-Y e BURST, todos de 3,58 MHz (apenas com diferenças de fase) que entram nos pinos 3, 2 e 4 do codificador de cor (no qual existe um potenciômetro ligado ao pino 6 que ajusta o contraste da imagem). O sinal de clock de 7,151 MHz originado no circuito contendo o cristal aparece no coletor do transistor conectado ao pino 5 da chave PAL e é ajustável através do trimmer C006 (para fazer o ajuste é necessário um frequencímetro capaz de registrar a frequência de 7,151 MHz). O segundo transistor do circuito de clock apresenta no seu emissor a frequência 3,58 MHz que é enviada para o pino 2 (XTAL I) do microprocessador.

TESTE DE JOYSTICK

O microprocessador é retirado. Nos três pinos 21 (P20), 22 (P21), 23 (P22), 29 (P12) do soquete vazio é posto o sinal 0, verificando-se se esse nível chega no decodificador 74LS156 nos pinos 13 (P20), 3 (P21), 1 (P22), 2 e 14 (STROBE) e em consequência seu pino 9 é levado a 0, enquanto os pinos 10, 11, 12, 7 e 6 permanecem em nível 1. O mesmo nível 0 do decodificador deve dar entrada no pino 1 do buffer 74LS365, consistindo no primeiro sinal de habilitação do joystick R. Nível 0 no terminal 8 (READ) do microprocessador (seu soquete vazio) deve aparecer no pino 15 do buffer como o segundo sinal de habilitação. Girando o joystick nas 4 direções (R,L,U,D), para cada uma deve ocorrer nível 0 nas respectivas entradas no microprocessador (pinos 12-D0, 13-D1, 14-D2, 15-D3); apertando o botão de ação o nível 0 aparecerá no pino 16 (D4) do microprocessador. Observe que somente uma das 4 entradas pode estar em nível 0. Se o sinal esperado não aparece no pino do microprocessador teste a saída do buffer (pinos 13, 11, 9, 5, 7) e em seguida a entrada (pinos 14, 12, 10, 4, 6). Se a falha não foi localizada até esse ponto retire o joystick e gere o sinal 0 diretamente em cada terminal do conector (terminais 6, 5, 4, 3, 2), testando se ele chega ao microprocessador; refaça para nível 1. Se o resultado é positivo para sinais colocados diretamente no conector então a falha está no próprio joystick. Repita toda a operação para o joystick L; apenas o sinal para o decodificador 74LS156 deve ser P20=1, P21=0, P22=0 e P12=0.

TESTE DE TECLADO

P12 (pino 29) no soquete vazio do microprocessador é mantido em 0, nível este que deve aparecer nos

pinos 1 e 15 do decodificador 74LS156. Ainda no soquete P22, P21 e P20 formam o número binário B = bit 2-bit 1 - bit 0, respectivamente. Fazendo B = 000 (ou seja, P22=P21=P20=0) a linha 0 do decodificador (pino 9) deve ser levada a 0, enquanto as outras permanecem em 1. Para B = 001 será a linha 1 (pino 10), B=010 a linha 2 (pino 11), B = 011 a linha 3 (pino 12), B = 100 a linha 4 (pino 7), B = 101 a linha 5 (pino 6). Para cada valor de B verifique se a correspondente linha do decodificador é levada a 0, enquanto as outras permanecem em 1, esses níveis chegam aos terminais de entrada do conector de teclado (terminais 1 a 6); ainda para cada valor de B e sem nenhuma tecla apertada teste se todos os terminais de saída do teclado (terminais 7 a 14) permanecem em nível 1, e se esses níveis chegam aos correspondentes pinos no codificador 74LS148 (pinos 1 a 4 e 10 a 13). Levando a 0 a linha 1 do decodificador (portanto B=000) aperte as 8 teclas correspondentes na matriz do teclado, uma por vez, e a cada uma teste se o respectivo terminal na saída do conector (pinos 7 a 14) é levado a 0 e esse nível chega aos pinos de entrada do codificador. No codificador temos 8 entradas (pinos 1 a 4 e 10 a 13); os três pinos de saída (9, 7, 6) formam um número binário BS= bit 2 bit 1 bit 0 (pinos 6, 7 e 9) cujo valor é igual ao número da entrada levada a nível 0. Mantenha o pino 14 (P24, CS) em nível 1, habilitando o codificador; coloque nível 0 na entrada número 1 (pino 10) e teste se BS=000; mude o nível 0 para a entrada 1 (pino 11) e teste se BS=001; nível 0 na entrada 2 (pino 12) e BS=010; nível 0 na entrada 3 (pino 13) e BS=011; nível 0 na entrada 4 (pino 1) e BS = 100; nível 0 na entrada 5 (pino 2) e BS=101; nível 0 na entrada 6 (pino 3) e BS=110; nível 0 na entrada 7 (pino 4) e BS=111. Para cada uma delas veja se BS, saindo nos pinos 9, 7 e 6 do codificador chega aos pinos 36 (P25), 37 (P26) e 38 (P27) do microprocessador. Aperte a tecla RESET e ve

verifique se ocorre nível 0 no pino 4 (RESET) do microprocessador.

