

UNIVERSIDADE DE SAO PAULO
INSTITUTO DE FISICA E QUIMICA DE SAO CARLOS

PROJETO E CONSTRUÇÃO
DE UM TERMINAL DE VIDEO
GRAFICO

CARLOS ANTONIO RUGGIERO

Dissertação apresentada no Instituto de Física
e Química de São Carlos, USP, para a obtenção
do título de Mestre em Física Aplicada.

Orientador: Dr. Jan F. W. Slaets



Departamento de Física e Ciência dos Materiais

1983

BIBLIOTECA DO INSTITUTO DE FISICA E QUIMICA DE SAO CARLOS - USP

MEMBROS DA COMISSÃO JULGADORA DA DISSERTAÇÃO DE MESTRADO DE _____
CARLOS ANTONIO RUGGIERO

APRESENTADA AO INSTITUTO DE FÍSICA E QUÍMICA DE SÃO CARLOS, DA
UNIVERSIDADE DE SÃO PAULO, EM 28 DE dezembro DE 1983.

COMISSÃO JULGADORA:



Dr. Jan F.W. Slaets - Orientador



Dr. Valentin Obac Roda



Dr. Arthur João Catto

BIBLIOTECA DO INSTITUTO DE FÍSICA E QUÍMICA DE SÃO CARLOS - USP
1111 A

A minha família

Patrocinadores: CNPq

FINEP

AGRADECIMENTOS

O autor agradece a todos aqueles que tornaram possível a realização deste trabalho. Em especial :

1. ao Prof. Jan Slaets, pela sua inestimável colaboração na elaboração desse trabalho e principalmente pela grande amizade.
2. ao pessoal do LIE, pelo constante incentivo e pelo enorme apoio técnico.

LISTA DE FIGURAS	i
RESUMO	iii
ABSTRACT	iv
INTRODUÇÃO	v

CAPITULO I INTRODUÇÃO AOS TERMINAIS DE COMPUTADORES

I.1	DEFINIÇÃO	1
I.2	ARQUITETURAS DE TERMINAIS DE SAÍDA	2
I.3	CARACTERES E GRÁFICOS	5
I.4	TERMINAIS APENAS DE SAÍDA	6
I.4.1	IMPRESSORA	6
I.4.2	'PLOTTER'	7
I.4.3	OUTROS	8
I.5	TERMINAIS DE VÍDEO	8
I.5.1	O TUBO DE RAIOS CATÓDICOS ('CRT')	9
I.5.2	TUBO DE ARMAZENAGEM E VISÃO DIRETA ('DVST')	15
I.5.3	TUBOS COLORIDOS	17
I.5.3.1	'CRT' COLORIDO POR PENETRAÇÃO DE FEIXE	17
I.5.3.2	'CRT' COLORIDO POR MÁSCARA DE SOMBRA	18
I.5.4	TERMINAL DE PLASMA	19
I.5.5	VÍDEOS DE DIÓDOS E CRISTAL LÍQUIDO	19
I.6	ARQUITETURAS DE VARREDURA ALEATORIA	20
I.7	VARREDURA ALEATORIA EM SISTEMAS COMERCIAIS	27
I.8	ARQUITETURAS DE VARREDURA POR RASTRO	27
I.9	COR E INTENSIDADE	31
I.10	APLICAÇÕES E PERSPECTIVAS	34

CAPITULO II	O PROJETO - ARQUITETURA	
II.1	ARQUITETURA GERAL	37
II.2	O TERMINAL ALFANUMERICO	40
II.3	TERMINAL GRAFICO	42

CAPITULO III	O PROJETO - "HARDWARE"	
III.1	INFORMACOES GERAIS	49
III.2	TERMINAL ALFANUMERICO	49
III.3	O TERMINAL GRAFICO	55
III.3.1	PRINCIPIOS BASICOS	55
III.3.2	DESCRICAO	58

CAPITULO IV	O PROJETO - "SOFTWARE"	
IV.1	INFORMACOES GERAIS	77
IV.2	TERMINAL ALFANUMERICO	77
IV.3	TERMINAL GRAFICO	83

CAPITULO V	RESULTADOS E CONCLUSAO	
V.1	GERAL	86
V.2	TESTES DE IMAGENS	86
V.3	TESTES COM RETAS	87
V.4	CONCLUSAO	88

CAPITULO VI	MELHORAMENTOS	
--------------------	----------------------	--

VI.1	CONSIDERAÇÕES GERAIS	89
VI.2	AUMENTO DE RESOLUÇÃO GRÁFICA	89
VI.2.1	AUMENTO DE RESOLUÇÃO COM MONITOR MAIS RÁPIDO . .	90
VI.2.2	AUMENTO DE RESOLUÇÃO GRÁFICA - ENTRELACAMENTO .	91
VI.3	AUMENTO DE VELOCIDADE DE ALTERAÇÃO DE TELA	92
VI.4	AUMENTO NO NÚMERO DE CORES E INTENSIDADE	94
VI.5	LEITURA DOS DADOS DA MEMÓRIA GRÁFICA	94
VI.6	MELHORIAS NO "SOFTWARE"	96

APENDICE I MANUTENÇÃO

I.1	CONSIDERAÇÕES GERAIS	97
I.2	TERMINAL ALFANUMÉRICO	97
I.2.1	DEFEITOS TOTAIS	97
I.2.2	DEFEITOS DE COMUNICAÇÃO	98
I.2.3	DEFEITOS DE TELA	100
I.3	TERMINAL GRÁFICO	102
I.3.1	DEFEITOS TOTAIS	102
I.3.2	DEFEITOS DE COMUNICAÇÃO	103
I.3.3	DEFEITOS DE TELA	104

REFERÊNCIAS BIBLIOGRÁFICAS	107
--------------------------------------	-----

LISTA DE FIGURAS

1 - Subsistemas de um computador.....	2
2 - Um desenho qualquer: um triângulo.....	3
3 - Triângulo desenhado por varredura aleatória.....	3
4 - Padrão "Raster Scan".....	4
5 - Triângulo desenhado por "Raster Scan".....	5
6 - Secção transversal de um tubo de raios catódicos.....	9
7 - Focalização no "CRT".....	11
8 - Deflexão magnética não compensada.....	12
9 - Linearidade.....	15
10 - Secção transversal do "DVST".....	16
11 - "CRT" colorido por máscara.....	18
12 - Terminal de plasma.....	20
13 - Traçado por ponto.....	21
14 - Traçado por reta.....	24
15 - Traçado analógico de retas.....	26
16 - Sistema de varredura por rastro.....	28
17 - Padrão para simular 5 níveis de intensidade.....	33
18 - Fluxo de dados no terminal completo.....	37
19 - Terminal apenas alfanumérico.....	38
20 - Arquitetura para sistemas com terminal incorporado.....	39
21 - Arquitetura do terminal alfanumérico.....	41
22 - Arquitetura do terminal gráfico.....	43
23 - O controlador gráfico.....	44
24 - Modos de escrita.....	48
25 - Matriz de decodificação de teclado.....	53
26 - Interface processador terminal gráfico.....	59

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE INFORMÁTICA
LABORATÓRIO DE SISTEMAS DE COMPUTAÇÃO

27 - Geração de relógio de requisição à memória.....	63
28 - Controlador de fim de linha.....	64
29 - Controle de memória dinâmica.....	65
30 - Geração de endereços.....	72
31 - Circuitos de endereço na escrita.....	73
32 - Circuito de apagamento de vídeo no retrazo.....	75
33 - Circuito conversor D/A para intensidade.....	75
34 - Fluxograma da rotina FRAME.....	79
35 - Fluxograma da rotina POPDAT.....	80
36 - Fluxograma do programa 'foreground'.....	81

RESUMO

Neste trabalho são estudados alguns tópicos referentes a terminais de computadores com enfoque especial nos terminais de vídeo, sendo descrito, também, um projeto de terminal alfanumérico e gráfico.

O terminal alfanumérico apresenta na tela, 25 linhas, cada uma com 80 caracteres e é construído a partir de somente 22 integrados cabendo completamente numa placa S-100 padrão.

O terminal gráfico, totalmente desenvolvido pelo autor no Laboratório de Instrumentação Eletrônica do DFCM, apresenta uma resolução de 512 por 250 pontos (128000 pontos) e pode ser expandido até ser capaz de mostrar na tela, 256 níveis de intensidade ou cor. No sistema mínimo, o terminal necessita de apenas duas placas do padrão S-100. A filosofia aqui é o mapeamento de bits, ou seja, todas as informações dos 128000 pontos são armazenadas numa memória. Tanto o terminal alfanumérico quanto o gráfico apresentam imagens através da varredura por rastro.

ABSTRACT

In this work, some topics concerning computer terminals are studied, specially focusing on video ones and a project of an alphanumeric and graphic terminal is described.

The alphanumeric terminal presents 25 lines of 80 characters each, is made of only 22 integrated circuits, being completely contained within a standard S-100 board.

The graphic terminal, totally projected by the author in the Laboratório de Instrumentação Eletrônica, has a resolution of 512 by 250 dots (128000 dots) and can be expanded up to 256 levels of intensity or 256 different colors per dot. The minimum system occupies only 2 standard S-100 boards. The operating principle is bit-mapped, i.e., all 128000 dots are stored in a memory. Both the alphanumeric and the graphic terminals display the memory images by raster scanning.

INTRODUÇÃO

O desenvolvimento de terminais gráficos para computadores tem sido muito acentuado nos últimos anos. O barateamento das memórias a semicondutor tem tornado esse equipamento cada vez mais acessível mesmo a particulares. O uso de receptores comuns de televisão como tela gráfica, também ajudou a universalizar esse equipamento, e acredita-se que essa tendência se deva pronunciar nos próximos anos.

São muitas as aplicações possíveis para um terminal gráfico. Somente na Física, o número de aplicações é incontável e, a cada dia, é sentida uma maior necessidade desse aparelho por parte dos pesquisadores. Na verdade, o terminal gráfico pode ser considerado a melhor forma de comunicação entre o computador e o ser humano. Assim, devido à grande aceitação que o computador tem encontrado nos meios científicos, não é difícil perceber por que existe tanto interesse no terminal gráfico.

Surgiu então, no Laboratório de Instrumentação Eletrônica do DFCM, a idéia de se desenvolver um terminal gráfico que fosse capaz de exibir desde simples curvas X-Y até complexas imagens com vários níveis de intensidade. Esse terminal deveria poder ser ligado ao computador central ou ao sistema modular que ora está sendo desenvolvido, e deveria também poder funcionar como um terminal alfanumérico comum. Dessa idéia então, surgiu este trabalho.

O trabalho focaliza basicamente o 'hardware' de um terminal alfanumérico e gráfico de baixo custo. O 'software' desenvolvido é o mínimo necessário ao funcionamento do aparelho mas, neste momento, ele já está sendo aperfeiçoado e será apresentado em futuros trabalhos por outros membros do LIE.

A presente dissertação é dividida em cinco capítulos. No primeiro, dá-se uma visão geral das características técnicas dos mais variados tipos de terminais existentes. É um capítulo de fácil leitura e que exige pouco conhecimento anterior do assunto.

O segundo capítulo começa com o projeto propriamente dito, explicando a arquitetura geral do terminal.

No terceiro capítulo, analisa-se o 'hardware' do terminal com certo detalhe. É um capítulo de grande densidade de informações e de leitura um tanto difícil para pessoas não diretamente ligadas à área, sendo exigidas para o seu completo entendimento, muitas referências a manuais técnicos.

No quarto capítulo, descreve-se o 'software' utilizado no terminal, enquanto que, no quinto, são sugeridas várias linhas de trabalho com a finalidade de melhorar o terminal.

No transcorrer do texto, as palavras em língua inglesa consagradas em Eletrônica e Computação pelo uso contínuo, foram mantidas em inglês, vindo sempre entre aspas, o mesmo acontecendo com as siglas de origem inglesa. As siglas utilizadas no texto são dadas abaixo com o respectivo significado:

1. UCP - Unidade Central de Processamento
2. CRT - 'Cathode Ray Tube'
3. LED - 'Light Emittins Diode'
4. LCD - 'Liquid Cristal Display'
5. CAD - 'Computer Aided Design'
6. DVST - 'Direct View Storage Tube'
7. D/A - Digital/Analógico
8. ASCII - 'American Standard Code for Information Interchange'
9. RGB - 'Red, Green and Blue'
10. NMR - 'Nuclear Magnetic Ressonance'
11. HP - Hewlett-Packard Inc.
12. IBM - International Business Machines
13. EIA RS-232c - Norma da 'Electronic Industries Association'

Os sinais elétricos citados no texto são (Para a relação de sinais S-100, consulte Elmquist (1)):

1. RESET OUT - indica inicialização do processador
2. CLK OUT - saída de relógio do processador
3. RD* - leitura de dados pelo processador
4. WR* - escrita de dados pelo processador
5. A0 --> A15 - duto de endereço do processador

6. D0 --> D7 - duto de dados do processador
7. MRD*,MWR* - leitura,escrita de dados no espaço de memória
8. IOR*,IOW* - leitura,escrita no espaço de entrada e saída
9. EXT* - indica memória externa (gráfica)
10. MERD*,MEWR* - leitura,escrita em memória externa
11. S0,S1 - Dão o modo de escrita nos bancos de memória gráfica
12. S2,S3,S4 - Selecionam o bit do duto de dados
13. RAS* - 'Row Address Strobe'
14. CAS* - 'Column Address Strobe'
15. XACK* - 'Transfer Acknowledge'
16. R*/C - 'Row/Column'
17. HRTC - 'Horizontal Retrace'
18. VRTC - 'Vertical Retrace'
19. VSP - 'Video Supression'

Finalmente deve ser notado que:

1. Todo o sinal ativo baixo (ativo em 0), vem seguido de um '*'
2. Os números seguidos por 'H' são hexadecimais e os seguidos por 'B', binários.

CAPITULO I

INTRODUÇÃO AOS TERMINAIS DE COMPUTADORES

I.1 DEFINIÇÃO

Um computador é uma máquina capaz de armazenar um conjunto de dados e de instruções (operações sobre números, comparações) e de executá-las quantas vezes for necessário.

O computador, visto como um sistema que aceita dados, age sobre eles e produz uma saída, pode ser subdividido em 3 subsistemas básicos, como mostra a figura 1 :

1. A unidade central de processamento (UCP), responsável pelo controle geral do computador e pela execução das instruções.
2. A memória, principal ou secundária, responsável pelo armazenamento do programa (conjunto de instruções) e dos dados, e pelo fornecimento dos mesmos para a unidade central, quando esta assim o desejar.
3. Um conjunto de dispositivos de entrada e saída (E/S), responsável por toda interação entre a máquina e o ser humano que a utiliza.

É importante notar que a memória e a unidade central de processamento são os subsistemas que realmente realizam o trabalho esperado de um computador. Porém, o subsistema de entrada e saída é fundamental, pois é ele que permite que o ser humano tire algum proveito do trabalho executado pelo computador na medida em que é ele que liga a máquina ao mundo externo.

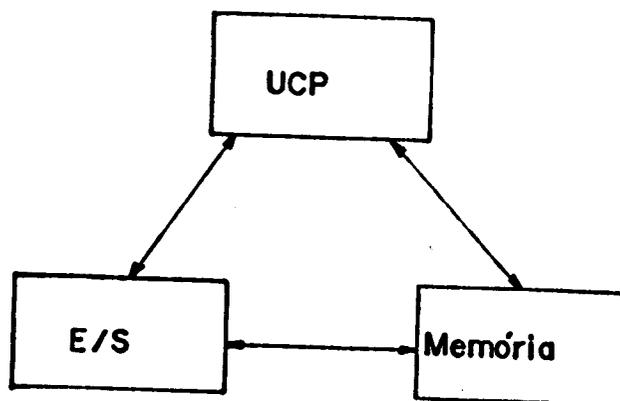


Fig. 1- Subsistemas de um computador

Todos os dispositivos de entrada ou saída podem ser chamados de terminais do computador. O uso, porém, consagrou o nome 'terminal' para dispositivos que são, ao mesmo tempo, entrada e saída, como por exemplo o terminal de vídeo, que é constituído de um teclado, pelo qual se podem entrar dados, e de uma tela, na qual se pode ver a saída.

Dispositivos somente de entrada ou somente de saída receberam nomes específicos como leitora de cartões (em geral, só de entrada), impressora, 'plotter' (só de saída), etc.

Este trabalho descreve o projeto de um terminal de vídeo que use como dispositivo de entrada e saída. A ênfase maior, porém, é sobre o terminal como uma saída de dados.

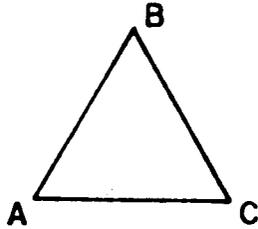
1.2 ARQUITETURAS DE TERMINAIS DE SAÍDA

Existem fundamentalmente dois modos de se desenhar uma figura, ambos bastante utilizados:

1. Varredura aleatória ('Random Scan').
2. Varredura de rastro ('Raster Scan').

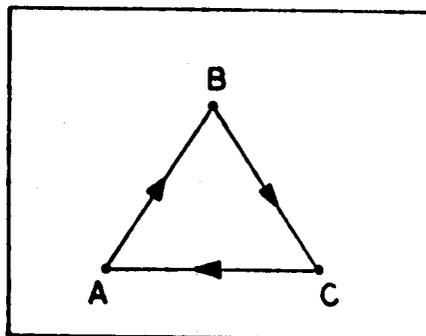
Na varredura aleatória ('Random, Vector, Stroke ou Calligraphic') podem-se desenhar pedaços da figura em qualquer ordem. Parte-se de um ponto qualquer e faz-se a figura. Esse método é utilizado pelo ser humano, por exemplo, ao escrever.

Seja o triângulo da figura 2.



Fis. 2- Um desenho qualquer: um triângulo

Para desenhá-lo com varredura aleatória, por exemplo, parte-se do vértice A e então faz-se como na figura 3.



Fis. 3- Triângulo desenhado por varredura aleatória

Se o terminal for um tubo de raios catódicos ('CRT') faz-se:

1. Traz-se o feixe de elétrons, deslizado, para a posição A.
2. Liga-se o feixe e traça-se o segmento de reta AB.
3. Traça-se o segmento de reta BC.
4. Traça-se o segmento de reta CA.

No método de varredura por rastro, o lugar onde será feito o desenho é varrido sistemática e completamente, sendo que só se desenha nos pontos onde realmente a figura assim o exige.

A varredura por rastro é utilizada nos aparelhos receptores de televisão. O padrão de varredura pode ser visto na figura 4.



Fig. 4- Padrão 'Raster Scan'

Repare que o traço sai do canto superior esquerdo e vai horizontalmente até a extremidade direita quando então retorna rapidamente para a margem esquerda (retrazo horizontal) e começa a próxima linha.

Para se desenhar o mesmo triângulo da figura 2 pelo método de varredura de rastro, deixa-se o traço sempre deslizado, a não ser nos pontos do triângulo. Obtém-se, assim, algo como a figura 5.

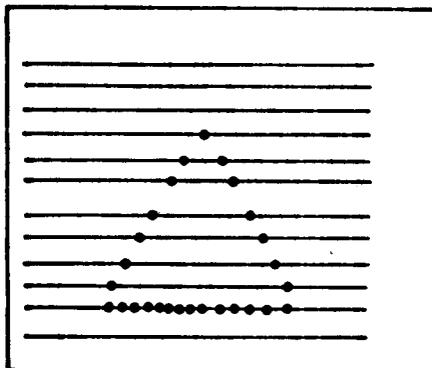


Fig. 5- Triângulo desenhado por "Raster Scan"

As vantagens e desvantagens de cada método serão vistas nas secções I.7 e I.8. Mais detalhes em (2).

I.3 CARACTERES E GRAFICOS

Pode-se dividir o tipo de saída visual de um computador em dois grandes grupos quais sejam:

1. **Caracteres Alfanuméricos:** São fundamentais na medida em que são necessários para a leitura e escrita de informações na linguagem do ser humano. Praticamente, todos os computadores atuais apresentam algum tipo de terminal de entrada e saída com caracteres alfanuméricos.
2. **Gráficos:** Para muitas aplicações, é conveniente que o computador consiga apresentar os seus resultados numa forma gráfica, já que a representação gráfica é uma das que mais se adaptam à compreensão do ser humano. Os gráficos vão desde curvas desenhadas em eixos X-Y até imagens complexas do tipo fotografia. Nem todos os computadores têm terminais com capacidade gráfica, o que é devido a dois motivos básicos:

1. Os gráficos apesar de melhorarem bastante a relação homem-máquina, não são sempre absolutamente necessários.
2. O preço de terminais com capacidade gráfica é consideravelmente maior que o dos outros.

A tendência atual é, porém, reduzir-se bastante o preço de terminais gráficos, principalmente utilizando-se cada vez mais aparelhos comuns de recepção de TV, como será visto à frente.

I.4 TERMINAIS APENAS DE SAÍDA

Têm-se utilizado principalmente dois tipos de terminais apenas de saída que são a impressora e o 'plotter'.

I.4.1 IMPRESSORA

É a saída mais comum encontrada nos computadores.

Ela é ideal para a saída de caracteres para o que, aliás, ela é projetada.

Para saída gráfica, ela apresenta limitações sérias das quais a principal é sua baixa resolução.

Por resolução entende-se o número de elementos distinguíveis por unidade de comprimento.

Apesar de sua baixa resolução, a impressora tem sido bastante utilizada para produzir gráficos devido ao seu baixo custo e à sua disponibilidade.

Os gráficos mais comuns apresentados pela impressora são funções de uma variável, gráficos de barra e cartas de fluxo. Porém, mesmo imagens têm sido conseguidas em impressora. Para simular-se os níveis de intensidade característicos de imagens, tem-se utilizado o recurso da sobre-impressão, ou seja, quando se quer um ponto mais escuro, ele é impresso mais de uma vez. A qualidade da imagem porém, é, em geral, apenas aceitável.

É importante notar que a maioria das impressoras trabalha por varredura de rastro.

I.4.2 "PLOTTER"

Pode ser visto como uma impressora que trabalha pela varredura aleatória, apesar de existirem "plotters" que trabalham com varredura de rastro.

Assim como a impressora, pode produzir qualquer caracter alfanumérico, somente que com muito mais versatilidade (vários tamanhos, várias cores) apesar de que muito mais lentamente.

Devido à sua característica de trabalhar, em geral, com varredura aleatória, a sua resolução é bem melhor que a da impressora. Assim, é ideal para muitos tipos de desenhos, curvas de funções, etc. Sua limitação só é bastante sentida quando se quer a produção de imagens contínuas, ou quando se exige alta velocidade de saída.

I.4.3 OUTROS

Têm-se desenvolvido ultimamente, várias outras tecnologias de terminais apenas de saída, das quais, talvez, as mais importantes sejam as técnicas xerográficas desenvolvidas pela companhia Xerox e a impressora por "laser".

Essas outras tecnologias ainda não encontraram aceitação geral e são atualmente pouco utilizadas.