TESTE DA RAM

O endereçamento da RAM (A0-A7, pinos 17 a 23) mais CS5-A7, pino 15, é retirado dos latches 74LS175, que devem ser testados antes. Retire o microprocessador e coloque nível 1 nos pinos 12 a 19 (B0 a B7) do soquete vazio. Assente também nível 1 no pino 11 (ALE) e veja se ele chega aos pinos 9 dos dois latches como nível 0 (há inversão do sinal na porta NAND). O byte 1111 1111 assentado no soquete do microprocessador deve aparecer nas entradas dos latches (pinos 4,5,12,13) e nas saídas (pinos 2, 7, 10,15). Altere o sinal no pino 12 (B0) do soquete, de 1 para 0 e depois de 0 para 1, verificando se essa mesma alteração ocorre na entrada (pino 12) e saída (pino 10) do latch; repita para todos os terminais de B0 a B7. Para cada teste B0 a B7 verifique se o mesmo nível saindo do latch chega ao correspondente pino de endereço da ROM (pinos 17 a 23) mais A7-CS5 no pino 15. Coloque o byte 0000 0000 nos pinos 12 a 19 (B0 a B7) do soquete e faça ALE nível 1; veja se esse byte aparece na saída dos latches e nos pinos de endereço mais CS5 da RAM; leve ALE a nível 0, mude o byte no soquete para 1111 1111 e verifique se o byte anterior (0000 0000) ainda continua na saída dos latches e nos pinos da RAM; veja se o novo byte (1111 1111) está aparecendo nos terminais de dados da RAM (pinos 2 a 9); coloque nível 0 no pino 31 (P14) e nível 1 no pino 33 (P16) do soquete; faça pino 8 (READ) em nível 1 e pino 10 (WRITE) nível 0, no soquete; verifique se na RAM o pino 11 (CS1) está em 0, pino 16 (CS3) em 1 e pino 13 (W/R) em 0. Se tudo está correto o byte 1111 1111 foi escrito no endereço 0000 0000 da RAM. Para confirmar faça a leitura: retire o byte 1111 1111

do soquete, leve o pino 10 (WRITE) a 1 e o pino 8 (READ) a 0; no pino 13 (W/R) da RAM deve aparecer sinal 1; verifique se o byte 1111 1111 também aparece nos terminais de dados da RAM (pinos 2 a 9) e se ele chega até os pinos do soquete do microprocessador. Repita toda a operação escrevendo e depois lendo o byte 0000 0000 no endereço 0111 1111 da RAM (note que A7 deve continuar em 0 para habilitar CS5).

TESTE DE CARTUCHO

Tire o circuito impresso com a ROM do cartucho e encaixe-o no console, sem as duas tampas. Faça ALE nível 1 e coloque o byte 0000 0000 nos pinos de dados (B0-B7) do soquete vazio do microprocessador, verificando se ele aparece na saída dos latches (pinos 2,7,10,15) e nos pinos de endereço da ROM. Repita para o byte 1111 1111. No soquete coloque em nível 1 os pinos 21 (P20), 22 (P21), 23 (P22), 24 (P23), 27 (P10) e 28 (P11) e comprove se cada sinal chega até a ROM; altere o nível para 0 e refaça o teste. Coloque em nível 1 os pinos 31 (P14) e 33 (P16) do soquete e verifique se os pinos correspondentes da ROM recebem esse nível. Leve a 0 o pino 10 do soquete (PSEN) e teste se o pino correspondente na ROM também vai a 0. Em todos os testes acima verifique se o nível na junção conector-placa de circuito é o mesmo dos respectivos pinos na ROM.

CARTUCHO-TESTE

A PHILIPS oferece um cartucho-teste para o ODISSEY, com as seguintes indicações de defeitos:

ESTÁCIO DEFEITUOSO SINTOMA(USANDO O CARTUCHO TESTE)

Codificador de cor - tela branca, sem chuvaisco
 - tela escura, aparece a pala -
 vra SELECT GAME com ou sem cor
 - não faz demais funções
 - perde uma das cores
 - leve chuvaisco e nos terminais
 2,3 e 4 do codificador não apa
 rece 2,8 a 3 Volts.

Modulador RF - sem vídeo (tela branca), sem
 chuvaisco, com ou sem som.
 - com chuvaisco, sem vídeo, sem
 som.

Cristal - tela branca, leve chuvaisco,sem
 som.

RAM - não faz a função da tecla 1
 - função tecla 2 aparece OVERLAP
 BAD
 - função tecla 3 perde a cor,cai
 o sincronismo horizontal, no
 pino 1 do CI 3 fica zero Volt
 - função tecla 9 aparece a pala-
 vra RAM BAD.

APÊNDICE A – ASSEMBLER/CÓDIGO DE MÁQUINA DO 6507

O microprocessador 6507 é compatível com o 6502 (usado nos micros APPLE), com algumas restrições. A principal é que o 6507 não tem terminal de interrupção (INT), não possuindo portanto as instruções do 6502 que se refiram a ele. Apresentaremos a lista completa do 6502, devendo o leitor subtrair as instruções não aplicáveis ao 6507. Iniciaremos com uma explicação dos 9 modos de endereçamento do 6502/6507.

MODOS DE ENDEREÇAMENTO DO 6502/6507

- (1) ACUMULADOR - a instrução refere-se a uma operação a ser executada no acumulador, tendo apenas 1 byte.
- (2) IMPLÍCITO - não é necessário especificar endereço na instrução
- (3) IMEDIATO - o 2º byte da instrução contem o dado sem necessidade de especificar endereço
- (4) RELATIVO - o 2º byte da instrução é somado ao PC (contador de programa) para formar o novo endereço.

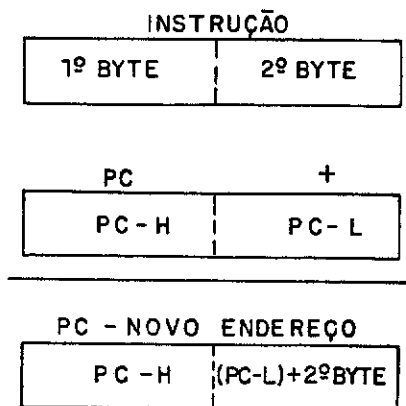


figura A-1 - Endereçamento relativo

- (5) PÁGINA ZERO - o byte superior do novo endereço é subentendido como 00H e o byte inferior é o 2º byte da instrução, resultando um endereçamento das posições de 0000H até 00FFH (de 0 a 256, em decimal) que é conhecido por "página zero".

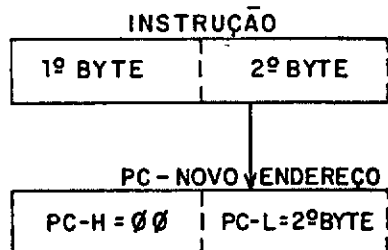


figura A-2 - Endereçamento página zero

- (6) ABSOLUTO - o 2º e o 3º bytes da instrução formam os bytes inferior e superior, respectivamente, do novo endereço.

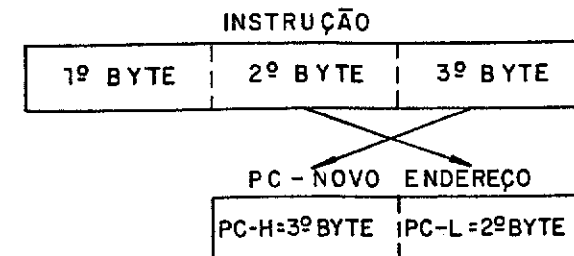


figura A-3 - Endereçamento absoluto

- (7) INDEXADO ABSOLUTO - o 2º e o 3º bytes da instrução são somados aos registros X ou Y, e o resultado é o novo endereço.

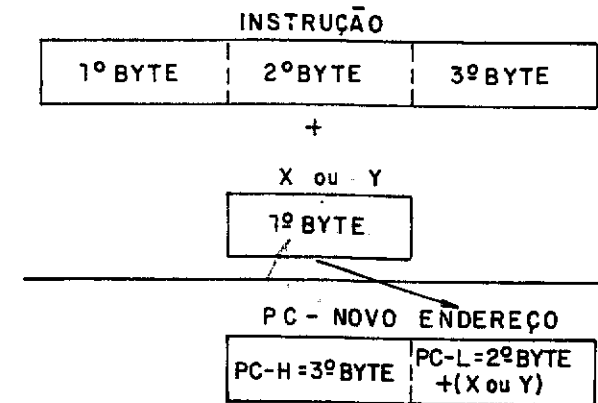


figura A-4 - Endereçamento indexado absoluto

- (8) INDEXADO PÁGINA ZERO - o 2º byte da instrução é somado aos registros X ou Y, e o resultado é o novo endereço na página zero (byte superior = 00H).

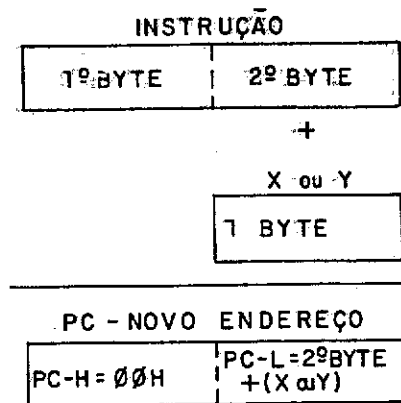


figura A-5 - Endereçamento indexado página zero

- (9) INDIRETO ABSOLUTO - o 2º e o 3º bytes da instrução são carregados nos bytes inferior e superior, respectivamente, do PC, que passa a indicar a posição de memória onde está o byte inferior do novo endereço; o byte superior deste estará na próxima posição.
- (10) INDIRETO INDEXADO X - o 2º byte da instrução é somado ao registro X; o resultado aponta uma posição na página zero, cujo conteúdo é o byte inferior do novo endereço; o byte superior é o conteúdo da próxima posição.