I.5 TERMINAIS DE VIDEO

Um problema sério com os terminais de saída analisados até agora é que eles são muito lentos. Eles não podem ser utilizados em aplicações onde é exigida uma rápida mudança na imagem produzida, como é o caso nas aplicações interativas. Por aplicações interativas, entendem-se aquelas em que um usuário dinamicamente controla o conteúdo, cor e tamanho da imagem através de alguns dispositivos de entrada como teclado, "Joystick", etc.

Nesse caso, é necessário um terminal de entrada e saída, saída esta que deve poder ser modificada o mais rapidamente possível.

Ao mesmo tempo, pode-se notar que nenhum dos terminais apresentados até agora, se presta bem à confecção de complexas imagens contínuas.

Os terminais de vídeo vêm justamente a corrigir todos esses problemas, sendo, portanto, extremamente utilizados atualmente.

Existem muitos tipos de vídeo como: o tubo de raios catódicos comum, o tubo de raios catódicos colorido por penetração de feixe, o tubo de raios catódicos colorido por máscara de sombra, o de painel de plasma e os vídeos de diodos emissores de luz ("LED") e os de cristal líquido ("LCD").

Entre todas essas tecnologias, o vídeo mais utilizado é, indiscutivelmente, o tubo de raios catódicos ("CRT").

1.5.1 O TUBO DE RAIOS CATÓDICOS ("CRT")

O "CRT" utilizado em terminais de vídeo de computadores é basicamente o mesmo utilizado em televisores comuns.

Na figura 6, vê-se a seção transversal do CRT.

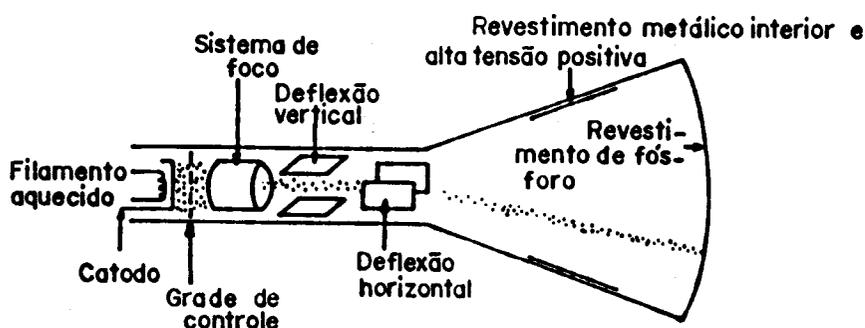


Fig. 6- Seção transversal de um tubo de raios catódicos

Por efeito termiônico, um conjunto de elétrons é emitido do cátodo aquecido pelo filamento. Esse conjunto de elétrons, então, passa por uma grade de controle e também por um sistema de focalização elétrica, que faz o conjunto de elétrons se transformar num feixe. O sistema de deflexão se encarrega de dirigir o feixe para um determinado ponto da tela, onde os elétrons se chocam com o fósforo. Produz-se, então, luz

visível. A velocidade com que o feixe se choca com a tela é determinada pelo valor da alta tensão positiva aplicada na camada metálica interna ao tubo. A essa diferença de potencial, dá-se o nome de voltagem de aceleração.

A intensidade do ponto na tela é controlada pelo potencial da grade. Quanto mais negativo for, menor a quantidade de elétrons que saem do cátodo e vão para a tela sendo menor a intensidade, já que esta é proporcional ao número de elétrons na unidade de tempo. Eventualmente, se a grade for muito negativa, nenhum elétron passa e a tela fica apagada.

Para se desenhar algo na tela, é preciso agir sobre a deflexão horizontal e a vertical, de forma a fazer o feixe se deslocar segundo a figura que se deseja criar. Repare que a emissão luminosa do fósforo cai exponencialmente com o tempo, de forma que é preciso constantemente redesenhar a figura para que ela pareça parada a um observador humano.

O sistema de focalização de lentes usa campos magnéticos ou elétricos para fazer a convergência dos elétrons. Quando se usa focalização magnética, colocam-se os focalizadores externamente ao tubo, o contrário acontecendo quando se utilizam campos elétricos. A situação é bastante parecida com a focalização ótica, com a única diferença que o feixe de elétrons tem uma tendência natural a diversir.

Na figura 7, vê-se o sistema de focalização do 'CRT' que apresenta um dos problemas do mesmo.

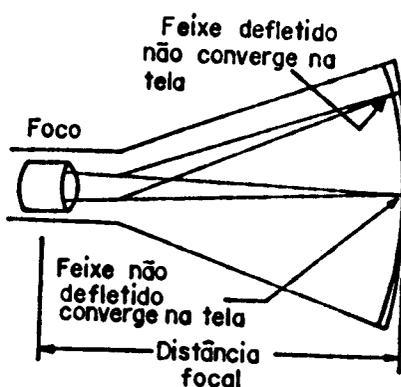


Fig. 7- Focalização no "CRT"

Vê-se que, se o feixe for focalizado para o centro da tela, ele não estará focalizado nos outros pontos da tela, pois o raio de curvatura do tubo não é o mesmo do foco. Assim, a imagem nas extremidades aparecerá borrada. Em sistemas gráficos de alta precisão, a focalização é feita dinamicamente e o foco é ajustado, dependendo da posição onde o feixe atingirá a tela.

Também o sistema de deflexão do "CRT" pode ser elétrico ou magnético. Assim como na focalização, se for elétrico, será interno, enquanto que, se for magnético, será externo ao tubo e formado por duas bobinas.

Na deflexão eletrostática, o controle da posição do feixe é muito simples, já que esta é linear com a tensão aplicada. A mesma facilidade, porém, não é encontrada na deflexão magnética, pois a posição não é linear com a corrente, sendo a dependência dada por:

$$D = \frac{L \times k \times I}{\sqrt{1 - (k \times I)^2}}$$

onde D é a deflexão, L é a distância das bobinas defletoras até a tela, I é a corrente e k, uma constante (3), (4), (5).

Repare que D é apenas grosseiramente linear com I.

Na verdade, se for suposto D linear com I e se não for considerado o efeito que a deflexão dum bobina tem sobre a outra, será obtido na tela o que é mostrado na figura 8 (3).

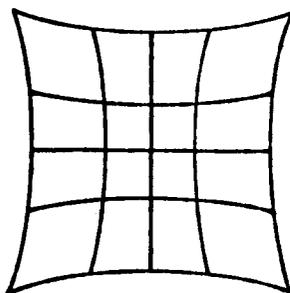


Fig. 8- Deflexão magnética não compensada

Na figura vê-se que as linhas não centrais aparecem bastante distorcidas. Mesmo nas linhas centrais, as distâncias entre pontos que deveriam ser igualmente espaçados, variarão.

Para corrigir esses problemas, pode-se colocar ímãs permanentes ao redor do tubo ou utilizar circuitos de compensação.

Apesar dessa desvantagem, e também do fato de a velocidade de escrita no caso da deflexão eletrostática ser maior (o que permite colocar mais informações na tela sem "flicker"), a deflexão magnética tem sido amplamente preferida pelos seguintes aspectos:

1. A construção do tubo é mais simples, visto que o sistema de deflexão é externo.

2. Em sistemas magnéticos, as voltagens de aceleração utilizadas são maiores, o que permite maior brilho.
3. Consegue-se menor tamanho de ponto.
4. Os tubos são menores.

Outra característica muito importante do "CRT", que deve ser levada em conta num projeto de terminal de vídeo, é o fósforo que é utilizado.

O elétron acelerado choca-se com a tela que é recoberta com uma camada de fósforo. Nesse choque, ele transfere energia para que os elétrons dos átomos de fósforo mudem de níveis de energia. Assim, por fluorescência e fosforescência, o fósforo emite luz visível. A emissão de luz, porém, decai exponencialmente com o tempo, decaimento esse que muda para os vários tipos de fósforo.

Para estudar-se o decaimento do fósforo, define-se o tempo de persistência que é o tempo contado a partir do instante em que se retira o feixe de elétrons, até o instante em que a emissão cai a 10 por cento do valor inicial.

Em fósforos utilizados em terminais de vídeo, a persistência varia entre 10 e 60 microssegundos.

A persistência do fósforo utilizado é um dado muito importante no projeto do terminal, pois é ele que determina a complexidade máxima da imagem apresentada que não terá tremulação ("flicker"). A taxa de refrescamento exigida é menor quanto maior for a persistência do fósforo utilizado.

Observe que, se a taxa de refrescamento não for suficientemente alta, a imagem apresentará tremulação ('flicker'). A frequência de refrescamento, a partir da qual não existe mais tremulação e a imagem parece completamente estável ao observador, é chamada de frequência de fusão.

Para um 'CRT' normal, utilizam-se taxas de refrescamento de trinta a sessenta por segundo. Na verdade, costuma-se utilizar trinta ou sessenta exatamente, pois, somente nessas frequências, não se terão problemas com a interferência da rede (a oscilação induzida pela rede não terá efeito algum no terminal).

Existem vários tipos de fósforos que são regularmente utilizados em terminais de vídeo como P1, P4 (utilizado em televisão branco e preto), P7 e P31 (6).

Pode-se obter fósforos de persistência bastante elevada, da ordem de segundos. Normalmente, porém, estes fósforos de alta persistência não são utilizados em terminais gráficos, pois, se for mudada rapidamente a imagem, o gráfico antigo ainda estará presente, causando uma confusão de imagens. Esse fósforo só é utilizado quando se necessita de grande qualidade de imagens fixas, o que ocorre algumas vezes em projetos auxiliados por computador ('CAD').

A persistência não é o único fator que determina a frequência de fusão. Quanto mais intensidade, brilho ou luz ambiental tiver-se, maior ela será. Também não se deve esquecer que a fusão é um fenômeno biológico, e, assim, varia para cada observador.

A qualidade de um "CRT" é medida através de várias características como: resolução, tamanho de ponto, linearidade e velocidade.

A resolução é o inverso do tamanho do ponto. Caso se queira uma resolução de 20 linhas por centímetro, deve-se ter pontos de tamanho 0.5mm.

A linearidade é uma medida que diz quanto uma linha, que deveria ser reta, é realmente reta. É, em geral, medida como a distância máxima entre a linha apresentada pelo terminal e uma reta que lisa os mesmos pontos. Veja a figura 9.

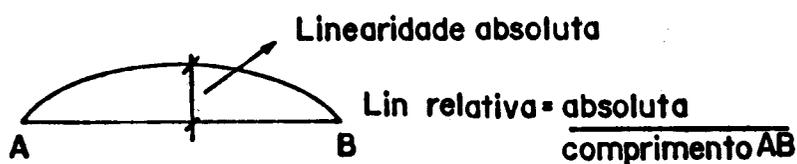


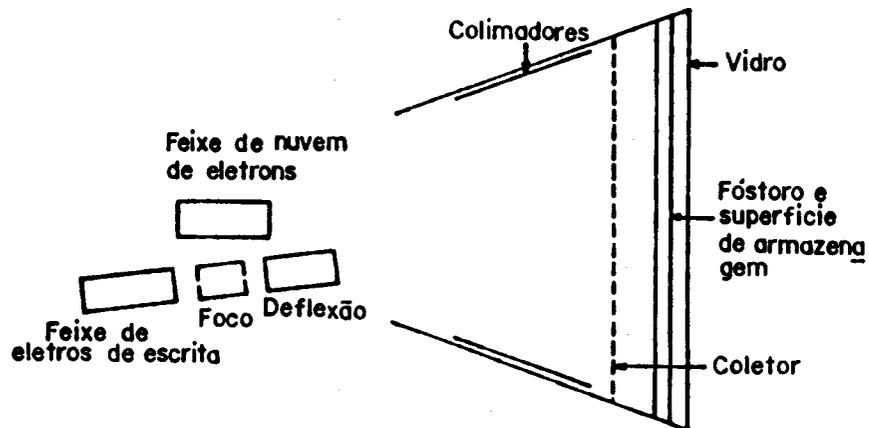
Fig. 9- Linearidade

A velocidade pode ser definida como o quão rápido se consegue mover o feixe até uma determinada posição, e, então, fazer um ponto lá.

1.5.2 TUBO DE ARMAZENAGEM E VISÃO DIRETA ("DVST")

É bastante parecido com o tubo de "CRT" visto na última seção, somente com a diferença que não precisa ser refrescado, já que ele armazena a imagem como uma distribuição de cargas.

A focalização e a deflexão são idênticas às do "CRT". Existe, porém, junto ao fósforo, uma superfície de armazenagem que contém inicialmente cargas negativas. O feixe de elétrons choca-se com essa superfície, retirando dela elétrons, o que faz com que a região atingida fique com polaridade relativamente positiva. Os elétrons retirados da superfície são atraídos por um coletor positivo existente dentro do tubo. Veja figura 10.



Fis. 10- Seção transversal do "DVST"

Uma camada de elétrons não focalizada está constantemente se dirigindo ao fósforo em ângulos retos (devido aos colimadores). Ela, porém, só se choca com o fósforo nas regiões onde a carga na superfície armazenadora é relativamente positiva, sendo repelida no restante da tela. Vê-se assim uma imagem nítida na tela somente nos pontos que foram atingidos pelo feixe de elétrons focalizado e defletido.

Para apagar um padrão armazenado na superfície, primeiramente coloca-se uma tensão positiva na superfície, o que atrai a camada de elétrons (e acende a tela completamente, produzindo o "flash" característico desses terminais). Então, coloca-se uma tensão negativa, o que retém os elétrons, e faz

com que a superfície volte a ficar completamente carregada negativamente.

O "DVST" não é capaz de apagamento seletivo. Além disso, outra grande desvantagem é a não possibilidade, em "DVST" comuns, de níveis de intensidade. Em compensação, a precisão pode ser altíssima, já que não é preciso refrescamento, e também os circuitos de controle são normalmente mais simples e mais baratos.

Esse tipo de tubo é bastante usado na linha de produtos da Tektronix e IBM (7), (8).

1.5.3 TUBOS COLORIDOS

Para algumas aplicações, principalmente as que envolvem imagens, é bastante interessante ter-se terminais coloridos.

Têm-se dois tipos básicos: o "CRT" colorido por penetração de feixe e o colorido por máscara de sombra.

1.5.3.1 "CRT" COLORIDO POR PENETRAÇÃO DE FEIXE -

É construído como o "CRT" comum, somente que com 2 camadas de fósforo de cores diferentes e de níveis de excitação diferentes. Normalmente se utilizam o fósforo vermelho e o verde. Como a energia do feixe é dependente da voltagem de aceleração, controlando-se esta, pode-se excitar o fósforo vermelho ou o verde, produzindo-se as duas cores. Ainda mais, se forem utilizadas voltagens intermediárias, consegue-se excitar parcialmente um fósforo e o outro, produzindo-se assim uma mistura das duas cores.

Apesar de ser o método mais barato de se conseguir cores num tubo de raios catódicos, e de apresentar uma largura de linha menor do que o conseguido no por máscara, o método tem 2 desvantagens muito sérias:

1. O número de cores possíveis é bastante limitado.
2. O tempo para mudar a voltagem de aceleração é grande (de 2 a 200 microssegundos) o que limita bastante a quantidade de informação que pode ser colocada na tela sem tremulação quando as cores mudam constantemente.

1.5.3.2 'CRT' COLORIDO POR MÁSCARA DE SOMBRA -

É o método utilizado em receptores de televisão comerciais.

A camada de fósforo é constituída de uma sucessão de pontos dos quais um é de fósforo azul, outro de verde e o outro de fósforo vermelho. Ao invés de um feixe, têm-se agora três, correspondendo a cada uma das cores. Cada um dos feixes só atinge o fósforo correspondente devido a uma máscara colocada entre a origem do feixe e a tela. Veja a figura 11.

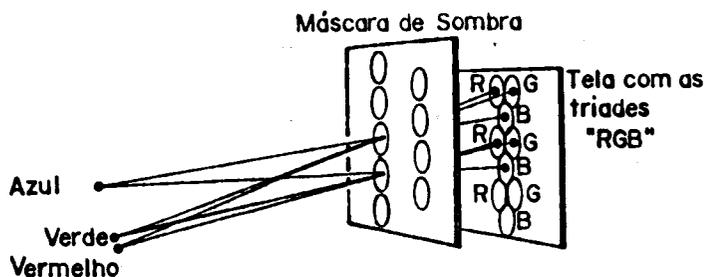


Fig. 11- 'CRT' colorido por máscara

A grande limitação desse tubo é a resolução, já que dois fósforos da mesma cor acabam ficando relativamente distantes. Na verdade, em receptores comuns de TV, a distância é de 0.6mm, enquanto que, em monitores de alta resolução, consegue-se 0.35mm ou menos.

I.5.4 TERMINAL DE PLASMA

Tem sido utilizado algumas vezes pela IBM. Por ter características bem diferentes dos terminais até agora vistos, é facilmente reconhecido por uma peculiaridade: a tela, mesmo grande, é completamente plana. Lembre-se que a curvatura no vídeo do "CRT" era devida à focalização do feixe de elétrons, que aqui não existe.

O terminal de plasma, como o visto na figura 12, é formado por um conjunto de pequenas lâmpadas de um gás, em geral o neon, que podem estar ligadas ou desligadas. Uma vez ligada ou desligada, a pequena lâmpada permanece naquele estado até este ser especificamente mudado.

A resolução do terminal de plasma não é muito alta, ao contrário do seu preço que é relativamente elevado. Assim, ele não tem sido muito utilizado. Para maiores detalhes, consulte Slottow (9) e Tannas (10).

I.5.5 VIDEOS DE DIODOS E CRISTAL LIQUIDO

São tecnologias de estado sólido que prometem muito para o futuro. Os vídeos com diodos emissores de luz ("LED") e os de cristal líquido ("LCD") são compostos por um conjunto de células arranjadas matricialmente.

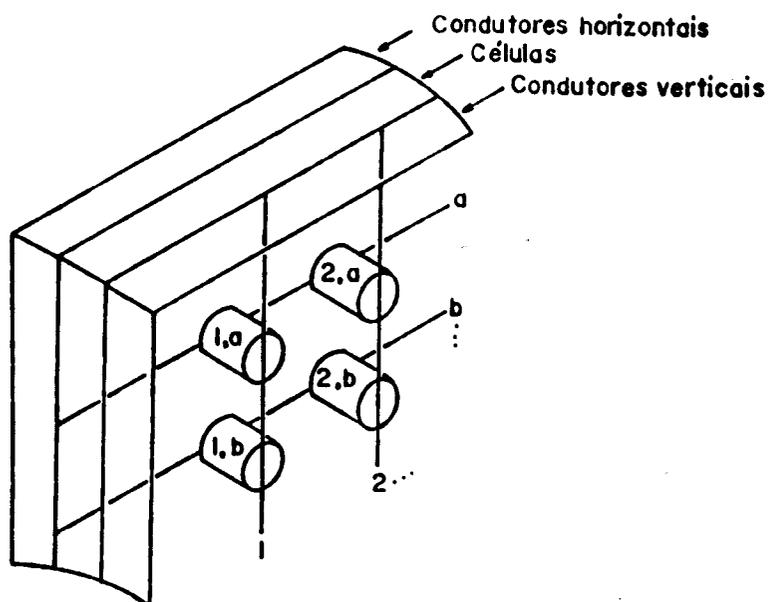


Fig. 12- Terminal de plasma

Podem ter um grande futuro mas, atualmente, ainda não são utilizados. Detalhes sobre tecnologias avançadas podem ser encontrados em Woodsford (11).

1.6 ARQUITETURAS DE VARREDURA ALEATORIA

Como foi visto na seção 1.2, por varredura aleatória ou randômica entende-se o método de criar um desenho partindo-se de um seu ponto qualquer, e, então, tracando a figura. É exatamente o método utilizado pelo ser humano.

Pode-se, porém, dividir o processo de varredura aleatória em dois tipos, segundo a unidade básica de tracado, quais sejam:

1. Por ponto

2. Por reta

Como será visto, ambos apresentam vantagens e desvantagens. O método de traçado por reta, porém, tem sido amplamente preferido nos sistemas gráficos encontrados comercialmente.

No traçado por ponto, traz-se o feixe para o ponto desejado (X_0, Y_0) e liga-se o feixe. Para continuar o desenho, desliga-se o feixe, leva-se o feixe para o próximo ponto (X_1, Y_1) e então religa-se o feixe e assim por diante até formar todo o desenho. Se o terminal de vídeo existir refrescamento, deve-se repetir esse procedimento numa frequência maior que ou igual à de fusão, para se ter uma imagem sem tremulações.

A arquitetura desse sistema está mostrada na figura

13.

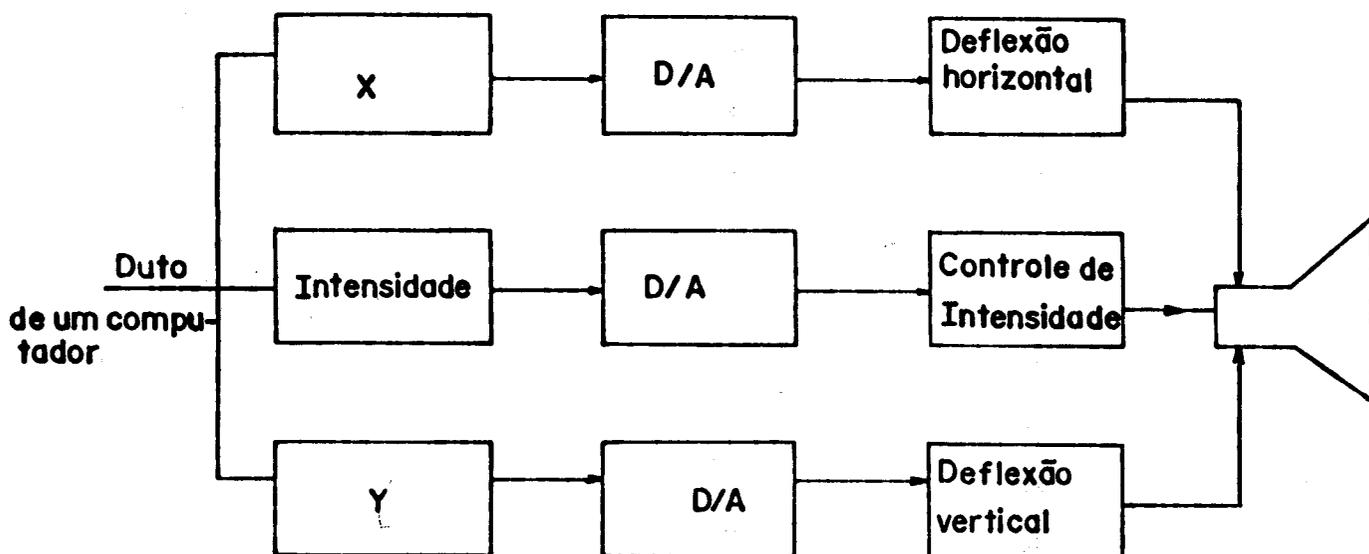


Fig. 13- Traçado por ponto

Numa arquitetura desse tipo, a resolução é dada pelo número de bits dos conversores D/A de X e Y, enquanto que os níveis de intensidade são determinados pelo número de bits do conversor D/A da intensidade. Isso, é claro, se não houver fatores limitantes no interior do monitor, como, por exemplo, uma baixa resolução do 'CRT'.