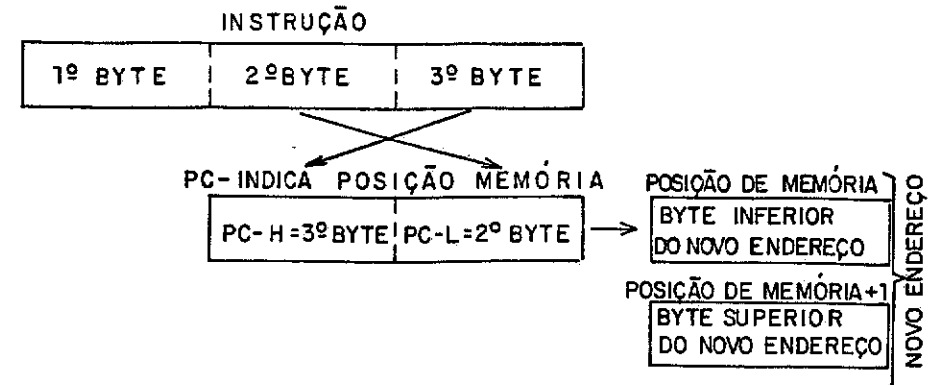


figura A-6 - Endereçamento indireto absoluto

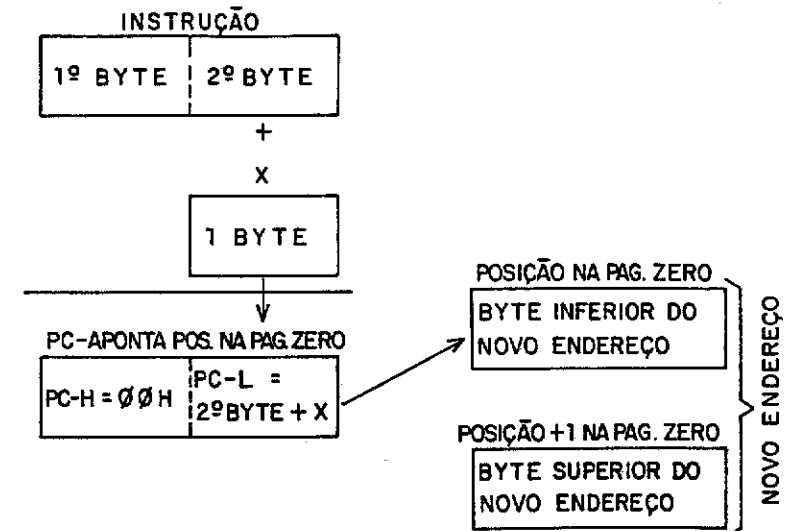


figura A-7 - Endereçamento indireto indexado X

- (11) INDEXADO INDIRETO Y - o 2º byte da instrução a ponta uma posição na página zero cujo conteúdo juntamente com o da próxima posição, formam os bytes inferior e superior, respectivamente, de um valor que sonado ao registro Y constitui o novo endereço.

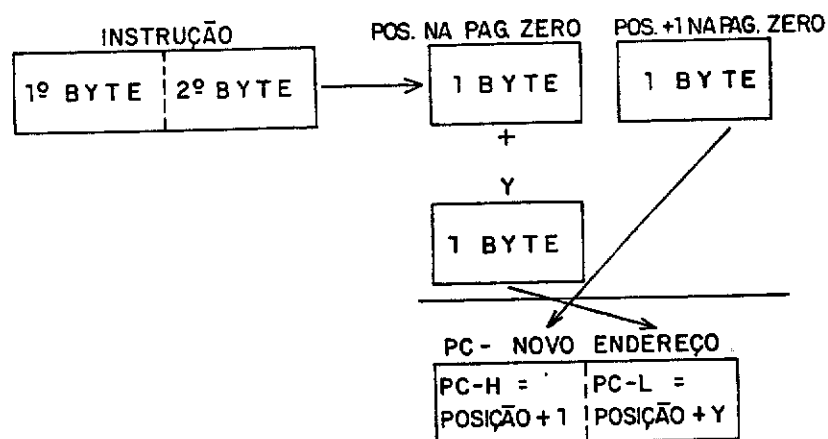


figura A-8 - Endereçamento indexado indireto Y

LISTA DE INSTRUÇÕES DO 6502/6507

MNE- MÔNICO	TIPO DE ENDEREÇAMENTO	CÓDIGO	DESCRIÇÃO
ADC	imediato	0110 1001	adiciona com carry
	absoluto	0110 1101	
	pág.zero	0110 0101	
	ind.ind.X	0110 0001	
	ind.ind.Y	0111 0001	
	pág.zero	0111 0101	

MNE- MÔNICO	TIPO DE ENDEREÇAMENTO	CÓDIGO	DESCRIÇÃO
	ind.absol. X	0111 1101	
	ind.absol. Y	0111 1001	
AND	imediato	0010 1001	acumulador AND dado
	absoluto	0010 1101	
	pág.zero	0010 0101	
	ind.ind. X	0010 0001	
	ind.ind. Y	0011 0001	
	ind.pág.z.X	0011 0101	
	ind.absol.X	0011 1101	
	ind.absol.Y	0011 1001	
ASL	acumulador	0000 1010	todos os bits do da
	pág. zero	0000 0110	do deslocam 1 casa
	absoluto	0000 1110	para a esquerda; o
	ind.p.z. X	0001 0110	MSB é carregado na
	ind.absol.X	0001 1110	flag C e o LSB = 0.
BCC	relativo	1001 0000	desvia se flag C=0.
BCS	relativo	1011 0000	desvia-se flag C=1.
BEQ	relativo	1111 0000	desvia se flag Z=1.
BIT	pág.zero	0010 0100	acumulador AND dado
	absoluto	0010 1100	
BMI	relativo	0011 0000	desvia se flag S=1.
BNE	relativo	1101 0000	desvia se flag Z=0.
BPL	relativo	0001 0000	desvia se flag S=0.
BRK	implícito	0000 0000	para programa
BVC	relativo	0101 0000	desvia se flag V=0.
BVS	relativo	0111 0000	desvia se flag V=1.
CLC	implícito	0001 1000	reseta flag C
CLD	implícito	1101 1000	reseta flag D
CLI	implícito	0101 1000	reseta flag I
CLV	implícito	1001 1000	reseta flag V
CMP	imediato	1100 1001	compara acumulador e dado

<u>MNE- MÔNICO</u>	<u>TIPO DE ENDEREÇAMENTO</u>	<u>CÓDIGO</u>	<u>DESCRIÇÃO</u>
	absoluto	1100 1101	
	pág.zero	1100 0101	
	ind.ind. X	1100 0001	
	ind.ind. Y	1101 0001	
	ind.p.z. X	1101 0101	
	ind.absol.X	1101 1101	
	ind.absol.Y	1101 1001	
CPX	imediato	1110 0000	compara registro X e dado
	absoluto	1110 1100	
	pág.zero	1110 0100	
CPY	imediato	1100 0000	compara registro Y e dado
	absoluto	1100 1100	
	pág.zero	1100 0100	
DEC	absoluto	1100 1110	decrementa o dado de -1
	pág.zero	1100 0110	
	ind.p.z. X	1101 0110	
	ind.absol.X	1101 1110	
DEX	implícito	1100 1010	decrementa x de -1
DEY	implícito	1000 1000	decrementa Y de -1
EOR	imediato	0100 1001	acumulador EOR da- do
	absoluto	0100 1101	
	pág.zero	0100 0101	
	ind.ind.X	0100 0001	
	ind.ind.Y	0101 0001	
	ind.p.z.X	0101 0101	
	ind.absol.X	0101 1101	
	ind.absol.Y	0101 1001	

<u>MNE- MÔNICO</u>	<u>TIPO DE ENDEREÇAMENTO</u>	<u>CÓDIGO</u>	<u>DESCRIÇÃO</u>
INC	absoluto	1110 1110	incrementa dado de +1
	pág.zero	1110 0110	
	ind.p.z.X	1111 0110	
	ind.absol.X	1111 1110	
INX	implícito	1110 1000	incrementa X de +1
INY	implícito	1100 1000	incrementa Y de +1
JMP	absoluto	0100 1100	pula para novo en- dereço
	indireto	0110 1100	
JSR	absoluto	0010 0000	pula para sub-roti- na
LDA	imediato	1010 1001	carrega dado no a- cumulador
	absoluto	1010 1101	
	pág.zero	1010 0101	
	ind.ind.X	1010 0001	
	ind.ind.Y	1011 0001	
	ind.p.z.X	1011 0101	
	ind.absol.X	1011 1101	
	ind.absol.Y	1011 1001	
LDX	imediato	1010 0010	carrega dado no re- gistro X
	absoluto	1010 1110	
	pág.zero	1010 0110	
	ind.p.z.X	1011 0110	
	ind.absol.Y	1011 1110	
LDY	imediato	1010 0000	carrega dado no re- gistro Y
	absoluto	1010 1100	

<u>MNE- MÔNICO</u>	<u>TIPO DE ENDEREÇAMENTO</u>	<u>CÓDIGO</u>	<u>DESCRIÇÃO</u>
	pág.zero	1010 0100	
	ind.p.z.X	1011 0100	
	ind.absol.X	1011 1100	
LSR	acumulador	0100 1010	desloca todos os
	absoluto	0100 1110	bits do dado para
	pág.zero	0100 0110	a direita; o MSB
	ind.p.z.X	0101 0110	é mantido
	ind.absol.X	0101 1110	
NOP	implícito	1110 1010	nenhuma operação
ORA	imediato	0000 1001	acumulador OR da-
			do
	absoluto	0000 1101	
	pág.zero	0000 0101	
	ind.ind.X	0000 0001	
	ind.ind.Y	0001 0001	
	ind.p.z.X	0001 0101	
	ind.absol.X	0001 1101	
	ind.absol.Y	0001 1001	
PHA	implícito	0100 1001	armazena acumula-
			dor na pilha
PHP	implícito	0000 1000	armazena registro
			P na pilha
PLA	implícito	0110 1000	resgata acumula -
			dor da pilha
PLP	implicito	0010 1000	resgata registro
			P da pilha
ROL	acumulador	0010 1010	desloca todos os
	absoluto	0010 1110	bits do dado para
	pág.zero	0010 0110	a esquerda; o MSB
	ind.p.z.X	0011 0110	é carregado na
	ind.absol.X	0011 1110	flag C e esta no
			LSB

<u>MNE- MÔNICO</u>	<u>TIPO DE ENDEREÇAMENTO</u>	<u>CÓDIGO</u>	<u>DESCRIÇÃO</u>
ROR	acumulador	0110 1010	desloca todos os
	absoluto	0110 1110	bits do dado para
	pág.zero	0110 0110	a direita; o LSB
	ind.p.z.X	0111 0110	é carregado na
	ind.absol.X	0111 1110	flag C e esta no
			MSB
RTI	implícito	0100 0000	retorna de inter-
			rupção
RTS	implícito	0110 0000	retorna de sub-
			rotina
SBC	imediato	1110 1001	subtrai dado e
	absoluto	1110 1101	carry do acumula-
	pág.zero	1110 0101	dor
	ind.ind.X	1110 0001	
	ind.ind.Y	1111 0001	
	ind.p.z.X	1111 0101	
	ind.absol.X	1111 1101	
	ind.absol.Y	1111 1001	
SEC	implícito	0011 1000	seta flag C
SED	implícito	1111 1000	seta flag D
SEI	implícito	0111 1000	seta flag I
STA	absoluto	1000 1101	carrega acumula -
	pág.zero	1000 0101	dor no endereço
	ind.ind.X	1000 0001	
	ind.ind.Y	1001 0001	
	ind.p.z.X	1001 0101	
	ind.absol.X	1001 1101	
	ind.absol.Y	1001 1001	
STX	absoluto	1000 1110	carrega registro
			X no endereço
	pág.zero	1000 0110	
	ind.p.z.Y	1001 0110	

<u>MNE- MÔNICO</u>	<u>TIPO DE ENDEREÇAMENTO</u>	<u>CÓDIGO</u>	<u>DESCRIÇÃO</u>
STY	absoluto	1000 1100	carrega registro
	pág. zero	1000 0100	Y no endereço
	ind.p.z.X	1001 0100	
TAX	implícito	1010 1010	transfere acumu- lador para regis- tro X
TAY	implícito	1010 1000	transfere acumu- lador para regis- tro Y
TSX	implícito	1011 1010	transfere regis- tro S para regis- tro X
TXA	implícito	1000 1010	transfere regis- tro X para acumu- lador
TXS	implícito	1001 1010	transfere regis- tro X para regis- tro S
TYA	implícito	1001 1000	transfere regis- tro Y para acumu- lador.