O trabalho do computador é simplesmente carregar os 3 registradores X, Y e intensidade com os valores apropriados para a confecção do desenho.

O computador, porém, pode ser dedicado ao terminal de vídeo ou não. Obviamente, no caso de se ter vídeo que necessite de refrescamento, o tempo livre do processador será muito baixo, principalmente se o desenho for muito complexo. Vê-se, assim, que, nesse caso, é fundamental ter-se uma unidade de processamento dedicada ao terminal. Ainda assim, pode ocorrer que, em figuras complexas, o processador não seja rápido o suficiente para produzir uma imagem sem tremulação. Além disso, espera-se que o processador mesmo sendo dedicado, execute algumas rotinas simples, como por exemplo, de conversão de varredura ("scan conversion algorithms").

Vê-se, portanto, que o problema de tempo nessa arquitetura é seriíssimo, e que o processador ligado a ela deve ser muito rápido. Também deve ter uma memória grande, pois deve armazenar todas as partes da figura.

A grande vantagem aqui é a simplicidade e baixo custo. De fato, com esse esquema simples e barato, consegue-se utilizar praticamente ao máximo a resolução do monitor que se dispõe. Também o processador, apesar de rápido, não necessita ser muito 'inteligente' na medida em que tudo o que ele tem a fazer é carregar os três registradores com conteúdos que estejam numa memória.

Em Pascal, um programa para esse processador seria:

```
Repeat
    Espera_tempo_refrescamento;
    For I:=1 to N do
        Begin
            Repeat
                until Pronto;
            Registro_X(X[I]);
            Registro_Y(Y[I]);
            Intensidade(Z[I]);
        End
    until Fim
```

onde a rotina Espera_tempo_refrescamento simplesmente espera pelo próximo instante de fazer refrescamento (de 1/30 em 1/30 s ou de 1/60 em 1/60s tipicamente), pronto é uma variável booleana que avisa quando o último ponto já foi desenhado, as rotinas RegistroX, RegistroY e Intensidade somente carregam os respectivos registradores enquanto Fim também é booleana e indica o momento de apagar o desenho.

Para solucionar alguns problemas do método do traçado por ponto, criou-se o método de traçado por retas que é basicamente idêntico ao do ponto somente que agora é acrescentado 'hardware' suficiente para se desenhar uma reta e não somente um ponto.

A arquitetura básica desse esquema é also como o mostrado na figura 14.

Repare que foram acrescentados dois registradores e um gerador de vetores em relação ao esquema da figura 13.

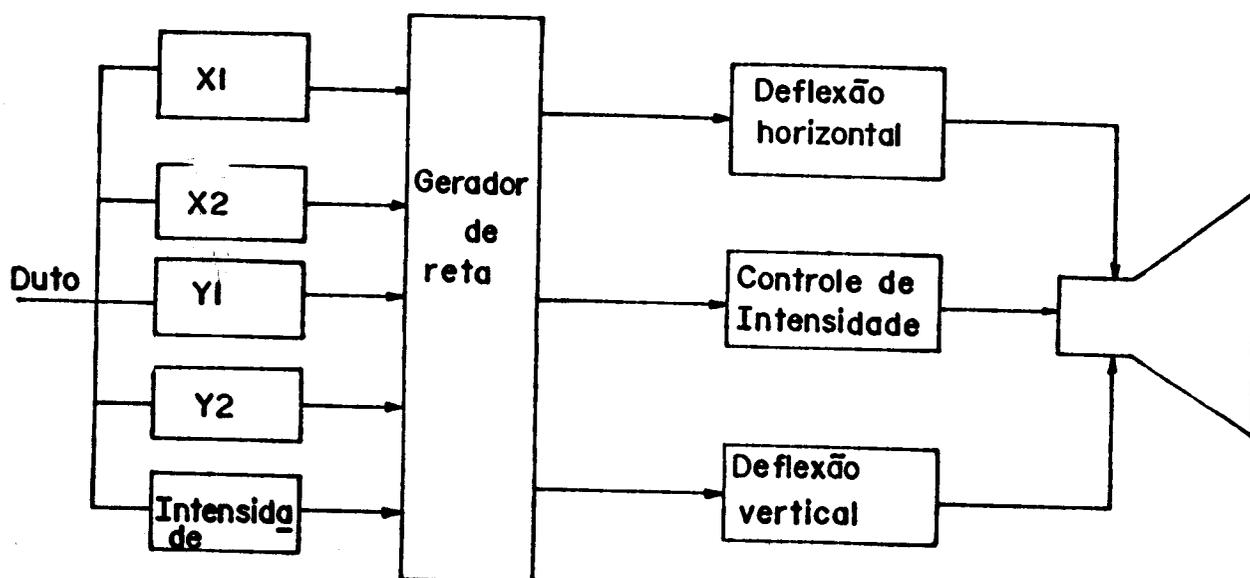


Fig. 14- Traçado por reta

O computador deve carregar os registradores X1 e Y1 com as coordenadas do ponto inicial da reta e X2 e Y2 com as coordenadas do ponto final e também o registrador de intensidade.

Desta forma, o traçado de figuras fica bastante simplificado e o problema de tempo para o processador se torna bem menos crítico. Também, uma figura precisará de bem menos memória para ser armazenada, já que somente os pontos iniciais e finais de cada segmento necessitam ser guardados.

Na verdade, o que se fez, foi passar um trabalho que era do processador para o 'hardware' do terminal. O gerador de retas introduzido, porém, traz consigo alguns problemas. Em primeiro lugar, aumenta o custo do terminal. Também, para desenhos que não contenham muitas retas, o ganho em tempo será bastante pequeno. Acrescente-se a isso o fato de que não é, em hipótese alguma, trivial a construção de um gerador de retas. De fato, é responsabilidade do gerador de retas traçar uma linha, o mais reta possível de (X_1, Y_1) para (X_2, Y_2) , traçar todas as retas que tenham a mesma intensidade, realmente com igual intensidade, e ligar e desligar o feixe exatamente no começo e no fim da reta.

Essas características que podem parecer fáceis, são bastante difíceis de serem conseguidas. Traçar as retas com mesma intensidade, por exemplo, exige que a velocidade de varredura do feixe seja exatamente igual para todas as retas.

Há dois tipos de gerador de retas, os analógicos e os digitais. Os analógicos são feitos à base de integradores que produzem rampas que serão as deflexões nos eixos X e Y. Já os digitais partem do valor (X_1, Y_1) e vão incrementando X e Y proporcionalmente à inclinação da reta até chegar a (X_2, Y_2) . Apesar dos digitais serem mais simples de ajustar e de, em certos casos, serem mais rápidos, o gerador de retas analógico

desenha uma reta bem mais perfeita, sendo que ela não terá aqueles patamares característicos da geração digital. Existem sistemas gráficos comerciais que conseguem, com geração analógica, uma qualidade de tracado de reta extremamente boa.

Na figura 15 vêem-se os sinais gerados na deflexão X e na Y para traçar uma reta pelo método analógico.

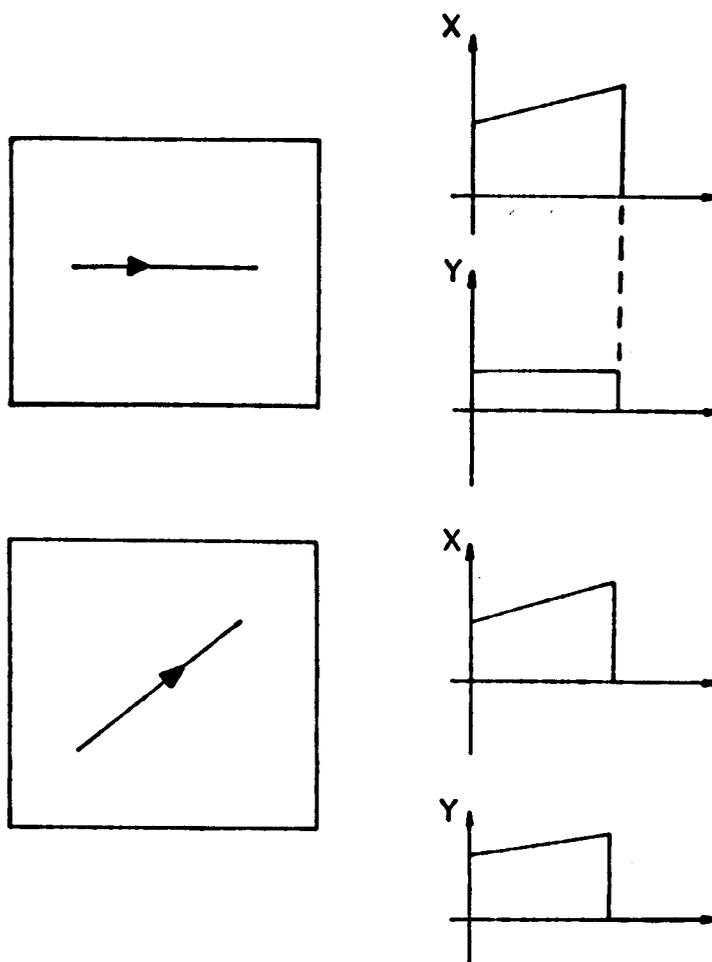


Fig. 15- Tracado analógico de retas

Cumpré notar que em aparelhos comerciais que usam essa arquitetura, duas outras facilidades são acrescentadas. Em geral, existe um somador no circuito para que seja possível fazer-se desenhos com coordenadas relativas. Assim, um desenho pode ser mudado de posição na tela com uma simples mudança de um

ponto de referência. Normalmente, também, existe um gerador de caracteres que recebe uma entrada em código (por exemplo, ASCII) e gera os sinais para deflexão X e Y. Repare que, sem isso, o trabalho do processador para textos seria muito grande e consumiria muito tempo.

1.7 VARREDURA ALEATORIA EM SISTEMAS COMERCIAIS

Como foi visto, a varredura aleatória não é muito boa para imagens, mas é ideal para confecção de curvas. Dessa forma, é muito útil em projetos auxiliados por computador ("CAD") devido à sua facilidade em apresentar, com muita perfeição, contornos de peças e máquinas em geral.

Comercialmente, tem sido utilizada em sistemas da HP, IBM e Tektronix. Esta última apresenta sistemas gráficos de alta resolução com telas do tipo "DVST". Repare que, com tubos com memória, o refrescamento não é necessário, o que alivia sobremaneira o trabalho do processador gráfico.

1.8 ARQUITETURAS DE VARREDURA POR RASTRO

Têm sido bastante mais usadas que as por varredura aleatória, já que aqui se pode utilizar um aparelho receptor de televisão comum sem qualquer ajuste ou adaptação.

Repetindo o que já foi dito na seção 1.2, a grande diferença entre a varredura por rastro e a aleatória de traçado por ponto é que na por rastro todos os pontos são varridos numa determinada ordem, em geral como o mostrado na figura 4.

Assim, cada ponto da tela tem associada uma informação de n bits, onde n pode ir de 1 a 30 bits (dificilmente são encontrados sistemas com $n > 30$ bits o que será explicado adiante). Esse ponto é chamado de pixel (do Inglês, 'Picture Element').

Cada vez que a tela é montada, (30 a 60 vezes por segundo, tipicamente), todas essas informações são exigidas pelo sistema. Vê-se assim que a existência de memória é bem maior aqui do que o era nos sistemas por varredura aleatória.

Uma arquitetura típica está esboçada na figura 16.

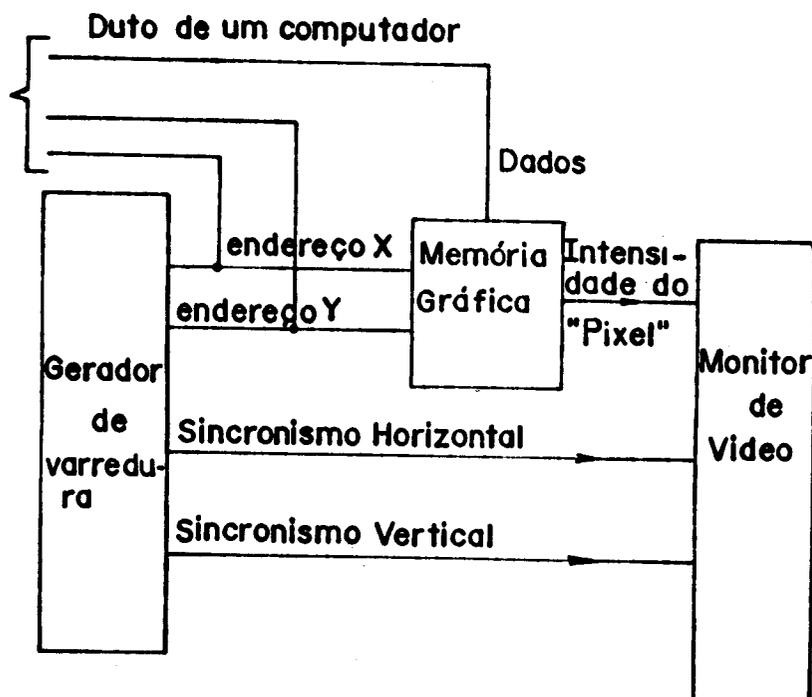


Fig. 16- Sistema de varredura por rastro

O gerador de varredura gera os endereços corretos da memória e os pulsos de sincronismo para o monitor. Ele começa a desenhar acessando a memória que contém a informação referente ao pixel (0,0). A memória responde diretamente ao monitor e

então é escrito o primeiro pixel na tela. Depois, é acessada a posição $(0,1)$ e o processo se repete sucessivamente até a posição $(0,C-1)$, que constitui o último pixel da linha. Nesse ponto, já foi desenhada uma linha horizontal completa e, então, é gerado um pulso que avisa o monitor que ele deve voltar para o começo da linha seguinte. A esse pulso de sincronização horizontal, dá-se o nome de retrazo horizontal. O processo continua a partir de $(1,0)$ até $(1,C-1)$, quando novo pulso de retrazo horizontal é gerado, e, assim por diante, até a última linha de $(L-1,0)$ a $(L-1,C-1)$, quando então é gerado um pulso que faz o traço voltar para a posição $(0,0)$. Esse pulso é chamado de pulso de retrazo vertical e marca o fim da tela. Nesse instante, tem-se uma imagem de L linhas por C colunas na tela.

O produto do número de pontos numa linha pelo número total de linhas determina a resolução do terminal de vídeo.

Como o controle desses terminais é digital, é comum encontrar-se pelo menos uma das dimensões (L ou C), com valores que são do tipo 2 elevado a n, com n inteiro. Assim, resoluções 256x256, 512x256, 512x512, 1024x512, 1024x1024 são bastante comuns.

Repare que, agora, ao contrário da varredura aleatória, todas as informações são varridas, o que, se por um lado existe uma grande quantidade de memória, por outro faz com que imagens arbitrariamente complexas sejam tão fáceis de se colocar na tela quanto as mais simples.

Muito cuidado deve ser tomado com o cálculo de tempo num terminal de video com varredura de rastro. É preciso fazer pelo menos 30 refrescamentos por segundo, o que fica cada vez mais difícil quanto maior a resolução do terminal. Seja, por exemplo, o refrescamento de um terminal 512x512 numa taxa de 30Hz.

Ora, têm-se 512 linhas o que leva a um tempo por linha de

$$\frac{1}{30 \times 512} = 65 \text{ microssegundos}$$

Considerando-se o retraco vertical que é tipicamente 1.5ms, têm-se

$$\frac{\frac{1}{30} - 1.5 \times 10^{-3}}{512} = 61 \text{ microssegundos}$$

o que mostra que se têm 61 microssegundos por linha (traco + retraco). Considerando um retraco horizontal de 10 microssegundos, vem que o tempo disponível para cada pixel é

$$\frac{61 - 10}{512} = 100 \text{ nanossegundos}$$

Se, porém, tiver-se uma resolução de 1024 por 1024 tem-se, segundo os mesmos cálculos acima, um tempo de 21 nanossegundos que, claramente, é um tempo extremamente pequeno.

Fica claro que, a esse nível, é necessário um monitor de altíssima qualidade, pois o amplificador de vídeo deverá ter uma largura de banda muito grande.

No caso de se utilizar um televisor comum, devido a pequena largura de banda de seu amplificador de vídeo, pode-se conseguir apenas média resolução, ao nível de 256x256. *4/2/77*

1.9 COR E INTENSIDADE

Como já foi dito, quando se utiliza o método de varredura de rastro, imagens complexas podem ser desenhadas. Para reproduzir imagens, porém, é preciso ter-se vários níveis de intensidade ~~e de cor~~. Considerando somente imagens tipo preto e branco, ou seja, acromáticas, a pergunta que surge imediatamente é: quantos níveis de cinza deve-se ter para que a imagem mostrada seja como uma fotografia contínua?

Tem-se verificado que, quando se utiliza 4, 8 e 16 níveis, contornos indesejáveis aparecem, enquanto que, com 32, eles praticamente não existem, desaparecendo completamente quando se alcançam 64 níveis de intensidade. Assim, na maioria das fotografias, 32 ou 64, ou seja, 5 ou 6 bits de intensidade, são suficientes.

O problema que surge em seguida é como distribuir os vários níveis de intensidade. Pode ser provado que as distribuições de intensidade devem ser logaritmicamente espaçadas.

Seja então I_0 , a mais baixa intensidade possível no 'CRT' em uso e 1 a referência para máxima intensidade. Então, se forem necessários 64 níveis, deve-se fazer:

$$I_0 = I_0 \quad I_1 = r \times I_0 \quad I_2 = r^2 \times I_0 \quad \dots \quad I_{63} = r^{63} \times I_0 = 1$$

$$r = \frac{1}{\sqrt[63]{\frac{1}{I_0}}}$$

Como, porém, colocar essas intensidades no 'CRT'?

Sabe-se que a intensidade luminosa está ligada ao número de elétrons que atinge a tela, pela relação:

$$I = K \times N^Y$$

onde K e Y são constantes.

Porém, N está diretamente ligado a tensão da grade de controle e, assim, para uma outra constante C , tem-se

$$I = C \times V^Y$$

Para cada um dos I_j , tem-se então um V_j dado pela relação

$$V_j = \sqrt[Y]{I_j / C}$$

e está resolvido o problema (3), (12).

Há alguns terminais que só possuem dois níveis de intensidade: ligado e desligado.

Esses terminais podem, porém, apresentar imagens com razoável qualidade. O truque utilizado é conhecido como aproximação de meio tom e é bastante utilizado em fotografias de jornais e revistas.

A aproximação de meio tom é baseada no fato de que o olho humano, quando à considerável distância do objeto, não apreende todos os detalhes do objeto, mas realiza uma média desses detalhes. Assim, em livros e jornais, utiliza-se tamanho de ponto inversamente proporcional à intensidade da fotografia original.

Em gráficos, pode-se aproximar o ponto de tamanho variável, utilizando-se de mais de um pixel como unidade elementar. É claro que é perdida resolução espacial, mas, em compensação, ganha-se resolução em intensidade. A figura 17 apresenta um padrão para simular 5 níveis de intensidade, utilizando-se, como unidade, 4 pixels organizados 2x2. A resolução nesse caso cai, obviamente, à metade em cada eixo.

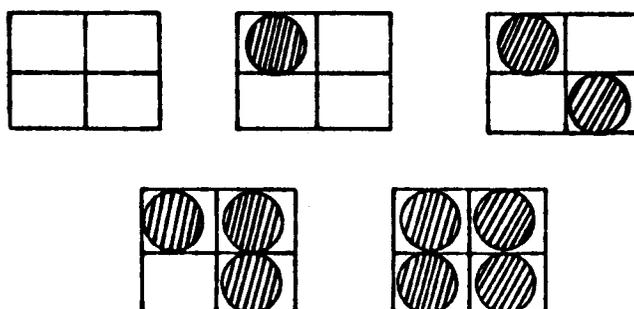


Fig. 17- Padrão para simular 5 níveis de intensidade

Quando se analisa cor, percebe-se que, infelizmente, o problema não é tão simples como no caso de intensidades acromáticas (13), (14), (15), (16), (17), (18). Pode-se demonstrar que, se forem adotadas três cores fundamentais, por exemplo, vermelho, verde e azul ("RGB"), é necessário, para que se possa perceber uma determinada imagem como contínua, reservar 8 bits para cada cor. Assim, o pixel colorido deverá possuir 24 bits de informação. Existem sistemas comerciais que se utilizam de 10 bits por cor, resultando em 30 bits por pixel.

I.10 APLICAÇÕES E PERSPECTIVAS

O terminal de vídeo alfanumérico, hoje, já é uma realidade e, é praticamente impossível num sistema computacional, não haver pelo menos um. Já o terminal gráfico, apesar de ainda não sempre usado, tem encontrado a cada dia mais aplicações e tem experimentado um verdadeiro "boom" em microcomputadores caseiros.

A gama de aplicações dos terminais gráficos é tão grande e aumenta tão acentuadamente que se torna difícil uma listagem completa.

Atualmente, talvez, e no futuro próximo, o maior impacto dos sistemas gráficos seja na área da educação (19), (20), (21), (22). Todas as abstrações necessárias para os grandes cientistas podem ser visualizadas numa tela de televisão, tornando o processo de assimilação bastante simples. Processos de difícil entendimento como a diferenciação, a integração, limites serão triviais se os alunos puderem vê-los. O terminal gráfico será, portanto, uma espécie de professor para o aluno com a vantagem que, com o baixo custo, cada um poderá

ter o seu. Também a impessoalidade desse "Professor", apesar de ter algumas desvantagens, evitará certos constrangimentos.

Também nas artes, principalmente no cinema e na televisão, os gráficos têm sido bastante utilizados. Quem não conhece o famoso "Plim-Plim" de uma conhecida emissora de televisão brasileira e sua imagem associada que foi certamente gerada por computador? No conhecido filme que quebrou todos os "records" de bilheteria do cinema mundial, "Guerra nas Estrelas", algumas cenas foram geradas totalmente por computadores.

Nos projetos auxiliados por computador ("CAD"), os terminais gráficos têm sido de importância fundamental.

Na "NASA", projetistas fizeram simulações de vôos reais de espaçonaves, o que proporcionou bom treinamento aos pilotos, já que o computador mostrava na tela tudo o que o piloto realmente veria.

Na medicina, os gráficos também têm sido de muita utilidade na visualização de partes do corpo humano. Na verdade, certos métodos de geração de imagens como por exemplo o utilizado pela ciência da Ressonância Magnética Nuclear ("NMR") criam uma imagem digital, sendo fundamental, nesse caso, um terminal gráfico.

No Instituto de Física e Química de São Carlos, no Departamento de Física e Ciência dos Materiais, existe um grupo emergente em "NMR" que já conseguiu, inclusive, resultados notáveis. Surgiu, então, a necessidade de um terminal gráfico, o que motivou o presente trabalho.

Ao leitor interessado em se aprofundar no assunto deste capítulo, aconselham-se Foley (3), Waite (23) e Lerner (24).