APÊNDICE B – ASSEMBLER/CÓDIGO DE MÁQUINA 8048

<u>MNEMÔNICO/ CÓDIGO</u>	<u>FUNÇÃO / DESCRIÇÃO</u>
- ADD A, Rr 0111 01rrr	soma conteúdo do registro Rr com acumulador
- ADD A, @Rr 0110 000r	soma conteúdo do endereço indicado por Rr com acumulador
- ADD A, #data 0000 0011 XXXX XXXX	soma byte XXXX XXXX com acumulador
- ADDC A, Rr 0111 1rrr	soma carry mais conteúdo de Rr com acumulador
- ADDC A, @Rr 0111 000r	soma carry mais conteúdo do endere- ço indicado por Rr com acumulador
- ADDC A, #data 0001 0011 XXXX XXXX	soma carry mais byte XXXX XXXX com acumulador
- ANL A, Rr 0101 1rrr	conteúdo de Rr AND acumulador
- ANL A, @Rr 0101 000r	(conteúdo do endereço indicado por Rr) AND acumulador

MNMÔNICO/ CÓDIGO	FUNÇÃO / DESCRIÇÃO
- ANL A,## data XXXX XXXX	byte XXXX XXXX AND acumulador
- CLR A 0010 0111	zera acumulador
- CPL A 0011 0111	complementa acumulador
- DA A 0101 0111	ajuste decimal do acumulador
- DEC A 0000 0111	decrementa -1 o acumulador
- INC A 0001 0111	incrementa +1 o acumulador
- ORL A, Rr 0100 1rrr	conteúdo de Rr OR acumulador
- ORL A,@ Rr 0100 000r	(conteúdo do endereço indicado por Rr) OR acumulador
- ORL A,## data 0100 0011 XXXX XXXX	byte XXXX XXXX OR acumulador
- RLA 1110 0111	roda acumulador 1 bit para a esquerda; bit 7 vai para bit 0
- RLCA 1111 0111	roda acumulador 1 bit para a esquerda; bit 7 vai para o carry e este para bit 0
- RRA 0111 0111	roda acumulador 1 bit para a direita; bit 0 vai para bit 7

MNMÔNICO/ CÓDIGO	FUNÇÃO / DESCRIÇÃO
- RRC A 0110 0111	roda acumulador 1 bit para direita; bit 0 vai para o carry e este para bit 7
- SWAP A 0100 0111	no acumulador bit 0 a bit 3 trocam de posição com bit 4 a bit 7
- XRLA, Rr 1101 1rrr	conteúdo de Rr XOR acumulador
- XRL A,@ Rr 1101 000r	(conteúdo do endereço indicado por Rr) XOR acumulador
- XRLA,## data 1101 0011 XXXX XXXX	byte XXXX XXXX XOR acumulador
- DJNZ Rr,addr 1110 1rrr XXXX XXXX	decrementa conteúdo de Rr; se o resultado não for zero salta para endereço XXXX XXXX
- JBB addr bbb1 0010 XXXX XXXX	salta para endereço XXXX XXXX se o bit bbb (valor binário convertido em decimal) do acumulador estiver setado
- JC addr 1111 0110 XXXX XXXX	salta para endereço XXXX XXXX se o carry está setado
- JFO addr 1011 0110 XXXX XXXX	salta para endereço XXXX XXXX se flag FO está setada
- JF1 addr 0111 0110 XXXX XXXX	salta para endereço XXXX XXXX se flag F1 está setada

MNEMÔNICO/ CÓDIGO	FUNÇÃO / DESCRIÇÃO
- JMP addr MMMO 0100 XXXX XXXX	salta para endereço OMMM XXXX XXXX se flag MB estiver resetada ou 1MMM XXXX XXXX se estiver setada
- JMPP@A 1011 0011	carrega byte inferior do PC com conteúdo do acumulador (salta den- tro da página)
- JNC addr 1110 0110 XXXX XXXX	salta para endereço XXXX XXXX se o carry está resetado
- JNI 1000 0110 XXXX XXXX	salta para endereço XXXX XXXX se flag INT está resetada
- JNT0 addr 0010 0110 XXXX XXXX	salta para endereço XXXX XXXX se terminal T0 está resetada
- JNT1 addr 0100 0110 XXXX XXXX	salta para endereço XXXX XXXX se terminal T1 está resetado
- JNZ addr 1001 0110 XXXX XXXX	salta para endereço XXXX XXXX se acumulador não está zerado
- JTF addr 0001 0110 XXXX XXXX	salta para endereço XXXX XXXX se TIMER (TF) está setado
- JTO addr 0011 0110 XXXX XXXX	salta para endereço XXXX XXXX se terminal T0 está setado

MNEMÔNICO/ CÓDIGO	FUNÇÃO / DESCRIÇÃO
- JTI addr 0101 0110 XXXX XXXX	salta para endereço XXXX XXXX se terminal T1 está setado
- JZ addr 1100 0110 XXXX XXXX	salta para endereço XXXX XXXX se acumulador está zerado
- ENI 0000 0101	habilita interrupção externa (INT)
- DIS I 0001 0101	desabilita interrupção externa (INT)
- SEL RBO 1100 0101	reseta flag BS (seleciona banco de registro de 0 a 7 na RAM interna)
- SEL RB1 1101 0101	seta flag BS (seleciona banco de registro de 24 a 31 na RAM inter- na)
- SEL MBO 1110 0101	reseta flag MB (seleciona endere- ços de 0 a 2K na memória)
- SEL MB1 1111 0101	seta flag MB (seleciona endereços de 2K a 4K na memória)
- ENTO CLK 0111 0101	habilita saída de clock no termi- nal T0
- MOV A, #data 0010 0011 XXXX XXXX	carrega byte XXXX XXXX no acumula- dor
- MOV A, Rr 1111 1rrr	copia conteúdo de Rr no acumulador