CAPITULO II
O PROJETO - ARQUITETURA

II.1 ARQUITETURA GERAL

O projeto desenvolvido consta, logicamente, de duas unidades interdependentes que eventualmente podem ser ligadas separadamente. São elas:

1. Terminal Alfanumérico de Computador
2. Terminal Gráfico de Computador

Se os dois módulos forem usados juntos, como um terminal completo, o fluxo de informações pode ser visto na figura 18.

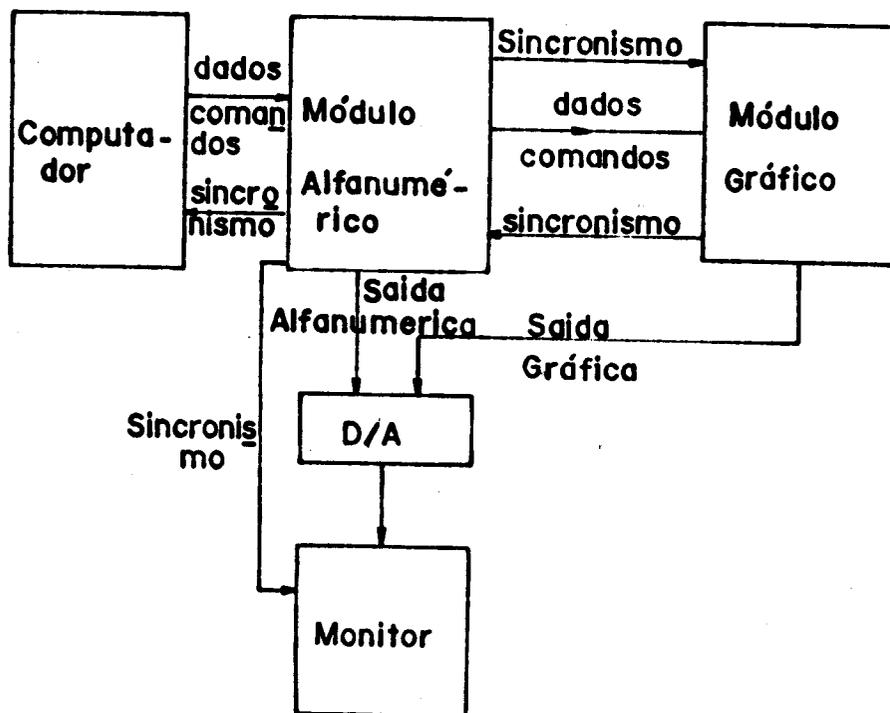


Fig. 18- Fluxo de dados no terminal completo

Repare que, nesse tipo de ligação, tem-se um terminal completo, que pode ser ligado a qualquer tipo de computador.

A comunicação entre o computador e o terminal é a tradicionalmente adotada em computadores. O terminal recebe caracteres serialmente (no caso, no padrão RS-232C), caracteres esses que podem ser comandos, inclusive para o módulo gráfico. Se não houver tempo para o processamento do caractere, é enviado um caractere de sincronismo (^X-OFF" ou ^S) desligando a comunicação e, quando o terminal estiver novamente disponível, é ligada novamente a comunicação ("X-ON" ou Q).

Os comandos sempre entram no terminal pelo módulo alfanumérico, inclusive comandos para o módulo gráfico. Se por acaso o módulo gráfico não tiver tempo para processar o comando, o módulo alfanumérico é avisado e envia um ^S para o computador.

Importante, porém, é notar que, a menos do sincronismo, o fluxo de dados só tem um sentido, qual seja:

```

computador ---> Alfanumerico ---> Grafico ---> Monitor
                |
                -----|
    
```

Para construção de terminais de baixo custo, pode-se eliminar o módulo gráfico e tem-se então a arquitetura simplificada da figura 19.

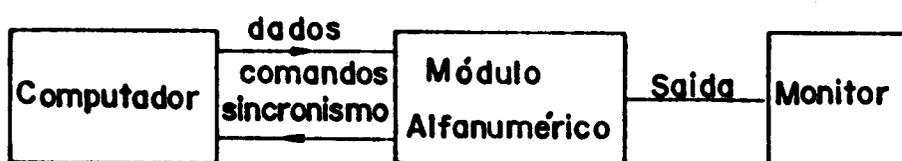


Fig. 19- Terminal apenas alfanumérico

A ligação dos módulos alfanumérico e gráfico como um terminal completo é bastante versátil, mas tem o defeito de apresentar uma taxa de transferência de dados do computador para os módulos muito baixa, principalmente devido à forma de comunicação serial que tem que ser adotada. Com pequenas modificações, entretanto, pode-se ligar os módulos a um duto paralelo 8-100 e, nesse caso, a arquitetura ficará como mostrado na figura 20.

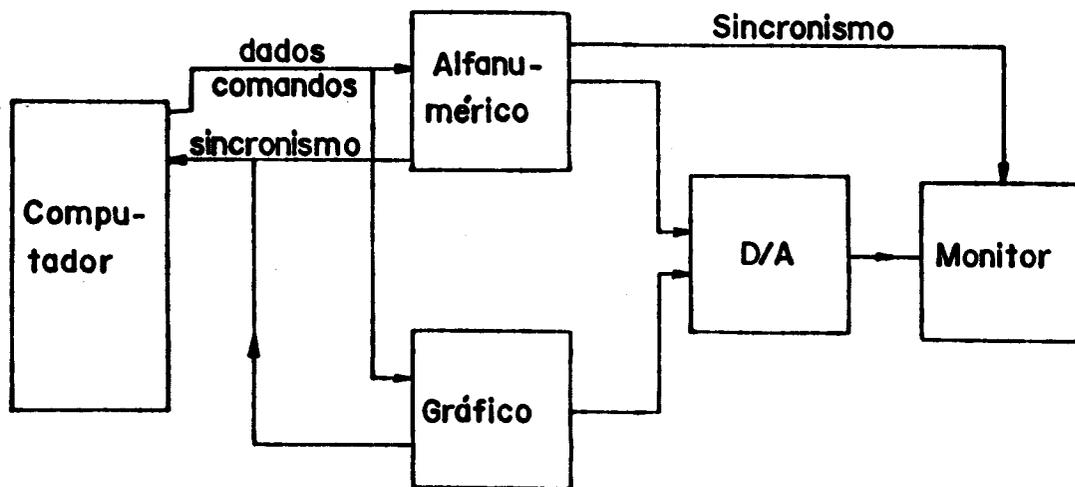


Fig. 20- Arquitetura para sistemas com terminal incorporado

Observe que a grande modificação agora é que os dados e comandos para o módulo gráfico não passam pelo módulo alfanumérico. A velocidade, aqui, aumenta muito, pois a comunicação é paralela. Em compensação, a versatilidade cai, pois o sistema gráfico só pode ser utilizado com o computador ao qual está incorporado.

A arquitetura de vídeo adotada é por varredura de rastros ('raster scanning', vide seção I.2 e I.8), já que tem como objetivo final a possibilidade de apresentação de imagens na tela.

Comparando as arquiteturas do terminal alfanumérico e do gráfico, verifica-se que existe uma diferença fundamental entre elas: no terminal alfanumérico, a memória de tela é justamente a memória de trabalho do processador. O controlador de vídeo deve então, sempre que necessário, requisitar do processador central os dados para a tela. Já no terminal gráfico, as memórias de trabalho e de tela são distintas. No primeiro caso, tem-se a vantagem de se escrever o dado somente uma vez enquanto no segundo, é necessária a transferência do dado, da memória de trabalho para a memória de tela, ou seja, são necessárias duas escritas. No segundo caso, no entanto, o problema de tempo de leitura da memória de tela é bem menos crítico.

II.2 O TERMINAL ALFANUMÉRICO

Por terminal alfanumérico, quer-se referir à parte do terminal especializada em caracteres e primariamente responsável pela comunicação homem-máquina.

O terminal alfanumérico aqui desenvolvido foi baseado numa nota de aplicação ('application note') (25) da companhia norte-americana Intel. A sua arquitetura básica está esboçada na figura 21. Cumpre ressaltar que se trata de uma arquitetura bastante tradicional de um terminal de vídeo.

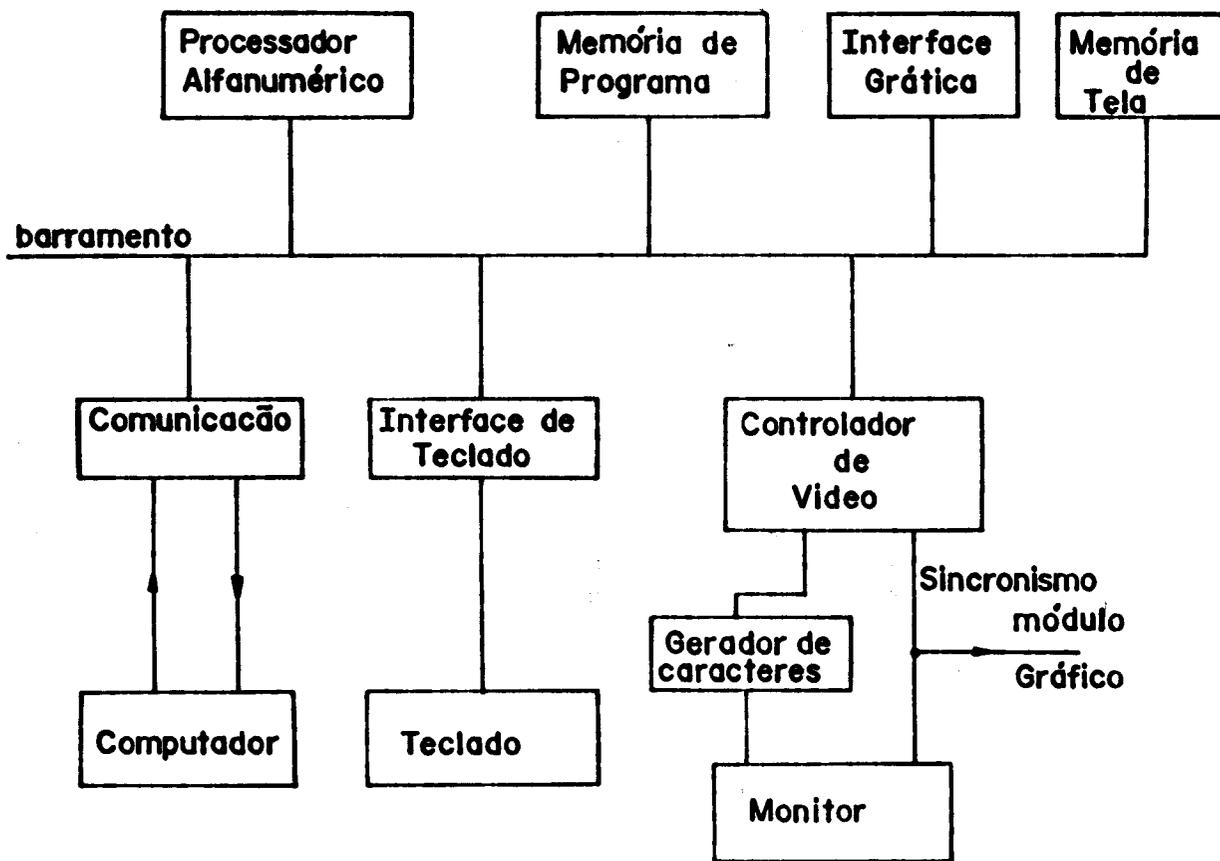


Fig. 21- Arquitetura do terminal alfanumérico

O processador alfanumérico tem basicamente 3 funções. Normalmente ele está pesquisando se um caracter foi recebido pela unidade de comunicação serial e, em caso afirmativo, tomando a ação apropriada. De tempos em tempos, porém, ele é interrompido pelo controlador de "CRT" para fornecer uma linha de caracteres (80 caracteres). Como uma linha de caracteres incorpora 10 varreduras horizontais do monitor, essa interrupção se dá de 10 em 10 varreduras horizontais (linhas do monitor). Além disso, o controlador também interrompe o processador a cada fim de tela para que este verifique se tudo está em ordem e se existe algum caracter a ser lido do teclado.

O controlador de 'CRT' mantém sempre duas linhas de caracteres armazenadas em 'buffers' internos (2x80 = 160 caracteres). Uma das linhas é a que está sendo enviada ao monitor e a outra é de reserva. Quando uma linha é enviada completamente para a tela, as funções dos 'buffers' se alternam e o controlador requisita ao processador uma nova linha de caracteres. O processador, então, lhe transmite 80 bytes que representam os códigos ASCII (7 bits) dos caracteres que compõem a futura linha.

O controlador, então, segundo uma temporização adequada, fornece para o gerador de caracteres o código ASCII do caracter a ser desenhado que, por sua vez, fornece o sinal de vídeo adequado para o monitor.

Além disso, o controlador de 'CRT' fornece os sinais de sincronismo, como retrazos horizontal e vertical e o sinal de supressão de vídeo, para o correto funcionamento do monitor.

O caracter ocupa um campo de 7x10 pontos, apesar de, por si, ocupar um campo de 5x7. Isso porque existem brancos ao redor do caracter para separá-lo dos outros.

O processador alfanumérico pode, também, enviar comandos para o terminal gráfico, através da interface gráfica.

II.3 TERMINAL GRAFICO

Por terminal gráfico, entende-se a parte do terminal de vídeo responsável pela criação de imagens na tela do monitor.

A arquitetura dessa parte pode ser vista na figura 22.

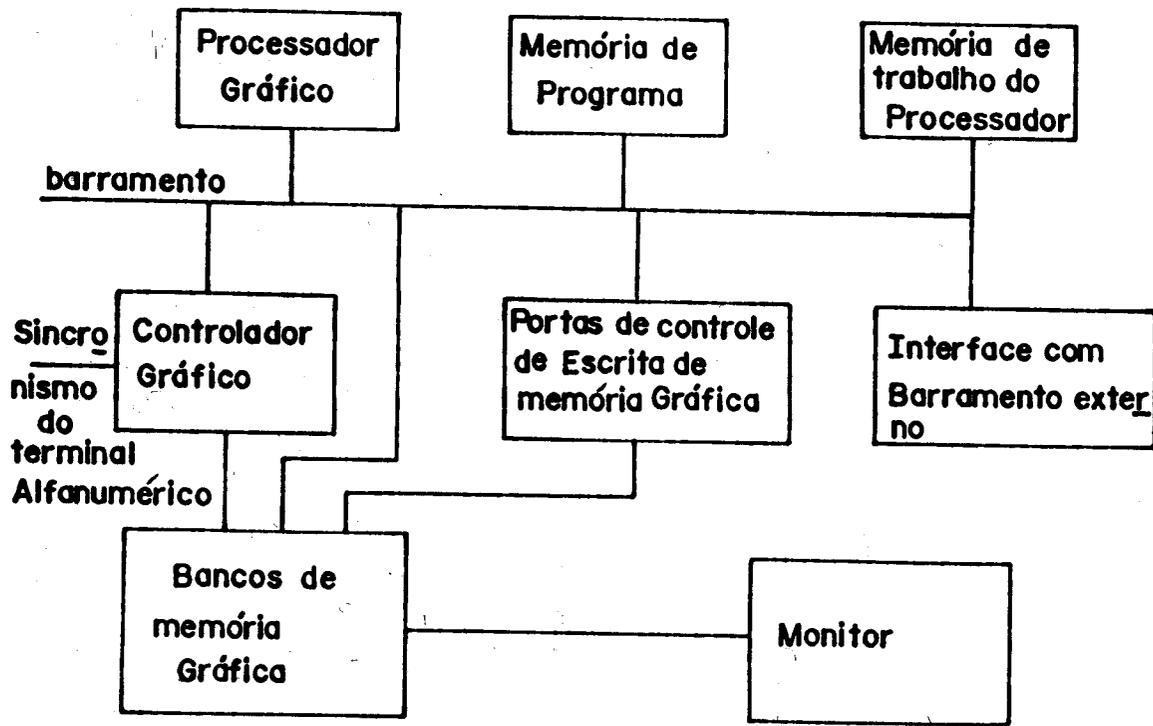


Fig. 22- Arquitetura do terminal gráfico

O processador gráfico lê um dado ou comando pela interface com o barramento externo, e executa o trabalho necessário. Quando o processador deseja modificar o conteúdo das memórias gráficas, ele primeiramente seleciona os bancos desejados e o modo de escrita, e depois espera o momento apropriado para modificar o conteúdo da memória (sempre durante o retrazo vertical) o que faz com o auxílio do controlador gráfico. Este último, como a memória utilizada é dinâmica, pode ser dividido como mostra a figura 23

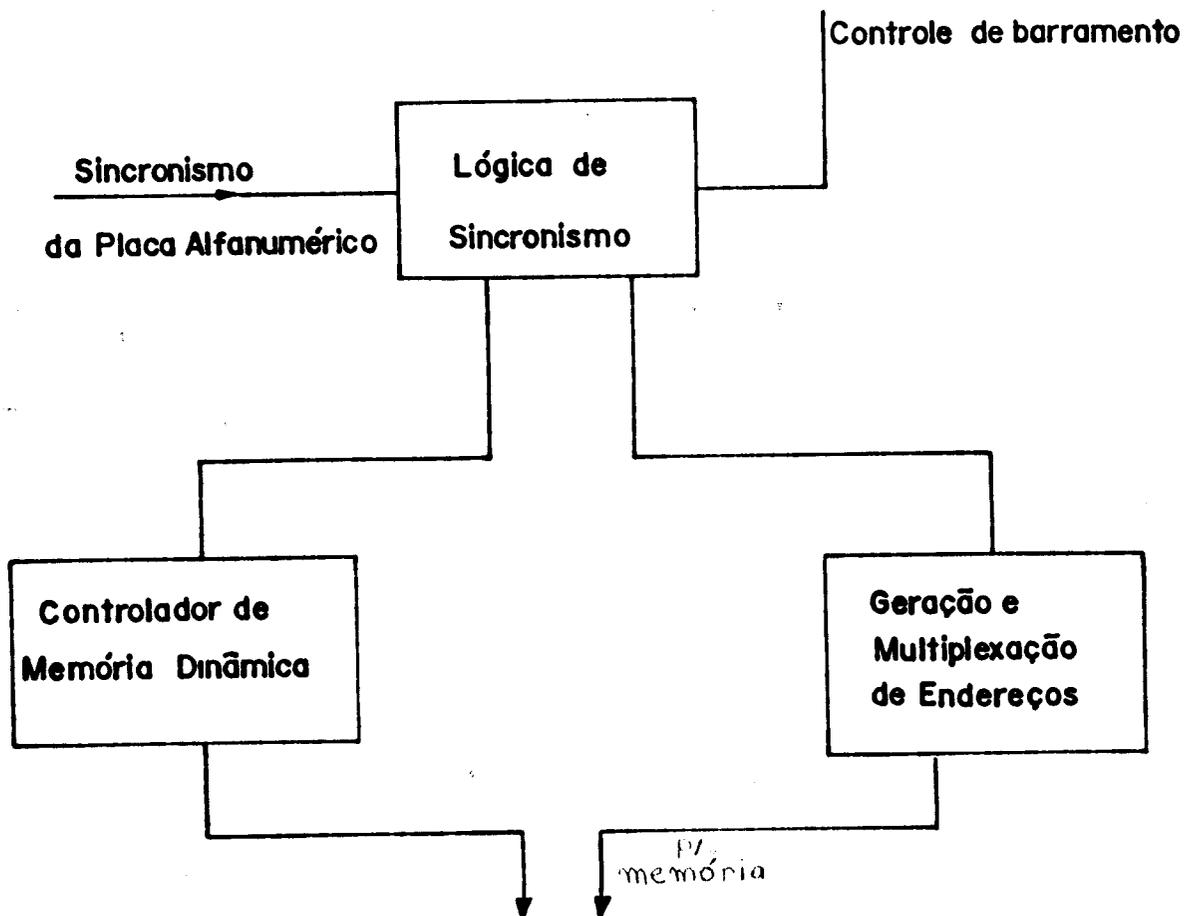


Fig. 23- O controlador gráfico

O controlador gráfico está constantemente acessando as memórias gráficas, que enviam os n bits referentes ao pixel para o monitor. Ele o faz sincronizado com a temporização produzida pela placa alfanumérica. Quando do tempo de um retraco, ele permite a escrita na memória, gerando os pulsos necessários sincronizados com o pulso de escrita do processador gráfico.

Cada banco de memória contém um bit de informação para cada pixel. Esses bits são arranjados em bytes de 8 bits. É possível utilizar-se de 1 a 8 bancos de memória.

A cada banco associa-se um bit do duto de dados da UCP gráfica. Assim, ao primeiro banco associa-se D0, ao segundo, D1, e assim sucessivamente. Também, cada banco tem uma habilitação para a escrita.

Existem quatro modos de escrita nos bancos de memória:

1. Para um determinado endereço de byte, escreve-se no byte todo o dado que está no bit do duto de dados associado àquele banco em todos os bancos habilitados.

Por exemplo, tem-se

Duto de dados = 0101 0101 byte endereçado = 2327
Bancos habilitados = todos

Então tem-se

banco 0	byte 2327	-->	1111 1111
banco 1	byte 2327	-->	0000 0000
banco 2	byte 2327	-->	1111 1111
			.
			.
			.
banco 7	byte 2327	-->	0000 0000

2. Para um determinado endereço de byte, escreve-se no byte todo o dado que está no duto de dados, em todos os bancos habilitados. Aqui o conceito de bit associado ao banco não importa.

Por exemplo, tem-se

Duto de dados = 0000 1111 byte endereçado = 4294

Bancos habilitados = todos com exceção do 1

Então tem-se

banco 0	byte 4294	-->	0000 1111
banco 1	byte 4294	-->	inalterado
banco 2	byte 4294	-->	0000 1111
	.		
	.		
	.		
banco 7	byte 4294	-->	0000 1111

3. Para um determinado endereço de byte, escreve-se somente no bit selecionado pela UCP o dado que está no bit do duto de dados associado àquele banco, em todos os bancos habilitados.

Por exemplo, tem-se

Duto de dados = 0011 0011 byte endereçado = 1000

Bancos habilitados = todos bit selecionado = D1

Então tem-se

banco 0	byte 1000	-->	IIII IIII
banco 1	byte 1000	-->	IIII IIII
banco 2	byte 1000	-->	IIII II0I

banco 3 byte 1000 --> IIII II0I

.

.

.

banco 7 byte 1000 --> IIII II0I

onde I significa inalterado.

4. Para um determinado endereço de byte, escreve-se somente no bit selecionado pela CPU o dado que está no duto de dados, em todos os bancos habilitados. Aqui o conceito de bit associado ao banco também não importa.

Duto de dados = XXX1 XXXX byte endereçado = 2000

Bancos habilitados = todos bit selecionado = D4

Então tem-se

banco 0 byte 2000 --> IIII IIII

banco 1 byte 2000 --> IIII IIII

.

.

.

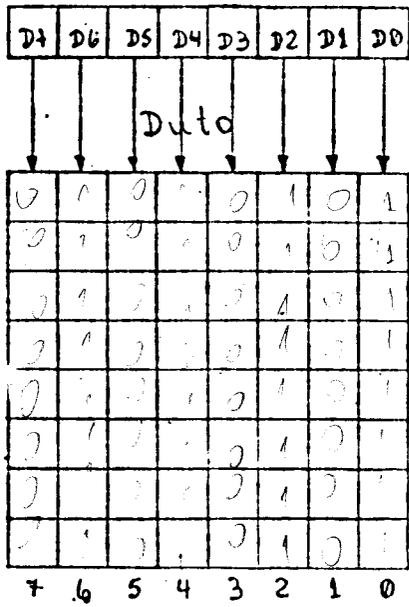
banco 7 byte 2000 --> IIII IIII

onde I significa inalterado e X significa não significativo.

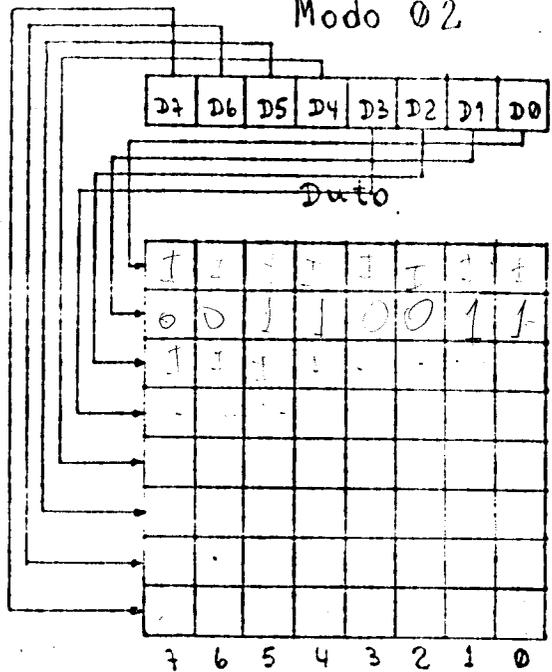
Observe que nos casos 1 e 2 pode-se modificar até 64 bits numa só escrita, enquanto que, nos modos 3 e 4, no máximo, modificam-se 8 bits.

Veja figura 24.

Modo 00

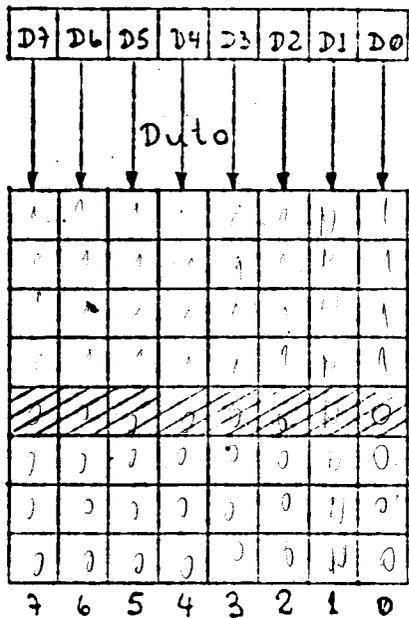


Modo 02



Modo 01

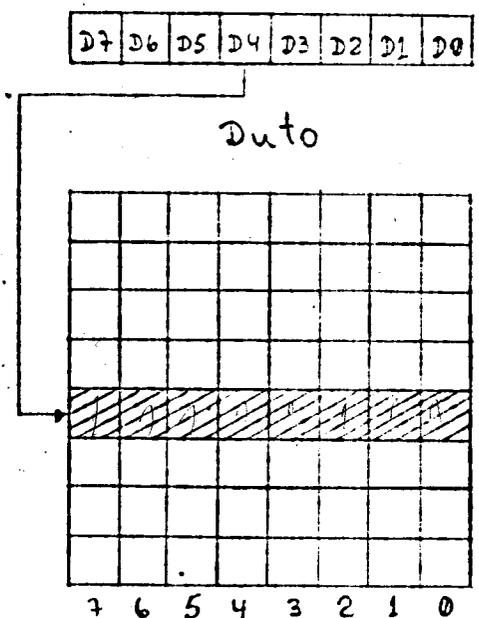
Bancos



Bancos

Modo 03

Bancos



Bancos

Fig. 24- Modos de escrita

CAPITULO III O PROJETO - "HARDWARE"

III.1 INFORMACOES GERAIS

O sistema é construído em placas do padrão S-100. Uma placa é utilizada pelo terminal alfanumérico enquanto ao menos duas são necessárias para o terminal gráfico. Assim, o terminal completo é constituído por um mínimo de três placas S-100. Os integrados utilizados são todos ou "MOS" ou "TTL".

III.2 TERMINAL ALFANUMERICO

O "hardware" aqui é bastante otimizado, visando ao custo mais baixo possível. Isso é conseguido se for utilizado um pequeno número de componentes. Nesse sentido, não foram usados "buffers", o que diminuiu bastante o número de integrados. Também, a filosofia de se usar a mesma memória tanto para memória de trabalho do processador quanto para memória de tela, também contribuiu para a citada redução de integrados. De fato, se fossem usadas memórias separadas, considerável quantidade de "hardware" adicional se faria necessário. Veja-se, por exemplo, o terminal gráfico (seção III-3), que contém todo um "hardware" de sincronização, para permitir o acesso do processador gráfico a ambas as memórias (de tela e de trabalho).

Para as explicações que se seguem, refira-se sempre à figura 21. É interessante também, ter à mão a referência (26).

O terminal alfanumérico contém um processador próprio que é o microprocessador 8085A da indústria americana INTEL. O programa que o 8085A executa está numa 'EPROM' de organização 2Kx8, a 2716. A memória de trabalho ('RAM') é constituída por 4 pastilhas 2114, cada uma 1Kx4, organizadas de forma a fornecer 2K bytes de memória. É utilizada uma pastilha 8212 ('Latch') que faz a demultiplexação de endereço e dado do 8085A.

A interface serial é implementada através de uma 'USART' 8251A que conta com uma 8253 para geração do relógio de transmissão e de recepção.

A unidade central, o 8085A, trabalha com um relógio de 3.072MHz, o que faz com que todas as memórias e periféricos trabalhem sem a necessidade de estados de espera.

O relógio do 8253 é o do 8085A dividido por dois através de um biestável, o 7474, o que dá 1.536 MHz. Esse relógio é, então, internamente dividido por quantias apropriadas, dependendo do "baud rate" que se desejar. O terminal pode funcionar em 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600 seleccionáveis por "Jumper".

As memórias 2716 e 2114, o 8253 e o 8251A estão todos em espaço de memória do processador. O endereçamento é o seguinte (determinado por uma 'PROM' TTL 74S188):

2716 == 0000H -> 07FFH

4x2114 == 0800H -> 0FFFH

Quanto ao 8253 e 8251 tem-se

8251 == 101X XXXX XXXX XXXXB

8253 == 011X XXXX XXXX XXXXB

ou seja, 8251 é endereçada sempre que A15 está em um. Para não selecionar os outros dispositivos, deve-se manter em nível alto o endereço A13, e para não selecionar o 8253 deve-se zerar o A14. A mesma análise vale para o 8253.

Repare que os endereços do 8251 podem ser AXXXH ou BXXXH enquanto que os do 8253 podem ser 6XXXH ou 7XXXH. Observe também que o endereço A0 é lido ao 8251 para selecionar controle ou dado (0-->dado;1-->controle), e que A0 e A1 são lidos no 8253 para selecionar contadores internos.

A interface de teclado é implementada através de 3 portas de entrada e saída. No caso, foi utilizada a 8255A da INTEL. Duas de suas portas são entradas e uma é saída. A porta C é uma entrada de onde são lidos os 'Jumpers' de 'baud-rate' e as teclas especiais 'shift', 'caps-lock' e 'control'. Os bits da porta C são:

PC0,PC1,PC2,PC3 - 4 bits de Baud Rate

PC4 - Caps lock (Ativo 0)

PC5 - Shift (Ativo 0)

PC6 - Control (Ativo 0)

O "baud rate" é dado por:

PC3,PC2,PC1,PC0	Baud rate
-----	-----
0 , 0 , 0 , 0	75
0 , 0 , 0 , 1	110
0 , 0 , 1 , 0	150
0 , 0 , 1 , 1	300
0 , 1 , 0 , 0	600
0 , 1 , 0 , 1	1200
0 , 1 , 1 , 0	2400
0 , 1 , 1 , 1	4800
1 , 0 , 0 , 0	9600

Para um caracter ser lido do teclado, é feita uma varredura na porta A do 8255A, que representa as linhas de varredura (ativo baixo), e é lida a porta B, cujas linhas representam os retornos (ativo baixo). Através da matriz definida na figura 25 é descoberto o caracter.

Cumpra notar que é feito o "debounce" do teclado por "software". Assim, como o processo de leitura do teclado é repetido de tela em tela, e, como isso representa um tempo de 16.67ms, deve-se manter a tecla pressionada por pelo menos 33ms.

No espaço de endereçamento do microprocessador, para se acessar o 8255A, deve-se levar para nível lógico 1 os bits 12 e 11 e manter baixos 15, 14 e 13. Os endereços A0 e A1 determinam, por sua vez, a porta interna acessada (A, B, C, ou controle). Os endereços usados pelo "software" são:

MATRIX TECLADO DIGIPONTO

Linhas de Retorno

Linhas de
Varredura

y ⁰	X ₀ ,KP	X ₁ SP	X ₂ ESC	X ₃ I	X ₇ 9KP	X ₆ 6KP	X ₅ 3KP	X ₄ BS
y ¹	.KP	A	Q	2	8KP	5KP	2KP	Z
y ²	∅ _{KP}	S	W	3	7KP	4KP	1KP	X
y ³	DEL	D	E	4	·	CR	LF	C
y ⁷	TAB	F	R	5	-	@	:	V
y ⁴	? /	6	T	6	≡	P	+	B
y ⁵	>	H	Y	7	O	O	L	M
y ⁶	<	J	U	8	9	I	K	N

Obs: Foi mantida a
convenção interna-
cional

Fig. 25- Matriz de decodificação de teclado

Porta A - 1800H

Porta B - 1801H

Porta C - 1802H

Controle - 1803H

A parte mais delicada, porém, é a parte de controle de vídeo. O controlador utilizado é o 8275 da INTEL. Essa pastilha gera todos os pulsos de sincronismo necessários para o funcionamento do monitor. Ele tem memória interna suficiente para armazenar duas linhas completas de 80 caracteres. Quando uma linha acaba de ser mostrada na tela, o 8275 começa a mostrar a outra e imediatamente requisita do processador os próximos 80 caracteres da linha seguinte. Isso acontece a cada 617

microssegundos. Claramente, então, o processador deve ser capaz de fornecer os 80 bytes num tempo menor que esse. De fato, se é desejado que o processador faça algo mais do que simplesmente transferir dados para o controlador, espera-se que o tempo de transferência seja bem menor do que 617 microssegundos.

Na verdade, o 8275, para máxima eficiência, requisita um acesso direto à memória quando ele precisa de dados. Porém, isso exige uma pastilha de controle de acesso direto à memória e toda uma lógica de apoio, o que aumenta e encarece o projeto.

Foi utilizado aqui, então, um truque que evita a necessidade do acesso direto com pouca perda de eficiência. Utilizou-se um sistema convencional de interrupção e, assim, o 8275 simplesmente interrompe o processador central (8085) e dele requisita os caracteres. Para otimizar o processo de transferência, aciona-se um bit especial de transferência e executam-se 40 instruções POP. Como cada instrução POP acessa 2 bytes, com as 40 instruções ter-se-ão acessado os 80 caracteres exigidos. Esses bytes vêm para o processador, mas isso não interessa, pois o processador não necessita deles. Ao mesmo tempo, porém, como o bit de transferência está ativo, o 8275 vai colhendo esses dados como se os estivesse recebendo de um controlador de acesso direto à memória ("DMA"). O selecionador de endereços (74188) será, então, simultaneamente a habilitação de pastilha para as 2114 e para o 8275.

Sendo as instruções POP relativamente rápidas (10 ciclos de relógio, o que dá 5 ciclos por byte), a eficiência do processo é muito boa. Com pastilhas de "DMA" convencionais, conseguir-se-ia, na melhor das hipóteses, fazer a transferência

a 4 ciclos por byte.

No momento correto (dado por um relógio de caracteres), o 8275 envia para um serador de caracteres, o código ASCII do caracter a ser escrito, e também a linha do caracter. A saída do serador é carresada num registrador de deslocamentos, o 74166 que, segundo um relógio de ponto (o relógio de caracteres nada mais é do que o de ponto dividido por 7, que é a largura do caracter), envia o ponto para o vídeo.

O 8275 gera os sinais de retraco horizontal (HRTC), retraco vertical (VRTC) e supressão de vídeo (VSP). Esses sinais, porém, são atrasados de um caracter para haver sincronismo com o sinal de vídeo produzido pelo 74166, já que esse, devido ao atraso no serador de caracteres, está sempre atrasado de um caracter.

O 8275 no espaço de endereçamento está assim colocado:

0001 00XX XXXX XXXXB - Para escrita de comandos ou parâmetros.

(A0=0 parâmetros, A0=1 comandos)

0001 01XX XXXX XXXXB - Para leitura de "status"

O esquema básico do terminal de vídeo bem como os "timings" podem ser encontrados na referência (25).

III.3 O TERMINAL GRAFICO

III.3.1 PRINCIPIOS BASICOS

O terminal gráfico é constituído de duas unidades interdependentes: o processador gráfico e o controlador gráfico. Como já foi visto no capítulo II, a memória de trabalho do processador gráfico e memória de tela são distintas.

Enquanto o processador trabalha na "sua" memória, o controlador fica constantemente acessando a memória de tela e enviando o seu conteúdo para o monitor. Para modificar o conteúdo da tela, o processador deve esperar o momento de inatividade do controlador (retrazos). Repare que essa modificação é feita com auxílio do controlador.

Essa arquitetura foi escolhida, do ponto de vista do "hardware", devido ao problema do tempo. De fato, o processador é lento e não conseguiria passar os dados para o circuito de escrita na tela no tempo devido. Repare que, têm-se, apenas, 64 microssegundos para se enviar ao monitor 512 bits, ou seja, 64 bytes, o que é completamente impossível para o microprocessador utilizado (8085A). Com duas memórias, têm-se duas unidades trabalhando em paralelo: o processador e o controlador. A grande desvantagem, no entanto, é o fato de que os dados devem ser transferidos de uma memória para a outra (da memória de trabalho para a de tela) sempre que se desejar modificar a tela.

O controlador utiliza-se de integrados "TTL" do tipo S para gerar os pulsos de controle da memória. Esses integrados foram escolhidos devido ao problema de tempo que é crítico, como será visto mais adiante.

A cada conjunto de memória de tela, suficiente para fornecer um bit de informação para cada "pixel", dá-se o nome de banco. A memória utilizada nos bancos foi a memória dinâmica. O motivo é bem simples: custo. De fato, a parte do terminal que contém maior número de integrados é o conjunto de bancos de memória de tela, principalmente se for utilizado o número máximo de bancos possíveis (8). Assim, cada banco deve ser feito o

mais barato possível. Escolheu-se, então, a pastilha de memória dinâmica 4116 por seu preço ser o mais baixo encontrado no mercado. Essa escolha, é claro, implica numa maior complexidade do controlador gráfico. A alternativa seria utilizar memórias estáticas, o que simplificaria bastante o projeto, mas o encareceria demais (e ainda não esquecer que a memória estática não é tão compacta quanto a dinâmica, o que levaria a um número bem maior de componentes).

Apesar de, como foi dito, dever-se fazer cada banco o mais barato possível, foi utilizado, em cada banco, uma considerável quantidade de integrados lógicos para se ter versatilidade nos modos de programação. Isso se faz necessário para se otimizar, ao máximo, o processo de modificação da tela que, como foi visto, é a grande desvantagem da arquitetura escolhida. Repare que a versatilidade dos modos de programação é tal que se pode modificar, em casos extremos, 1 ou 64 bits de uma só vez dentre os 1024000 bits possíveis (no caso de 8 bancos).

O processador gráfico, ao contrário do alfanumérico, contém "drivers" nos endereços e nos dados. Isso foi feito tendo em vista futuras expansões. A memória de trabalho do processador é estática, pois esta é mais barata que a dinâmica quando em pequena quantidade (a memória estática não precisa de controlador).

Para se fazer a multiplexação necessária à memória dinâmica, utilizou-se "drivers tri-state". Multiplexadores seriam mais rápidos e baratos, mas não forneceriam a mesma corrente.

III.3.2 DESCRICAO

Como foi visto na seção II.1, o terminal gráfico pode ou não ser ligado, e se o for, pode ser-lo de duas maneiras.

1. Ligado diretamente num duto S-100 dum computador, quando então a comunicação é paralela
2. Ligado ao terminal alfanumérico. Neste caso tudo que o computador lhe enviar, deve passar antes pela placa alfanumérica. A comunicação computador - terminal alfanumérico é então serial enquanto a alfanumérico - gráfico é paralela.

Para a discussão que se segue, refira-se sempre a figura 22 da seção II.3, que dá a arquitetura do terminal gráfico.

A entrada de dados no terminal gráfico é, assim, sempre paralela e é feita através de uma 8212. Quando se quer escrever algo nela, é preciso fornecer um pulso para o pino 11, ativo alto, de duração mínima de 30 nanossegundos. Assim, a 8212 deve estar no espaço de endereçamento, ou do terminal alfanumérico, ou do processador central. Antes, porém, de se escrever algo, é necessário verificar se o terminal gráfico já leu o dado anterior. Assim, deve-se ler o 'status', que nada mais é que o pino 23 do 8212 convenientemente passado por uma porta com saída do tipo 'tri-state'. Se for 1, o 8212 está esperando um próximo dado ou comando. Essa informação é lida ao bit D0 do duto de dados S-100 e para ser acessada, deve-se tirar a porta de 'tri-state'. Vê-se, assim, que a comunicação entre qualquer processador e o terminal gráfico existe apenas a reserva de duas posições no espaço de endereçamento do

Processador, uma apenas de escrita para se passar o dado (ou comando) e outra apenas de leitura para verificar se já é permitido mandar outro dado (ou comando).

Uma sugestão para a ligação da 8212 ao processador é mostrada na figura 26.

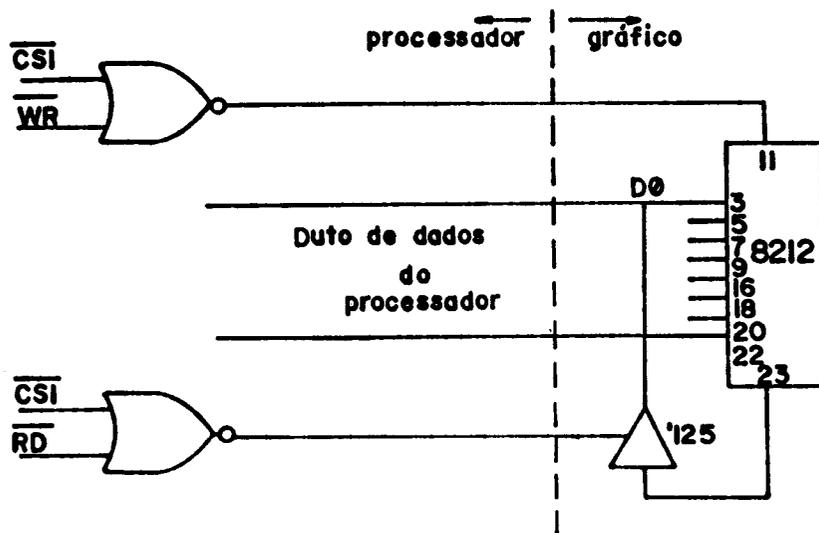


Fig. 26- Interface Processador-terminal gráfico

A leitura do dado da 8212 pelo processador gráfico é feita por 'polled operation'. Quando o dado é lido, automaticamente a 8212 torna inativo o pino 23. A pastilha 8212 está no espaço de entrada e saída do 8085A, na posição 80H.

Além dessa 8212, mais duas estão colocadas no espaço de entrada e saída do 8085A. São controles para a escrita na memória gráfica. Uma delas seleciona quais dos bancos de memória estão habilitados. Pode-se, assim, ter 8 bancos de memória gráfica. O endereço dessa 8212 é 88H. A outra 8212 determina o modo de escrita nas memórias gráficas, e também faz a seleção do bit. Aqui o endereço é 90H.

O selecionamento no espaço de entrada e saída é feito por um decodificador 74138.

No espaço de memória, tem-se lugar na placa para 3 'EPROMs' do tipo 2732 o que resulta em 12K bytes de programa básico de controle do terminal. Também, têm-se 4 2114 resultando em 2K bytes de memória de trabalho. Observe que essa memória é de trabalho para o processador, e não está relacionada com os bancos de memória gráfica.

A seleção desses dispositivos é feita também por um decodificador 74138 auxiliado por um 74139 e os endereços são os seguintes:

3x2732 (12K) == 0000H --> 2FFFH

4x2114 (2K) == 7000H --> 77FFH

Os sinais de controle RD*, WR*, IO/M*, são decodificados em MRD*, MWR*, IOR*, IOW* por uma pastilha 74LS257.

O duto de endereços e o de dados contém 'buffers', de modo que não há qualquer problema em termos de carga no circuito. Os sinais RESET OUT, CLK OUT e HLDA também contém 'buffers'. O processador conta ainda com dois 'drivers' 74LS245 para fazer a multiplexação de endereços necessária aos bancos de memória gráfica.

A outra parte da placa gráfica, o controlador gráfico, é a parte do terminal responsável por colocar o valor do pixel na tela no momento correto. Isso envolve, além de sincronismo com a placa alfanumérica (controle alfanumérico = 8275), o controle serial da memória dinâmica.

Como é sabido, a memória dinâmica precisa ser refrescada a cada 2ms pelo menos. Porém, cada leitura já automaticamente refresca a linha inteira. Assim, a 4116 que é a memória utilizada no terminal por seu baixíssimo preço por bit, é uma memória dinâmica organizada como uma matriz 128x128. Nela, uma leitura automaticamente refresca 128 bits.

A frequência de refrescamento deve ser, portanto, maior que uma vez a cada

$$\frac{2\text{ms}}{128} = 0.0156\text{ms} = 15.6 \text{ microssesundos}$$

Repare porém, que, ao contrário de memórias principais de computador, tem-se, nesse caso, a certeza de que todas as posições serão lidas num determinado tempo. Assim, se esse tempo for curto o suficiente, não é necessário o refrescamento. E de fato esse o caso, pois, se for arranjado o "hardware" de tal forma que o endereço das linhas seja o menos significativo, o que vale dizer que eles mudam mais rapidamente, então, tem-se que 64 posições serão lidas por linha horizontal (64x8=512 pontos). Como cada linha horizontal leva aproximadamente 60 microssesundos tem-se:

$$\frac{128}{64} = 2 \text{ linhas horizontais}$$

$$2 \times 60 = 120 \text{ microssesundos}$$

Ou seja, o refrescamento total da memória é feito em 120 microssegundos. Como no retraco vertical não existe leitura deve-se ter

$$120\text{us} + 1.3\text{ms} \approx 1.4\text{ms}$$

Pois 1.3 ms é o tempo aproximado do retraco vertical.

Como

$$1.4\text{ms} < 2\text{ms}$$

não é necessário refrescamento especial (27).