MNEMÔNICO/ CÓDIGO	FUNÇÃO / DESCRIÇÃO
- MOV A,@ Rr 1111 000r	copia no acumulador conteúdo do endereço indicado por Rr
- MOV A, PSW 1100 0111	copia no acumulador conteúdo do registro de flags PSW
- MOV Rr,# data 1011 lrrr XXXX XXXX	carrega byte XXXX XXXX no registro Rr
- MOV Rr,A 1010 lrrr	copia acumulador no registro Rr
- MOV @ Rr,A 1010 000r	copia acumulador no endereço indicado por Rr.
- MOV@ Rr,# data 1011 000r XXXX XXXX	carrega byte XXXX XXXX no endereço indicado por Rr
- MOV PSW, A 1101 0111	copia acumulador no registro de flags PSW
- MOVP A,@ A 1010 0011	carrega acumulador no byte inferior do PC, e o conteúdo do endereço apontado por ele é copiado no acumulador
- MOVP3 A,@ A 1110 0011	carrega 0010 no byte superior do PC (página 3) e o acumulador no byte inferior; o conteúdo do endereço apontado por ele é copiado no acumulador
- MOVX A,@ Rr 1000 000r	copia no acumulador o conteúdo do endereço na memória externa apontado por Rr

MNEMÔNICO/ CÓDIGO	FUNÇÃO / DESCRIÇÃO
- MOVX @ Rr,A 1001 000r	copia o acumulador no endereço na memória externa apontado por Rr.
- XCH A, Rr 0010 lrrr	troca os conteúdos do acumulador e registro Rr
- XCH A,@ Rr 0010 000r	troca o conteúdo do acumulador com o conteúdo do endereço indicado por Rr
- XCHD A,@ Rr 0011 000r	troca os 4 bits inferiores do acumulador com os 4 bits inferiores do conteúdo do endereço indicado por Rr
- CPLC 1010 0111	complementa (inverte) flag carry
- CPL FO 1001 0101	complementa flag FO
- CPL F1 1011 0101	complementa flag F1
- CLR C 1001 0111	reseta flag carry
- CLR FO 1000 0101	reseta flag FO
- CLR F1 1010 0101	reseta flag F1
- ANL BUS,# data 1001 1000 XXXX XXXX	(barra de dados - B0 - B7) AND byte XXXX XXXX; o resultado permanece na barra de dados

MNEMÔNICO/ CÓDIGO	FUNÇÃO / DESCRIÇÃO
- ANL Pp, #data 1001 10pp XXXX XXXX	(porta p) AND byte XXXX XXXX; o resultado permanece na porta
- ANLD Pp, A 1001 1lpp	(bits 4 a 7 da porta p)AND (bits 0 a 3 do acumulador); o resultado permanece na porta
- IN A, Pp 0000 10pp	copia porta Pp no acumulador
- INS A,BUS 0000 1000	copia BUS (barras de dados B0 - B7) no acumulador
- MOVD A,Pp 0000 1lpp	copia os bits 4 a 7 da porta Pp nos bits 0 a 3 do acumulador cujos bits 4-7 são zerados
- MOVD Pp, A 0011 1lpp	bits 0 a 3 do acumulador são carregados nos bits 4 a 7 da porta Pp
- ORLD Pp, A 1000 1lpp	(bits 0 a 3 do acumulador) OR (bits 4 a 7 da porta Pp); o resultado permanece na porta
- ORL BUS,#data 1000 1000 XXXX XXXX	(byte XXXX XXXX) OR (BUS - B0 - B7); o resultado permanece no BUS
- ORL Pp, #data 1000 10pp XXXX XXXX	(byte XXXX XXXX) OR (porta Pp);o resultado permanece na porta
- OUT L BUS, A 0000 0010	carrega acumulador no BUS(B0-B7)

MNEMÔNICO/ CÓDIGO	FUNÇÃO / DESCRIÇÃO
- OUTL Pp, A 0011 10pp	carrega acumulador na porta Pp
- DEC Rr 1100 1rrr	decrementa -1 o registro Rr
- INC Rr 0001 1rrr	incrementa +1 o registro Rr
- INC @ Rr 0001 000r	incrementa +1 o conteúdo do endereço indicado por Rr
- CALL addr MMM1 0100 XXXX XXXX	o conteúdo do PC e do PSW (flags) é armazenado no SP (ponteiro de pilha na RAM interna); o PC é carregado com o endereço MMM XXXX XXXX
- RET 1000 0011	o conteúdo da posição (na RAM interna) indicada por SP é carregado no PC
- RETR 1001 0011	o conteúdo da posição (na RAM interna) indicada por SP é carregado no PC e no PSW
- EN TCNTI 0010 0101	seta flag TCNTI
- DIS TCNTI 0011 0101	reseta flag TCNTI
- MOV A, T 0100 0010	copia no acumulador o conteúdo do TIMER/COUNTER

MEMÔNICO/ CÓDIGO	FUNÇÃO / DESCRIÇÃO
- MOV T, A 0110 0010	copia acumulador no TIMER/COUN - TER
- STOP TCNT 0110 0101	pára contagem no TIMER/COUNTER
- STRT CNT 0100 0101	inicia contagem de interrupção no TIMER/COUNTER
- STRT T 0101 0101	inicia contagem de tempo no TI- MER/COUNTER
- NOP 0000 0000	não executa nenhuma operação

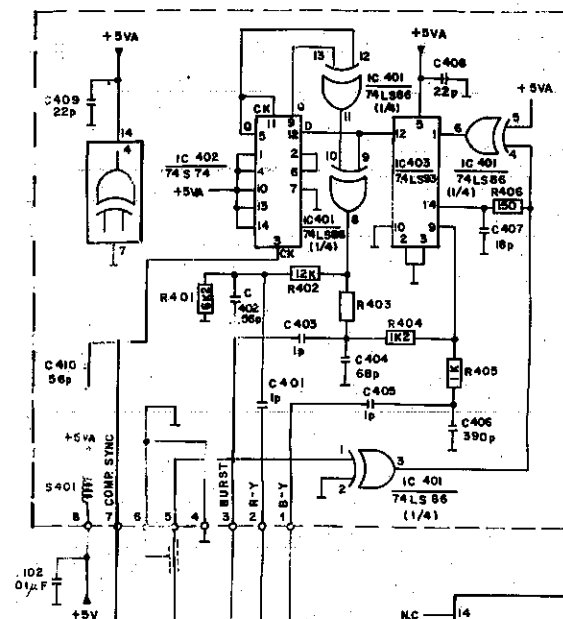
SE NUNCA SOFRESTE

Se nunca sofreste
 Lutando por alguma causa justa;
 Se nunca erraste,
 Atravessando a dor do desengano;
 Se nunca viste a incompreensão, em tor-
 no da ti,
 Aprendendo a entender os incompreen-
 didos;
 Se nunca choraste
 Reconhecendo quanto doem as lágrimas
 alheias ...
 Decerto ainda dormes,
 No berçário de amor da Natureza,
 E ninguém consegue, por enquanto,
 Predizer a hora em que despertarás
 Para iniciar a viagem
 Na direção da Luz Imperecível.

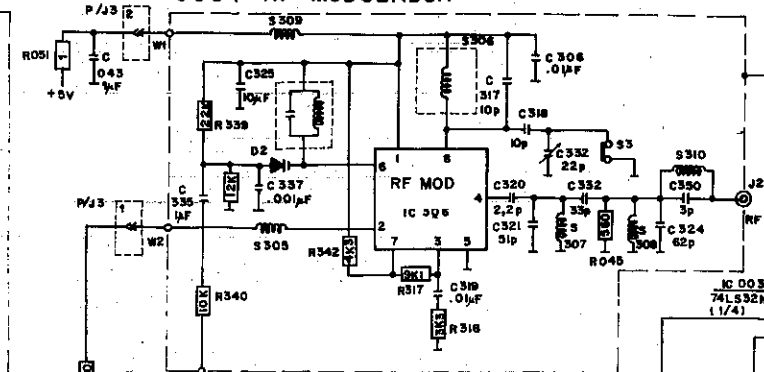
EMMANUEL

(Página recebida pelo médium FRANCISCO
 CÂNDIDO XAVIER, no Grupo Espírita de
 Prece, em reunião da noite de 21/janeiro/82
 em Uberaba, Minas).
 Oferta do Grupo Espírita "Os Mensageiros".

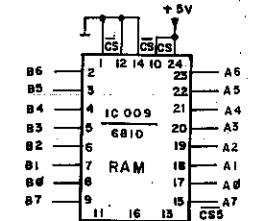
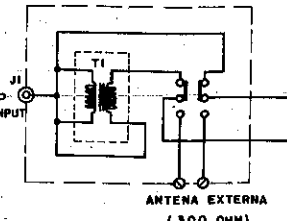
U002-CHAVE PAL-M



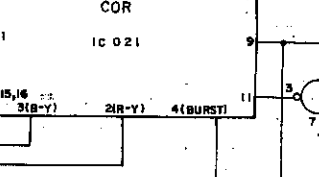
U004-RF MODULADOR



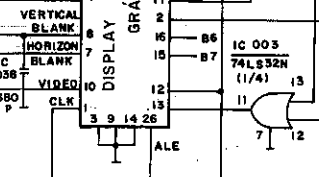
UI04-BOX DE ANTENA



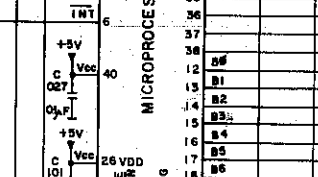
CODIFICADOR DE COR



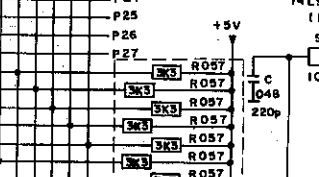
COMB SYNC BURST GATE



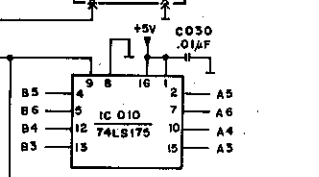
MICROPROCESSADOR



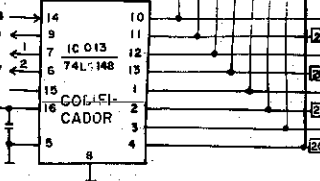
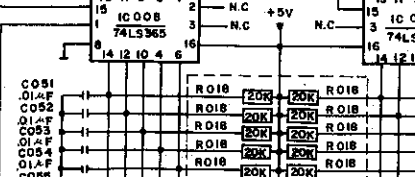
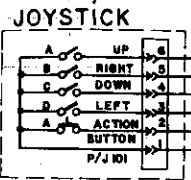
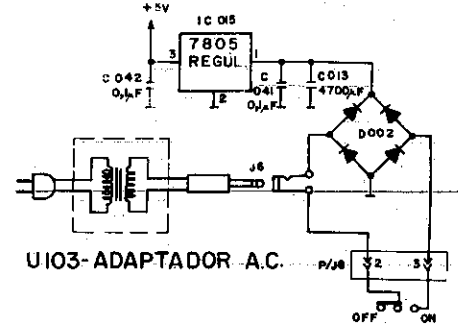
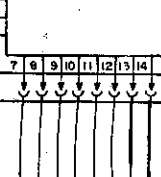
CONECTOR PARA CARTUCHO



DECODIFICADOR



KEYBOARD



ODISSEY-PETIT EDITORA