No controlador gráfico, tem-se um contador 74S163 para transformar o relógio de ponto em relógio de byte. Assim, a cada acesso à memória, os valores de 8 pixels são acessados. O tempo de acesso à memória deve ser, portanto, no máximo, oito vezes maior que o tempo de um pixel.

Trabalhando-se a 13 MHz no relógio de pixel tem-se

$$\frac{1}{13 \times 10^6} \approx 77 \text{ nanossegundos}$$

$$8 \times 77 = 616 \text{ nanossegundos}$$

O ciclo da memória deve ser portanto menor que 616 ns. Como a memória 4116 mais lenta (da Motorola) apresenta um ciclo de 480ns, esta pode ser utilizada. No protótipo desenvolvido, foi utilizada a MK4116N-3 da MOSTEK com ciclo de 375 ns.

O contador 74S163, já citado, deve então dividir o relógio de pixel por 8, e, então, requisitar uma leitura na memória. Esse circuito é mostrado na figura 27.

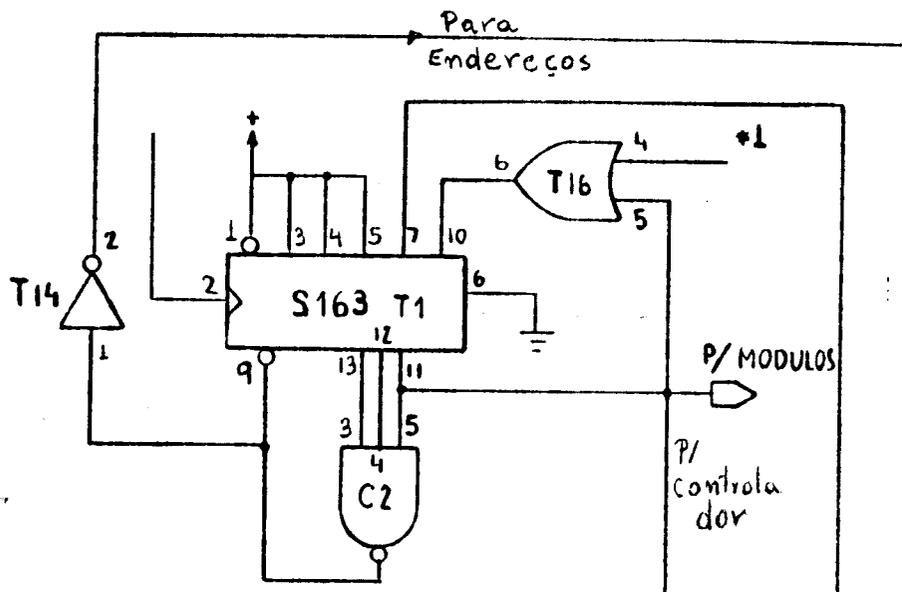


Fig. 27- Geração de relógio de requisição à memória

O processo é interrompido quando se deteta que o caracter de número 64 já foi mostrado. Então, espera-se a próxima linha para habilitar-se o contador novamente. O circuito que faz isso está na figura 28.

O acesso a memória é requisitado para um circuito que gera os sinais apropriados. Nele existe um relógio de período ajustável, que dá a referência de temporização. Um contador 74S163 conta 8 períodos para completar um ciclo de acesso. Sendo o período de 68ns, tem-se $8 \times 68 = 544\text{ns}$.

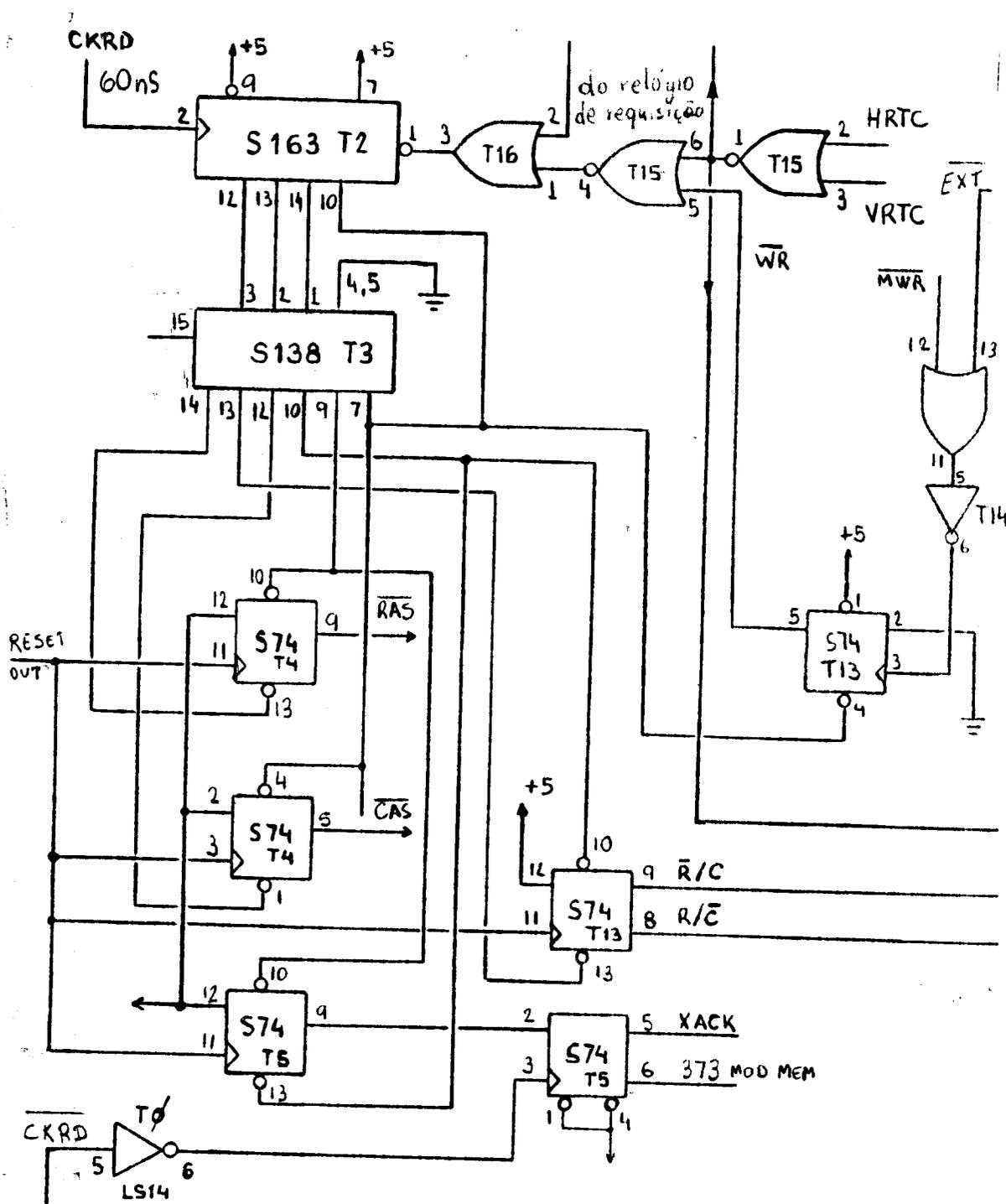


Fig. 29- Controle de memória dinâmica

- tempo entre estabilização de endereço de coluna e
ativação de CAS*:

$$1 \times 68 \text{ ns} = 68 \text{ ns}$$

- duração de RAS*

$$5 \times 68 \text{ ns} = 340 \text{ ns}$$

- duração de CAS*

$$4 \times 68 \text{ ns} = 272 \text{ ns}$$

- tempo entre ativação de RAS* e ativação de XACK*

$$5 \times 68 + \frac{68}{2} = 374 \text{ ns}$$

- tempo entre ativação de CAS* e ativação de XACK*

$$3 \times 68 + \frac{68}{2} = 238 \text{ ns}$$

- tempo entre ativação de RAS* e inativação do
endereço de coluna

$$4 \times 68 \text{ ns} = 272 \text{ ns}$$

- tempo de pré-carregamento de RAS*

$$3 \times 68 \text{ ns} = 204 \text{ ns}$$

- tempo entre ativação de RAS* e inativação de CAS*

$$6 \times 68 \text{ ns} = 408 \text{ ns}$$

- tempo entre ativação de RAS* e ativação de CAS*

$$2 \times 68 = 136 \text{ ns}$$

- tempo entre ativação de CAS* e inativação de RAS*

$$3 \times 68 = 204 \text{ ns}$$

- tempo de inativação de CAS* para nova ativação de RAS*

$$2 \times 68 = 136 \text{ ns}$$

- tempo de sustentação de endereço de coluna depois de CAS*

$$2 \times 68 = 136 \text{ ns}$$

Por uma comparação com os valores permitidos, vê-se que esses tempos são compatíveis com a MCM4116A-30, que é a mais lenta memória dinâmica 16Kx1 da Motorola (28).

Algumas observações em relação aos tempos anteriores devem ser feitas:

1. Se for variado o período do relógio de memória de 68ns para um T qualquer, para se saber os novos tempos basta substituir nas fórmulas acima o 68ns por T.
2. Os cálculos acima mostram que para a MCM4116-30 existe grande folga na maioria dos tempos. Não se deve usar, entretanto, um período menor que 68ns pois, para o tempo entre ativação de CAS* e inativação de RAS* (tRSH), têm-se 204ns com período de 68ns, e deve-se ter esse tempo maior que 200ns.
3. O projeto é totalmente síncrono (29), (30) devido à utilização de contadores síncronos (os 74S163). Assim, nenhum tempo é perdido em sincronização, apesar de os relógios de pixel e de memória serem completamente independentes. Assim, se o relógio de pixel está a 13MHz, e o período mínimo da memória é 68ns, pode-se ter o relógio de memória em qualquer frequência fm tal que:

$$\frac{1}{68\text{ns}} = 14.71\text{MHz} > f_m > 13\text{MHz}$$

4. Se a memória for mais rápida, obviamente pode-se usar uma frequência maior. Em geral, como o tempo citado na observação 2 é o mais crítico entre todos os tempos, para saber o período mínimo basta dividir aquele tempo por 3.

Se, por exemplo, tiver-se a MCM4116-25, tem-se

$$t_{RSHmin}=165ns \Rightarrow T_{min}=55ns$$

Assim f_m poderá ser tal que

$$13MHz < f_m < \frac{1}{55} = 18MHz$$

Esta não é, todavia, resra geral.

No caso da MK4116N-3 utilizada no protótipo tem-se

(31):

$$t_{RSH}=135ns \Rightarrow T_{min}=45ns$$

$$13MHz < f_m < \frac{1}{45} = 22.2MHz$$

mas o tempo mínimo de ciclo deve ser 375ns $\Rightarrow f_m < 21.3MHz$.

5. Nos tempos acima não se consideraram os atrasos das portas. Na verdade, mesmo com eles, não haverá problemas. Os extensos cálculos são aqui omitidos.

O controle de acesso à memória dinâmica também permite que o processador escreva na memória, apesar de que somente num período de atraso.

A maioria dos tempos no caso de escrita são os mesmos que no caso da leitura. A principal diferença é que agora deve-se agir sobre o pino de WR* da memória. Quem fornece o sinal WR* é o próprio processador. Os bancos de memória gráfica estão localizados nos 16K bytes superiores do espaço de endereçamento da unidade central gráfica. Assim, eles só serão acessados quando A15 = 1 e A14 = 1.

Uma vez requisitada pelo processador uma escrita à memória, no máximo um período de relógio de memória depois se dará o início do ciclo de escrita com a ativação do RAS*.

Deve-se ter certeza, aqui, de que o processador deverá manter o pulso de escrita, os endereços e também os dados constantes por um tempo suficiente. No caso da memória mais lenta (MC4116A-30), este tempo vale, para o caso de 68 ns:

$$68 \text{ ns} + 2 \times 68 \text{ ns} + 100 \text{ ns} = 304 \text{ ns}$$

onde 100ns é o tempo de sustentação de dados e endereços

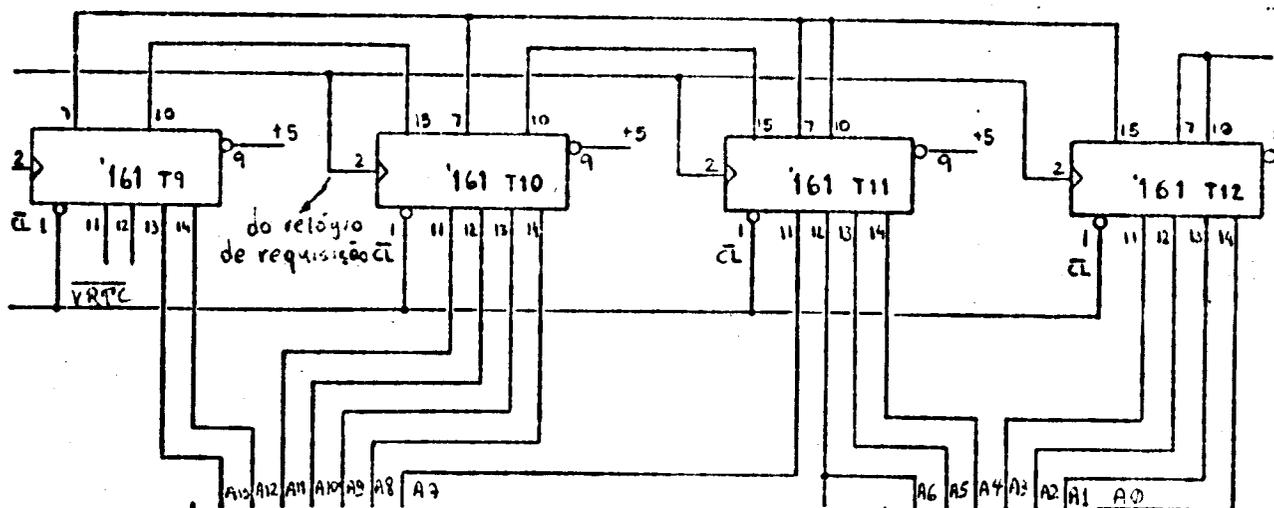
2 x 68ns é o tempo de ativação de CAS*

68ns é o tempo de ativação de RAS*

Como se está utilizando de um relógio no processador gráfico, de 3.000 MHz, o pulso de escrita e, conseqüentemente, os dados e endereços serão mantidos, pelo menos (26), por 420ns. Sendo $420 \text{ ns} > 304 \text{ ns}$ não haverá problema. Também o tempo de estabilização de endereços é compatível com as especificações das memórias.

A multiplexação de endereços, tanto na escrita como na leitura, é feita por 2 circuitos 74LS245. No controlador gráfico (ou seja, na leitura), os endereços são gerados por 4 contadores 74161. A geração de endereços é a única lógica do controlador gráfico que não foi feita com integrados tipo S. O motivo é que, aqui, não há necessidade de grande velocidade, já que a mudança de endereço, é requisitada por um período de relógio de pixel antes da requisição de acesso à memória. Sendo assim tem-se pelo menos $2 \times 77 \text{ ns}$ ('dot clock') = 154ns para a mudança de endereço o que é facilmente conseguido com circuitos 'TTL' comuns.

Aqui, o pior caso em termos de tempo, ocorre no último pixel da linha, quando, então, o endereço deve comutar e inibir a continuidade do relógio de pixel no contador (T1). Mesmo aqui contudo, não é necessário utilizar-se um integrado mais veloz. É ilustrativo mostrar como isso é calculado (esse tipo de cálculo de tempos, considerando-se atrasos em portas, teve que ser feito várias vezes durante o projeto. Como já foi dito, porém, na observação 5 anterior, não é possível reproduzi-los aqui). Para acompanhar os cálculos, o leitor deve observar as figuras 27 e 30 que mostram, respectivamente, a geração do relógio de acesso à memória e o circuito de geração de endereços.



Fis. 30- Geração de endereços

1. Vem o relógio de ponto --> A saída do 74S163 (T1) muda no máximo em 15ns (garantido pelo fabricante) (32).
2. A saída mudando, o sinal para o relógio dos contadores de endereço (4 x 74S161, T9 a T12) passa por 2 portas simples do tipo S (T2 e T14) => $2 \times 6 \text{ ns} = 12 \text{ ns}$ (no pior caso).
3. Os 74161 comutam no máximo em 30 ns.
4. Os bi-stáveis (T6), sendo do tipo S, respondem em 9 ns o que deve ser somado a 6 ns de um inversor S (T14), o que dá um resultado de 15 ns.
5. O sinal passa por duas portas S (T15 e T16) para então desativar T1 => $2 \times 6 \text{ ns} + 10 \text{ ns} = 22 \text{ ns}$.

Tudo isso deve ocorrer em menos de dois períodos de relógio. Assim:

$$2 \times 77 = 154 \text{ ns} > 15 + 12 + 30 + 15 + 22 = 94 \text{ ns}$$

Como a desigualdade se verifica, está tudo correto.

O circuito de endereços para a escrita está diretamente ligado ao duto de endereço do processador, e pode ser visto na figura 31.

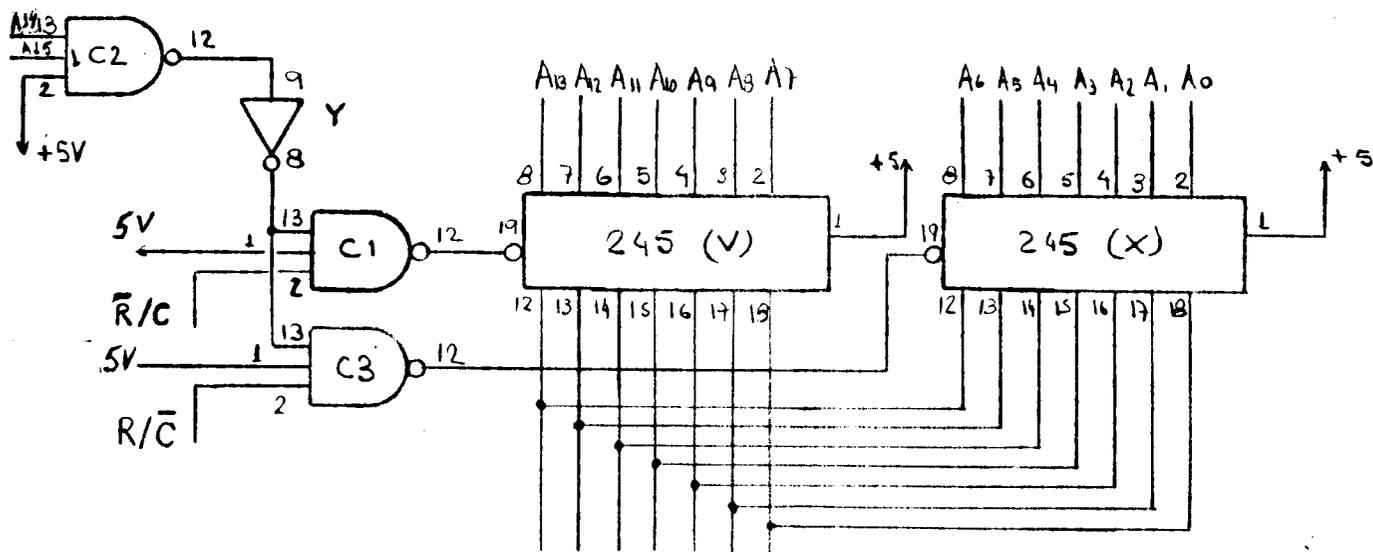


Fig. 31- Circuitos de endereço na escrita

Os bancos de memória gráfica, como já foi dito, são constituídos de 8 peças de memória dinâmica 4116 de organização 16k x 1 bit.

Além das memórias, o banco contém uma lógica para determinar de que modo será feita a escrita pelo processador e, se necessário, em que bit. Como já foi visto na seção II.3, existem 4 modos de escrita, que são, portanto, selecionados por dois bits que se encontram na 8212 (S) da unidade central.

bits são D0 e D1. Também nessa 8212, faz-se a seleção do bit a ser modificado. Aqui precisa-se, é claro, de 3 bits e então tem-se D2, D3 e D4. Repare que D5, D6 e D7 da 8212(S) estão livres para futuras expansões.

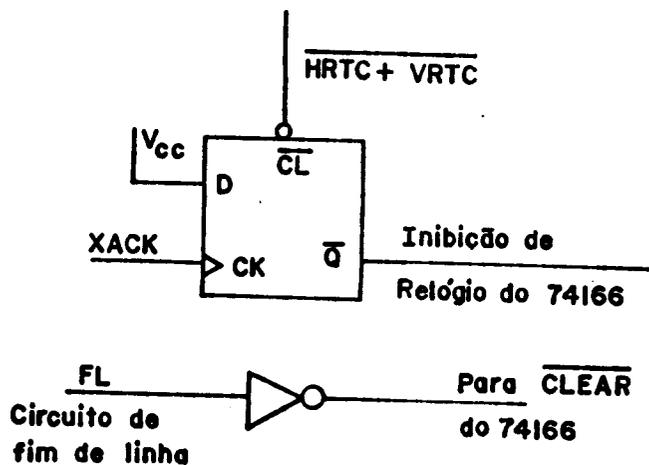
A decodificação do bit a ser modificado é feita por um 74138, cujas saídas entram na lógica que fornecerá o WR* para as 4116.

Os bancos de memória estão ligados, também, diretamente ao duto de dados do processador gráfico.

As saídas das memórias (8 saídas) são ligadas a um 74373 que nada mais é do que um 'tampão' colocado para facilitar a temporização. Esse integrado, quando recebe uma habilitação dada pelo XACK* do controlador gráfico, armazena os dados fornecidos pela memória. Por sua vez, a sua saída é a entrada paralela de um registrador de deslocamento (74166) que será o fornecedor da informação do pixel para o monitor. O carregamento do registrador de deslocamento é feito segundo o relógio gerado por T1 (controlador), e o deslocamento feito segundo o relógio de ponto.

É preciso, todavia, apagar-se o sinal nos retracos e, para isso, utiliza-se o 'clear' e a inibição de relógio do 74166 como mostra com detalhes a figura 32.

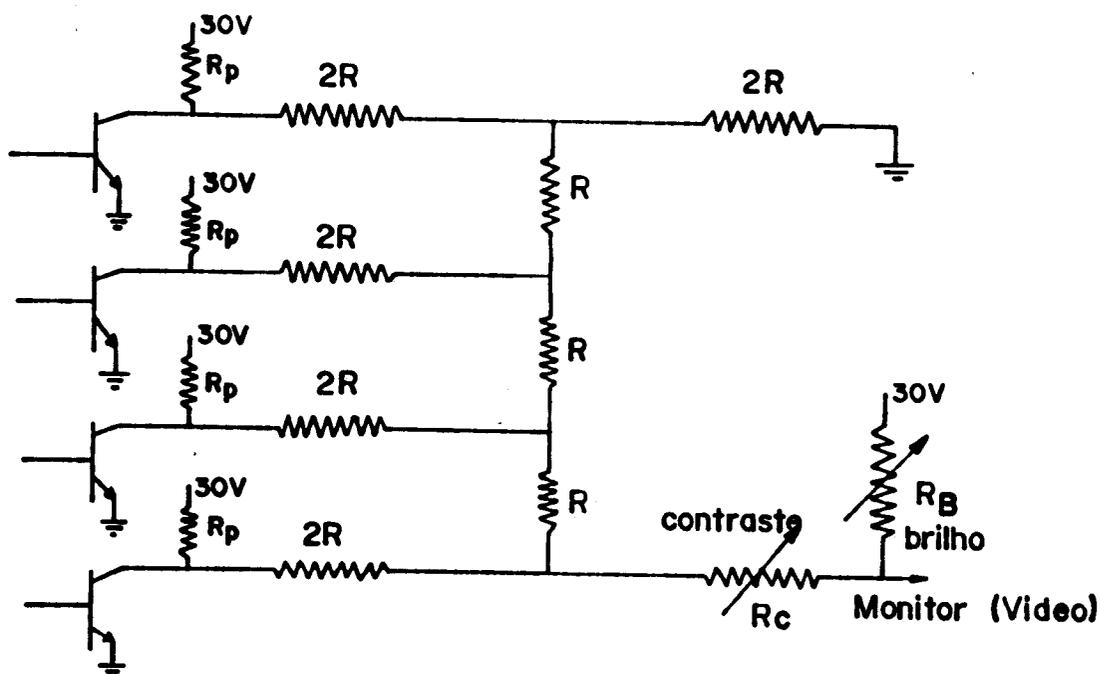
Assim, cada banco de memória fornece um bit de informação para o pixel.



Fis. 32- Circuito de apagamento de video no retraco

Esse bit é injetado num conversor D/A, juntamente com os dos outros bancos da memória, para fornecer a intensidade. Quando é retraco contudo, o monitor é mantido apagado.

O circuito D/A pode ser visto na figura 33 para o caso de 4 bancos de memória (16 níveis de intensidade).



Fis. 33- Circuito conversor D/A para intensidade

Os transistores vistos aqui são simplesmente as saídas de um integrado de coletor aberto para tensões altas (7406).

Sabe-se que (vide seção I.9) a vista humana percebe intensidades espaçadas logaritmicamente. A conversão D/A, porém, é linear. Verificou-se que a variação de intensidade de luz no monitor era aproximadamente logaritmica com a corrente de entrada no amplificador de intensidade (eixo Z).

Basta, então, fazer um conversor D/A que funcione como fonte de corrente. Para isso, na figura 33 deve-se fazer:

1. $R_p \ll R$ para que o resistor de polarização não influa na linearidade da conversão.
2. R_c e $R_b \gg R_{\text{entrada do amplificador}}$ para que o conversor seja realmente uma fonte de corrente.
3. $R \ll R_c$ e R_b para que R_c e R_b não interfiram no conversor.

Foram escolhidos os valores $R_p = 1K$, $R = 10K$ e R_c e R_b , 'trimpots' ajustáveis de 470K cada um.

CAPITULO IV O PROJETO - "SOFTWARE"

IV.1 INFORMACOES GERAIS

Existem, basicamente, no terminal, dois processadores: um é o alfanumérico enquanto o outro é o gráfico. Assim, existem dois programas distintos, um para cada processador. É claro que o programa do terminal alfanumérico é tal que permite a comunicação com o gráfico para passar dados vindos do computador, quando se faz esta opção de funcionamento.

O "software" desenvolvido para os terminais (nitidamente para o gráfico) é o mínimo necessário para o funcionamento do sistema. Atualmente, já estão sendo desenvolvidos programas mais elaborados tanto para a parte alfanumérica quanto para a parte gráfica.

Deve-se notar, também, que todos os programas desenvolvidos até agora, o foram na linguagem montadora do 8085A.

IV.2 TERMINAL ALFANUMERICO

Como já foi dito, o terminal alfanumérico é baseado numa nota de aplicação da INTEL americana. Como a nota já traz um programa, este foi utilizado com algumas alterações.

A estrutura do programa é do tipo 'foreground' e 'background'. No 'background', têm-se basicamente duas rotinas chamadas FRAME e POPDAT. A rotina POPDAT é aquela responsável por simular o acesso direto à memória, ou seja, é ela que transfere uma linha completa da memória de tela para o controlador 8275. Ela é executada uma vez a cada 80 caracteres, mais o tempo de retraco horizontal, o que dá aproximadamente 600 microssegundos, e é acionada pela interrupção 6.5 do 8085A. Já a rotina FRAME é executada no retraco vertical, ou seja, uma vez a cada 16.67 milissegundos. Tem como função ler e acertar o 'baud rate', ler o teclado, montar o carácter lido e também fazer o 'debounce' do teclado.

Na leitura do teclado, como este não é codificado, praticamente todo o trabalho fica para o programa. Cada vez que a rotina FRAME é executada, é lido o teclado. Inicialmente, uma variável chamada KEYDWN é colocada zero. No próximo FRAME, tenta-se ler o teclado, serando-se pulsos nas linhas de varredura, e lendo-se as linhas de retorno. Se uma tecla estiver pressionada, a variável KEYDWN é tirada de zero e são salvas as linhas de varredura e de retorno. No próximo FRAME, é verificado se a mesma tecla está sendo pressionada, e aí, então, busca-se o código ASCII correspondente à tecla. Nesse momento, coloca-se a variável KEYDWN num terceiro valor, indicando que a tecla já foi lida e decodificada. Tão logo a tecla seja liberada, a variável KEYDWN (no próximo FRAME) volta para zero. Isso tudo é feito por dois motivos:

1. Eliminar ruídos, de forma a ler somente as teclas intencionalmente pressionadas.
2. Ler uma tecla pressionada, somente uma vez.

A rotina FRAME também ajusta o endereço de início de tela, que é modificado quando se faz 'scroll'.

Os fluxogramas de operação das rotinas FRAME e POPDAT podem ser vistos nas figuras 34 e 35.

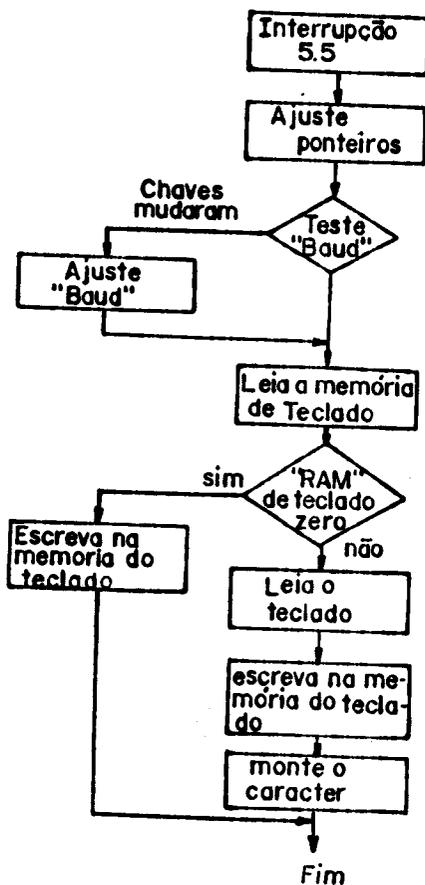


Fig. 34- Fluxograma da rotina FRAME

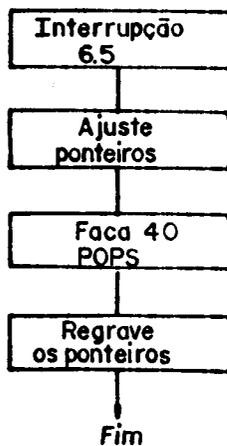


Fig. 35- Fluxograma da rotina POPDAT

O programa em 'foreground', em primeiro lugar, examina a chave de linha-local para saber o modo como o terminal está trabalhando. Se o modo for local, todo o dado a considerar é introduzido pelo teclado e a USART é totalmente ignorada. No modo linha, o dado recebido do teclado é enviado através da USART e qualquer ação a ser tomada é feita segundo os dados recebidos pela USART.

Um dado de entrada para o terminal pode ser de dois tipos:

1. Pode ser um caracter que existe um tratamento especial, como 'line feed', 'escape' ou 'backspace'.
2. Um caracter simplesmente a ser impresso.

No primeiro caso, é dado um tratamento conveniente, enquanto que, no segundo, o caracter é simplesmente colocado na tela na posição atual do cursor. Se o caracter recebido for 'escape', o caracter seguinte será considerado parte de uma seqüência de 'escape'. Se o caracter seguinte não formar uma seqüência 'escape' reconhecida pelo programa, a seqüência é

simplesmente ignorada. O fluxograma em 'foreground' pode ser visto na figura 36.

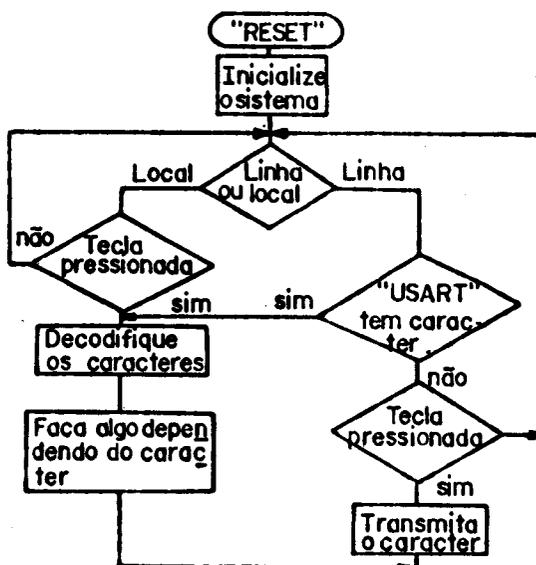


Fig. 36- Fluxograma do Programa 'foreground'

O protocolo é quase todo VT52 (33) a menos do 'line feed' reverso e da identificação (ESC Z). O protocolo implementado é o seguinte:

1. <ESC>A - move o cursor para cima uma posição.
2. <ESC>B - move o cursor para baixo uma posição.
3. <ESC>C - move o cursor para a direita uma posição.
4. <ESC>D - move o cursor para a esquerda uma posição.
5. <ESC>F - entra no modo gráfico.
6. <ESC>G - sai do modo gráfico.

7. <ESC>H - leva o cursor para a primeira linha e primeira coluna da tela.
8. <ESC>J - limpa todos os caracteres da posição do cursor até o fim da tela. A posição do cursor permanece inalterada.
9. <ESC>K - limpa todos os caracteres da posição do cursor até o fim da linha. A posição do cursor permanece inalterada.
10. <ESC>Y linha coluna - move o cursor para a linha e coluna especificadas. Os números de linha e de coluna são especificados como caracteres ASCII, cujos valores são o número mais 20 em hexadecimal, sendo que as linhas e colunas começam no zero.
11. <ESC>E - não é protocolo VT52. Limpa a tela completamente.

Cabe ao programa, também, inicializar os periféricos 8251A, 8253, 8255A, 8275. Um exame simples do programa e das especificações do fabricante desses dispositivos, é suficiente para se entender a programação do 8251A, 8253 e 8255A. Quanto ao 8275, está programado na inicialização para (parâmetros mais importantes):

1. 80 caracteres por linha.
2. 2 linhas por retraco vertical.
3. 10 linhas por caracter.
4. Cursor do tipo srifo piscante.
5. 32 caracteres por retraco horizontal.

O 'scroll' do terminal alfanumérico é feito automaticamente na última linha e é conseguido simplesmente mudando o valor de uma variável chamada CURAD no retraco vertical. O programa em 'foreground' modifica uma posição chamada TOPAD e, então, o 'background' (FRAME) troca TOPAD para CURAD.

IV.3 TERMINAL GRAFICO

Assim como no terminal alfanumérico, foi adotada a filosofia 'foreground'-'background'.

A rotina em 'background' é acionada uma vez a cada retraco vertical (como a rotina FRAME no alfanumérico). Ela é acionada através da interrupção 7.5 do processador 8085A. A sua função principal é transferir dados da memória de trabalho do processador para a memória gráfica. Ela pega dados de um 'buffer' organizado como 'FIFO' circular da memória de trabalho do 8085A, e os transfere para posições seqüenciais na tela gráfica. Repare que isso não é necessário na placa alfanumérica devido ao fato de lá a memória de trabalho do micro e a memória de tela serem a mesma entidade. Aqui, porém, uma vez escrita a informação na memória gráfica, o micro não precisa mais ficar passando informações para o controlador gráfico, o que o libera quase que totalmente para o trabalho de 'foreground'.

O 'background' verifica antes de qualquer coisa, se o 'buffer' de 256 caracteres com estrutura de 'FIFO' não está vazio. Se estiver, nada é efetuado e retorna-se para o 'foreground'. Se, porém, houver caracteres a serem transferidos, é preciso saber se o número de caracteres é maior ou menor que 70, pois este é o número máximo de transferências que podem ser feitas

por retraco vertical. Se o número for maior, transferem-se somente os primeiros 70 e deixa-se o restante para o próximo retraco. No outro caso, transferem-se todos os caracteres e zera-se a posição que indica que o "buffer" está vazio.

O "foreground", que aqui, como foi visto acima, tem muito tempo, é encarregado de receber os comandos do duto ou do terminal alfanumérico, e de executá-los. Os comandos interpretados pelo terminal gráfico no protótipo básico construído são:

1. H - leva o cursor gráfico para a primeira linha e primeira coluna, tão logo todo o "buffer" de transferência se esvazie.
2. Y N1 N2 N3 N4 - leva o cursor para a posição dada pelo endereço formado pelos caracteres ASCII N1 N2 N3 N4 tomados como valores hexadecimais. Repare que o mapeamento da tela gráfica é linear ou seja, a primeira posição da tela é 0C000H, a segunda 0C001H, e assim por diante. Esses endereços são exatamente os endereços do byte no espaço de endereçamento do processador. Se N1 N2 N3 N4 representarem um endereço menor que 0C000H, o comando é simplesmente ignorado.
3. Z - este comando limpa toda a tela.
4. A N1 N2 N3 N4 - é o comando de escrita na memória gráfica. N1 N2 N3 N4 formam um número até 16384 (se representarem um número maior, somente os 14 bits menos significativos serão considerados) e são considerados como sendo ASCII hexadecimais. Eles representam o número de posições que serão modificadas na tela, a partir da posição atual do

cursor. Depois, são aceitos N1 N2 N3 N4 bytes não codificados (8 bits) e, depois disso, retorna-se para a decodificação do próximo comando.

Como se pode ver, o programa básico é simples, e não necessita de maiores explicações.

CAPITULO V RESULTADOS E CONCLUSAO

V.1 GERAL

O terminal alfanumérico e gráfico descrito neste trabalho foi construído e testado no LIE. Foi utilizado um monitor de vídeo com fósforo verde (P31) de média persistência (vide capítulo I) produzido por uma indústria nacional (TDA). Os resultados obtidos são, agora, sucintamente descritos.

V.2 TESTES DE IMAGENS

Como o terminal foi projetado com a finalidade de apresentar imagens, tentou-se, em primeiro lugar, verificar a qualidade de imagem que poderia ser obtida.

Começou-se com apenas um banco de memória gráfica, ou seja, cada "pixel" poderia assumir somente dois valores. Claro está que, nesta situação, é preciso algum meio para simular níveis de intensidade. Utilizou-se, quando necessários níveis de intensidade, a aproximação de meio tom com um "pixel" sendo um conjunto de 2x2 pontos (figura 17 - capítulo I). A resolução espacial caiu, então, para a metade em cada eixo e, conseqüentemente, os contornos não ficaram muito nítidos, principalmente devido a baixa resolução da vertical ($250/2=125$).

Foram utilizadas 3 figuras para os testes (adaptadas para o terminal pelo Dr. Jan Slaets). Uma das figuras era simplesmente um desenho feito à mão, ou seja, era simplesmente um contorno, não contendo mais que 2 níveis de intensidade.

Esta figura, no terminal, manteve-se bastante fiel à original. Aqui, como não havia necessidade, não foi utilizada a aproximação de meio tom. Nas outras 2 figuras, utilizou-se desta aproximação, conseguindo-se 5 níveis de intensidade. As imagens, que eram um retrato da "Mona Lisa" e uma fotografia de Albert Einstein, apresentaram-se boas, mas contornos indesejáveis foram bastante nítidos e a qualidade de detalhes, um tanto fraca, além da já mencionada perda de resolução espacial.

Instalou-se então, o segundo banco de memória gráfica, ficando dois bits associados a cada ponto da tela. Utilizando-se novamente a aproximação de meio tom 2x2, com a diferença que agora com 13 níveis (vide Foley (3)), colocou-se a imagem de Albert Einstein na tela. A qualidade melhorou sensivelmente, mas a imagem continuou apresentando contornos indesejáveis. Acredita-se que, com 5 bancos de memória e sem o recurso da aproximação de meio tom, a qualidade de imagem seja muito boa.

V.3 TESTES COM RETAS

Foi testada a qualidade do terminal para apresentar retas. As retas foram geradas por um programa desenvolvido pelo Dr. Jan Slaets.

Verificou-se que, para inclinações próximas a 0 ou a 90 graus, a reta gerada era bastante ruim. Isso já era esperado teoricamente. Para melhorar a qualidade da reta, deve-se aumentar a resolução espacial ou utilizar técnicas de "antialiasing" (34). Para retas de inclinações próximas a 45 graus, a qualidade é bastante boa.

V.4 CONCLUSÃO

Acredita-se, devido aos resultados dos testes citados acima, que o terminal mostrou-se ser eficiente para o que ele foi projetado. São possíveis vários melhoramentos e alguns deles serão estudados no próximo capítulo.

CAPITULO VI MELHORAMENTOS

VI.1 CONSIDERAÇÕES GERAIS

Este capítulo tem por finalidade sugerir possíveis aperfeiçoamentos no terminal de vídeo, que é objeto deste trabalho. Ele busca dar uma visão geral das vantagens e ao mesmo tempo dos problemas de cada sugestão feita, de forma que o interessado tenha meios de avaliá-la e de escolher a expansão que mais se adapta às suas necessidades.

A maioria dos melhoramentos está voltada ao terminal gráfico, já que o terminal alfanumérico foi propositadamente feito com o objetivo de custo bastante reduzido e, conseqüentemente, suporta poucos melhoramentos sem uma mudança mais séria no "hardware".

Finalmente, deve-se salientar que as sugestões aqui apresentadas devem ser encaradas como possíveis futuras linhas de trabalho e não como produtos acabados, já que se trata de idéias apenas, que não foram testadas até o presente momento.

VI.2 AUMENTO DE RESOLUÇÃO GRAFICA

Analisam-se aqui, possíveis métodos para aumentar a resolução do terminal gráfico até agora descrito nas páginas anteriores. O terminal gráfico, como já foi visto, apresenta uma resolução de 512x250 pontos o que dá aproximadamente 65 linhas por polegada na horizontal e 45 linhas por polegada na vertical.

Pode-se obter maior resolução, basicamente, de duas maneiras:

1. aumentando-se a velocidade do monitor
2. usando-se varreduras entrelaçadas.

VI.2.1 AUMENTO DE RESOLUÇÃO COM MONITOR MAIS RÁPIDO

Por hipótese, suponha-se que é desejado aumentar-se a resolução do terminal gráfico de 512x250 para 512x500. O que é necessário?

Em primeiro lugar, deve-se garantir que o amplificador de vídeo do monitor tenha uma largura de banda suficiente, o que significa pelo menos igual à frequência de relógio de ponto. Pode-se pensar então que basta utilizar um amplificador de vídeo para 512x512 com o dobro da largura de banda do 512x250. No entanto, isto não é verdade. Quando se dobra o número de linhas, mantendo-se os 60Hz de frequência vertical, tem-se a metade de tempo para cada linha horizontal. Suponha então que se gaste 64 microssegundos por linha na horizontal na resolução baixa, sendo 50 microssegundos "úteis" e 14 microssegundos de retrazo. Como o tempo de linha deve ser reduzido à metade, então têm-se, apenas, 32 microssegundos. O retrazo não pode, porém, ser reduzido, e assim o tempo "útil" cai para 18 microssegundos. Assim, a frequência do relógio de ponto deve ser aumentada quase três vezes. Portanto, mantendo a mesma arquitetura, para se dobrar a resolução em linhas, deve-se mais do que dobrar a largura do amplificador de vídeo, além de acelerar bastante a temporização horizontal do monitor. Se o monitor disponível admitir estas modificações (não é o caso do que existe no LIE), aumentar a resolução fica bastante simples.

VI.2.2 AUMENTO DE RESOLUCAO GRAFICA - ENTRELACAMENTO

Para aumentar, ao máximo, a resolução com um determinado monitor, utiliza-se de um truque que é usado também na televisão, qual seja, entrelacar as telas. Para isso, duas imagens são colocadas na tela de forma alternada, cada uma, portanto, com frequência de refrescamento de 30Hz e cada uma com resolução (no nosso caso) de 512x250. Uma é deslocada em relação a outra de meio traço horizontal. Repare que, um dos dois campos (cada tela aqui é chamada de campo) deve ter meias linhas horizontais no começo e no fim da tela. Esse, porém, é o único problema pois toda a temporização permanece a mesma.

No caso do terminal gráfico descrito nesse trabalho, são utilizados bancos de memória (até em número de 8), cada um armazenando uma tela de 128000 pontos (512x250). Poder-se-ia transformar cada dois bancos de memória numa tela 512x500 utilizando-se do entrelacamento. É claro que aí os 256 níveis possíveis de intensidade passariam a somente 16.

Para se levar a cabo essa modificação, deve-se fundamentalmente acrescentar um biestável que indique qual campo será mostrado na próxima tela, e um circuito de temporização que atrase o retraco vertical ou não, segundo o estado do biestável. Assim, num dos campos, o sinal de retraco horizontal é o mesmo do terminal alfanumérico, enquanto no outro campo, o sinal é atrasado de exatamente meia linha. É claro que cuidados especiais devem ser tomados quando o retraco é atrasado, principalmente no sinal de vídeo, e no endereço de acesso da memória. De qualquer forma, é fundamental que ambos os bancos de memória recebam os sinais de RAS* sempre, pois, de outra

forma, haveria problemas de perda de dados (lembre-se de que não existe circuito especial de refrescamento da memória - veja capítulo III).

VI.3 AUMENTO DE VELOCIDADE DE ALTERAÇÃO DE TELA

Muitas vezes pode ser necessária uma maior velocidade de modificação de imagens no terminal gráfico. Como foi visto porém, a alteração deve ser feita, normalmente, somente nos períodos de retraco horizontal ou vertical que é quando o controlador de memória gráfica não a está acessando.

Da forma como está montado o protótipo, entretanto, o processador 8085A é que modifica a memória gráfica. Como ele é lento, não é possível aproveitar-se o retraco horizontal (em torno de 17 microssegundos), o que ao primeiro exame pode parecer desprezível, mas não se pode esquecer que uma tela apresenta 250 retracos horizontais. O tempo perdido só aqui é então 250×17 microssegundos o que dá a significativa quantia de 4.25 milissegundos por tela (16.67 milissegundos). Mesmo no retraco vertical, o rendimento do microprocessador é muito baixo, podendo-se, no máximo, modificar 100 posições, quando o retraco vertical é o máximo permitido (menos que 2 milissegundos). O problema é minimizado em certos casos, devido aos vários modos de escrita possíveis, alguns deles permitindo que cada byte de dado do microprocessador modifique 64 bits da memória gráfica (8 por banco).

Esse baixo rendimento, natural de microprocessadores, pode ser drasticamente elevado, se for utilizado um circuito de acesso direto à memória. Esse circuito teria a função de receber dados de uma memória secundária de alta capacidade de armazenamento e transferi-los para a memória gráfica, sem passar pelo 8085A. A razão de transferência pode ser tão alta quanto uma a cada oito pulsos de relógio de memória, o que dá uma transferência, a cada 544 nanossegundos, para o relógio de memória em 68 nanossegundos. Assim, pode-se conseguir pelo menos 30 modificações por retrazo horizontal, o que dá $250 \times 30 = 7500$ modificações por tela, somente devidas aos retraços horizontais. O retrazo vertical contribui com mais 4000 modificações, o que dá 11500 modificações por tela. Pode-se conseguir, então, usando-se memórias mais rápidas e alterando o relógio, modificar os 16kbytes de um banco de memória em uma só tela o que introduziria uma nova dimensão na tela: o tempo.

O circuito controlador de acesso direto à memória precisa apenas solicitar um "hold" ao microprocessador 8085A. Repare, porém, que, para máxima eficiência, o "hold" deverá ser mantido ativo durante toda a transferência. Quando o 8085A atender ao pedido, o controlador poderá modificar os dados das memórias simplesmente gerando um sinal de MWR* e fornecendo os endereços e dados à memória. É importante lembrar que os endereços da memória gráfica começam em 0C000H. Deve-se fazer também com que o controlador só modifique posições nos retraços. O controlador deverá, também, esperar que o pino 7 do integrado 74S163(T2) vá para nível alto para requisitar a próxima escrita. Enquanto isso não ocorrer, os dados e os endereços devem ser mantidos constantes.

É claro que o projeto do circuito controlador de acesso direto à memória não é trivial. Devem ser projetadas interfaces convenientes entre ele, o processador gráfico e uma memória secundária de alta velocidade. Provavelmente, ele deverá ter um "buffer" para armazenar os dados que vêm da memória secundária e não podem ser imediatamente escritos na memória gráfica.

Apesar da dificuldade de implantação, para muitas aplicações, o controlador de acessos à memória será fundamental.

VI.4 AUMENTO NO NÚMERO DE CORES E INTENSIDADE

Como foi visto nos capítulos anteriores, o número máximo de barras de memória que podem ser ligadas ao processador gráfico é oito. Essa limitação prende-se ao fato de que o micro processador é de 8 bits e em dois modos de carregamento (veja seção II.3 modos 1 e 3) ter-se o conceito de "bit do duto de dados, associado ao banco". Ora, como são só 8 bits no duto de dados, só se pode ter 8 bancos de memória. Se porém os modos 1 e 3 não forem fundamentais, para colocar-se mais bancos deve-se somente acrescentar outra 8212 de habilitação (Além da já existente (R)). Deve-se, porém, ao acrescentar mais bancos de memória, ter sempre o cuidado de verificar se os "drivers" existentes fornecem corrente suficiente.

VI.5 LEITURA DOS DADOS DA MEMÓRIA GRÁFICA

O terminal gráfico é, da forma como está construído (veja seção II.3), um terminal apenas de escrita ou seja, não se lê a memória gráfica uma vez escrita. Na verdade, o próprio processador gráfico não tem acesso às informações contidas nos bancos de memória.

Essa característica pode, entretanto, ser inconveniente em muitas aplicações onde, uma vez montada a tela, quer-se transferi-la para um outro dispositivo, um 'plotter' por exemplo.

A primeira coisa a ser feita seria lidar, de alguma forma, as saídas das memórias dos bancos gráficos, ao duto do processador gráfico. Isso pode ser feito curtocircuitando-se os pinos 14 e 2 de todas as memórias 4116. Nesse caso, deve-se fazer uma lógica entre um dos sinais de modo de carreamento (S2) e os pulsos de escrita MWR* e de leitura MRD* para que somente quando se escrever na memória, os integrados M9 e M10 estejam habilitados (atualmente sempre, ou M9 ou M10 está habilitado, dependendo de S2). A direção de condução dos integrados M9 e M10 seria dada ou pelo pulso de MERD* ou pelo pulso de MEWR* (esses pulsos nada mais são do que o MRD* e MWR* da saída do integrado 74257(P), quando o endereço acessado pelo micro é maior que 0C000H. O sinal de MERD* precisaria ser criado (pode ser utilizada a porta restante do 74S32(T16) para isso) e seria um 'ou' entre o MRD* e o EXT* e a entrada 3 de T13 seria um 'ou' entre MRD* e MWR*. Seria necessário também um 'latch' para armazenar o dado fornecido pela memória no instante certo (que é dado pelo sinal XACK* fornecido pelo controlador gráfico). Uma vez lido pelo processador 8085A, isso poderia ser transmitido ao computador central através de um 'handshake' similar ao utilizado no projeto para o recebimento de dados pelo processador gráfico (veja ligações do 8212(U)).

VI.6 MELHORIAS NO "SOFTWARE"

Grandes melhorias podem ser introduzidas nos programas, tanto do terminal alfanumérico como do terminal gráfico.

No terminal alfanumérico, alterando-se o programa residente na 2716, pode-se implementar muitos tipos de protocolo. No Laboratório de Instrumentação Eletrônica do IFQSC já está em andamento a adaptação desse programa para transformar o terminal num VT100, o que será de grande utilidade já que não existe nenhum terminal nacional seguindo esse protocolo (que é um protocolo padrão norte-americano).

Quanto ao terminal gráfico, o programa deve ser bastante expandido, de tal forma que o processador gráfico seja capaz de executar funções simples como o traçado de reta a partir de dois pontos, o traçado de um círculo a partir de três pontos, o traçado de um polígono a partir de seus vértices, etc. O programa deve, também, suportar mudanças de modo de carregamento para aproveitar ao máximo o "hardware". Atualmente, o programa existente é apenas o mais básico e fundamental para o funcionamento do terminal. Já existe trabalho nesse campo, também no Laboratório de Instrumentação Eletrônica do IFQSC.

APENDICE I MANUTENÇÃO

I.1 CONSIDERAÇÕES GERAIS

A intenção desse apêndice é dar uma idéia das linhas gerais de conduta a serem tomadas quando o terminal alfanumérico ou terminal gráfico apresentarem algum tipo de defeito. Claro está que uma previsão de todos os defeitos é completamente impossível por motivos óbvios, mas este apêndice tentará ajudar ao máximo o responsável pela manutenção a achar o defeito o mais rapidamente possível. Este apêndice foi aqui incluído devido ao fato de que o presente trabalho é fundamentalmente prático.

Três tipos básicos de defeitos poderão ocorrer e serão objetos das seções seguintes: defeitos de comunicação, defeitos perceptíveis na tela e defeitos totais ou seja, não se sabe a origem do defeito sendo que nem tela e nem comunicação parecem funcionar.

I.2 TERMINAL ALFANUMÉRICO

I.2.1 DEFEITOS TOTAIS

Por defeitos totais entendem-se defeitos que não se conseguem classificar em de tela ou de comunicação, pois o terminal simplesmente recusa-se a funcionar.

Comece testando as alimentações. Se tudo estiver bem, continue testando os relógios (pino 37 do 8085A, pino 9 do 8253, pino 9,20,25 do 8251A, pino 30 do 8275, pino 7 do 74166, pino 10 do 7474, pino 9 do 74175 e pino 18 do 8253). Com exceção dos pinos 9 e 25 do 8251A, todos os outros devem estar funcionando tão logo se ligue o aparelho.

No caso deles estarem corretos, devem ser testadas as habilitações das memórias de programa e trabalho. O pino 21 da 2716 de programa deve estar em 5 volts e o pino 20 deve ter pulsos, pelo menos no 'reset'. O mesmo deve ocorrer com os pinos 8 das 2114s. Em caso de problemas, verifique o programa da 2716 e o decodificador de endereços (74188).

1.2.2 DEFEITOS DE COMUNICACAO

São aqueles caracterizados por uma impossibilidade ou ao menos uma falha na comunicação entre o homem ou o computador central e o terminal. São caracterizados pelo fato de a tela aparentemente funcionar bem, mas não se conseguir modificá-la. Esse defeito pode ocorrer tanto no terminal alfanumérico quanto no gráfico.

O terminal alfanumérico é independente e, conseqüentemente, pode ser testado sem o terminal gráfico.

A primeira coisa a ser feita para se testar o terminal alfanumérico, se não se sabe o tipo de defeito (de comunicação ou de tela) é testá-lo em modo local e em modo linha. No modo local qualquer caracter teclado deve aparecer na tela. Da mesma forma, com o terminal no modo linha e ligado a um computador, deve-se mandar o computador escrever na linha um conjunto de caracteres conhecidos. Se em nenhum dos casos a tela responder,

é bastante provável que o defeito seja de tela (veja próxima seção). Se, porém, num dos casos o terminal funcionar corretamente, o problema é de comunicação.

Esses problemas podem ocorrer tanto no teclado como na 'USART' de comunicação. Se uma tecla não responder quando pressionada e todas as outras funcionarem, muito provavelmente aquela tecla está com defeito. Se porém, uma série de teclas, mas não todas, apresentarem defeito, examine as linhas de varredura e as linhas de retorno da matriz de teclado e verifique se estão todas intactas e se as resistências de retorno estão em ordem. Se tudo estiver bem, pode acontecer de o 8255A estar com problemas. Finalmente, se mesmo o 8255A estiver bem, a falha pode estar na ligação do 8255A no duto de dados do processador (veja defeitos totais). Se a 'EPROM' do terminal alfanumérico que contém o programa foi mudada, pode ter sido esquecido de se incluir no novo programa a inicialização do 8255A.

Um teste simples para verificar se o problema está 'antes' ou 'depois' do 8255A é verificar com o osciloscópio se estão sendo gerados pulsos em todas as linhas de varredura (pinos 4,3,2,1,40,39,38,37) do 8255A e também verificar se existem pulsos de retorno quando se tecla.

Se o problema for na comunicação de linha, ou seja, na 'USART' procure ter certeza de que o problema não é no computador principal. Depois procure ter certeza de que o protocolo de comunicação é o mesmo. Não se esqueça de que o terminal alfanumérico recebe caracteres numa linha RS232-C. Certifique-se de que a linha do computador é também RS232-C. Se

o terminal recebe caracteres completamente sem sentido quando o computador manda um texto, provavelmente o que está errado é velocidade de comunicação (as velocidades de transmissão e recepção são diferentes). Se o terminal receber a maioria dos caracteres corretos, mas receber vários errados, o que provavelmente está acontecendo é que o número de bits na transmissão e na recepção não é o mesmo.

Repare que o terminal está programado para receber 8 bits sem paridade, transmitir 2 'stop bits' e para receber 1 ou mais 'stop bits'. Verifique depois disso se os 'line driver' e 'line receiver' estão funcionando (não se esqueça de verificar os 15 volts de alimentação do 75188).

Se tudo isso estiver bem, o defeito pode estar no relógio de transmissão e recepção do 8251A. Se este estiver com 16 vezes o valor da velocidade de transmissão, tudo estará bem e o problema provavelmente está na ligação do 8251A nos dutos do processador. Se o relógio, porém, estiver errado, o problema será no 8253. Verifique se este último recebe o relógio do 8085A dividido por dois e se as suas ligações ao duto estão corretas.

I.2.3 DEFEITOS DE TELA

Se o problema estiver na tela, comece testando o monitor de vídeo. Se esse mostra os caracteres corretos e sem tremulação, mas apresentar também pontos ou retas acesos que não deveriam estar assim, provavelmente isso se deve a ruídos. Procure fazer a distância entre a placa do terminal e o monitor a menor possível, nunca se esquecendo do terra.

Se o monitor apresentar oscilações constantes de baixa frequência, o que está acontecendo é que a frequência da vertical não está exatamente em 60Hz. Coloque um medidor de frequência no sinal de vertical e varie o relógio de ponto até obter 60Hz exatamente. A oscilação, provavelmente, desaparecerá. Se o monitor apresentar um traço de luminosidade intensa numa região só da tela, verifique se as bobinas defletoras estão corretamente conectadas às placas de controle do monitor.

Se o problema não estiver no monitor, verifique se os sinais de retraco vertical e horizontal estão chegando corretamente. Esses pulsos são ativos baixos, têm duração de 1.4 milissegundos e 30 microssegundos respectivamente, e ocorrem a cada 16.6 milissegundos e 64 microssegundos respectivamente. Se eles não estiverem ocorrendo, verifique se existem pulsos nos pinos 7 e 8 do 8275. Em caso de haver pulsos no 8275 e eles não aparecerem na saída, examine os pinos 2 e 6 do 74175 para ver se este último não está com problemas. Se o problema estiver apenas no retraco horizontal, provavelmente quem está com problemas é o 8253. Caso o 8275 não forneça os pulsos corretos, obviamente é ele que está com problemas ou no 'hardware' (verifique o pino 30 para saber se há relógio, fio solto, pastilha estragada), ou no 'software'.

Se tudo estiver certo com os sinais de retraco, o problema só pode estar no sinal de vídeo. Uma observação muito simples, nesse caso, pode ajudar bastante: existe cursor? Se o cursor existe mas os caracteres não aparecem na tela, o problema é ou na 2716 decodificadora de caracteres ou no registrador de deslocamento 74166 ou no 7410. Aqui (como no caso do exame do

8275) não se esqueça de verificar se o relógio de ponto e o de caracter estão chegando no 74166 (ponto - pino 7, caracter - pino 15). Se mesmo o cursor não existir, verifique se o pino 37 do 8275 apresenta pulsos. Em caso afirmativo o defeito estará no 74175. Verifique se esse integrado recebe pulsos no pino 9.

Caso o terminal alfanumérico tenha acabado de ser montado, é fundamental, além de tudo que foi visto acima, examinar-se as habilitações de todos os integrados pois estas são frequentemente esquecidas nos diagramas e montagens.

I.3 TERMINAL GRAFICO

I.3.1 DEFEITOS TOTAIS

Como no terminal alfanumérico, comece testando as alimentações. Depois teste os relógios nos pinos 37 do 8085A (A), 2 do 74S163(T1) e 2 do 74S163(T2). Repare que esse teste deve ser feito, conectando-se o terminal gráfico ao terminal alfanumérico, pois o relógio de ponto vem de lá. Se não existir relógio no pino 2 de T1, que é justamente o relógio da placa alfanumérica, há problemas na conexão. Repare que todos esses relógios devem estar funcionando ao se ligar o terminal gráfico.

Estando tudo bem, continue verificando se chegam os sinais de retrazo horizontal e retrazo vertical do terminal alfanumérico. Verifique, então, se chegam pulsos nos pinos 11 do 74373 (M9) e 7 e 15 do 74166 (M10) (ambos do módulo de memória). Se não existirem pulsos, verifique a conexão entre o módulo do processador gráfico e o módulo de memória.

Quanto ao processador, verifique se os 'drivers' 74LS245 (B,D,E) estão corretamente habilitados. Os pinos 1 de D e E e 2 de C devem estar aterrados enquanto o de B deve pulsar. O pino 19 dos integrados B,D e E e o pino 10 de C devem também estar aterrados. Se em B,C e D eles não estiverem, cheque a ligação dos mesmos com o pino 5 de T13 que é a saída do sinal HLDA que deve estar baixo.

Teste os decodificadores 74138 (M,N) e 74139 (O) e verifique se eles enviam pulsos para os pinos 18 da 2732(J), 1 dos 8212(R,S,U) e 8 das 2114 (F,G,H,I) pelo menos no 'RESET'.

Verifique se chegam pulsos aos pinos 20 da 2732(J) e 10 das memórias, pelo menos no 'RESET'. Se não ocorrer isso, o problema provavelmente está no integrado 74LS257(P).

I.3.2 DEFEITOS DE COMUNICACAO

Só existe um meio de se comunicar com o terminal gráfico e essa comunicação é apenas num sentido. Assim fica simples descobrir o que pode produzir esse tipo de defeito. A comunicação é feita através da 8212(U). Verifique se o pino 11 do referido integrado apresenta um pulso ativo alto, quando o computador central (ou terminal alfanumérico) envia um dado e, também, se os pinos 3,5,7,9,16,18,20 e 22 de U estão corretamente conectados ao duto de dados do computador central (ou terminal alfanumérico).

É importante, aqui, verificar se a habilitação do 74126 (Z), que também vem do computador central, está baixa, pois senão o próprio computador será perturbado quando conectado ao terminal gráfico.

Deve-se verificar também a ligação entre o pino 23 de U e o 5 do 8085A e certificar-se de que esse pino abaixa quando vem um caracter. Ainda, deve-se verificar se o pino 13 de U eleva-se quando o processador tentar ler U, o que é um teste fácil de ser feito, já que no 'RESET' o programa lê a 8212 (U). Se assim não ocorrer, o problema estará no integrado 74257 (P) ou no 7404 (Y).

1.3.3 DEFEITOS DE TELA

Os sinais de retrazo vertical e horizontal são tirados do terminal alfanumérico e assim o problema que houver deverá ser analisado pela seção 1.2.3.

Aqui o grande problema é sempre no sinal de video que é o pino 14 do 74166 (M10) do módulo de memória. Se ele não apresentar pulsos (pelo menos quando se liga o aparelho e, conseqüentemente, a memória está com zeros e uns) verifique se os pinos 7, 15, 9 e 6 recebem pulsos. Os sinais dos pinos 7 e 15 são simplesmente trazidos pela conexão entre o processador gráfico e o módulo de memória. Já os sinais dos pinos 6 e 9 dependem dos sinais de retrazo e do sinal apresentado no pino 10 de T15.

Se tudo estiver correto, o problema pode ser no registrador 74LS373 (M9). Cheque o pino 1 desse integrado e certifique-se de que está aterrado. Verifique, também, se existem pulsos no pino 11. Se não houver, o problema está no controlador de memória dinâmica (circuitos compostos pelos integrados designados pelos símbolos T seguidos de um número) que será analisado logo adiante.

Se o pino 14 do 74166 (M10) apresenta pulsos, mas verifica-se na tela que a imagem é instável (muda com o tempo sem qualquer motivo aparente), o problema está no acesso às memórias dinâmicas 4116. Deve-se certificar de que as tensões de 4116 estão corretas e sem "ruído" (35). O responsável pelo acesso à memória dinâmica é o controlador de memória dinâmica. É preciso verificar se todos os tempos descritos no capítulo 3 estão sendo cumpridos e, caso não, deve-se ajustar o relógio de memória (o relógio de memória é aquele gerado com o integrado 7414 (T0) na placa do processador gráfico).

Uma vez ajustados os tempos, se ainda o problema persistir, deve-se verificar se não existe ruído nos pulsos RAS, CAS e endereços que chegam à memória.

Deve-se certificar, também, que os contadores T9, T10, T11 e T12 estão realmente contando e que existem pulsos nos pinos 19 dos integrados 74LS245 (T7 e T8) que fazem a multiplexação de endereços. Um teste para verificar se os endereços estão corretos é ligar-se um analisador lógico às saídas 12,13,14,15,16,17 e 18 (7 bits) de T7 ou T8 (devem estar ligados juntos) e fazer-se a aquisição a cada descida do sinal RAS. O analisador deve mostrar uma seqüência crescente de números entre 00H e 7FH circular. Repetindo-se o procedimento, somente que mudando o sinal de RAS para o de CAS, deve-se observar a constância de um determinado endereço durante 128 aquisições.

Se a saída do terminal gráfico estiver correta, mas não se conseguir escrever na tela, deve ser verificado diretamente o WR da memória (pino 3) e se não houver o pulso, toda a lógica anterior. Repare que no "RESET" é feita uma zeragem de toda a memória gráfica, de forma que vários pulsos devem surgir. Deve-se verificar, também, se a multiplexação feita pelos 74LS245 (V,X) do processador está correta.

REFERENCIAS BIBLIOGRAFICAS

1. Elmsquist, K. A. & Fullmer, H. ,Standard Specification for S-100 Bus Interface Devices, IEEE Computer, 28-52, 1979.
2. Sherr, S., Electronic Displays, Wiley, New York, 1979.
3. Foley, J. D. & Van Dam, A. , Fundamentals of Interactive Computer Graphics. Addison Wesley, Washington, 1982.
4. Goppert, D. V., Basic Electron Tubes. McGraw-Hill, New York, 1951.
5. Spansenberg, K. R., Vacuum Tubes. McGraw-Hill, Palo Alto, 1948.
6. DeVere, C., Cathode Ray Tube. Beaverton, 1967.
7. Mc Manisal, D. & Stevenson, D., Architecture of the IBM 3277 Graphics Attachment, IBM Systems Journal, 19(3), 331-334, 1980.
8. DeVere, C., Storage Cathode-Ray Tubes and Circuits. Beaverton, 1967.
9. Slottow, H. G., Plasma Displays, IEEE Transactions on Electron devices 23(7), 57, 1976.
10. Tannas, L., Flat Panel Displays: A Critique, IEEE Spectrum, 15(7), 1978.

11. Woodsford, P. A., The HRD-1 Laser Display System, SIGGRAPH 74 Proceedings, 10(2), 68-73, 1976.
12. Catmull, E., A Tutorial on Compensation Tables, SIGGRAPH 79, Proceedings, 13(2), 1-7, 1979.
13. Hunt, R. W. G., The Reproduction of Color, Wiley, New York, 1975.
14. Judd, D. & Wyszecki, G., Color in Business, John Wiley and Sons, 1975.
15. Wasserman, G., Color Vision: An Historical Introduction, John Wiley and Sons, 1978.
16. Munsell Color Company, Book of Color, Baltimore.
17. Ostwald, W., Colour Science, Winsor & Winsor, Londres, 1931.
18. Nemcsics, A., The Coloroid Color System, Color Research and Application, 5(2), 113-120, 1980.
19. Rine, D. C. & Aiken, R. M., Computer Education, IEEE Computer, 13(6), 1980.
20. Myers, W., Computer Graphics: The Human Interface, IEEE Computer, 13(6), 45-54, 1980.
21. Lipson, J. I., Technology in Science Education: The Next 10 Years, IEEE Computer, 13(7), 23-28, 1980.
22. Myers, W., Computer Graphics: Reaching The User, IEEE Computer, 14(3), 7-17, 1981.
23. Lerner, E. J., The Computer Graphics Revolution, IEEE Spectrum, 18(2), 35-39, 1981.

24. Waite, M. , Computer Graphics Primer. Howard W. Sams & Co., Inc., Indianapolis, 1979.
25. Intel, Peripheral Design Handbook. Intel Corporation, Santa Clara, 1981.
26. Intel, Component Data Catalog. Intel Corporation, Santa Clara, 1981.
27. Richter, W. H. , Altnether , J. P., Application note AP-133, Intel Corporation, 1982.
28. Motorola, Memory Data Manual. Motorola Inc., 1980.
29. Peatman, J. B., Digital Hardware Design. McGraw Hill, 1980.
30. Zuffo, J. A., Fundamentos de Arquitetura e Organização dos Microprocessadores. Edgard Blöcher Ltda., São Paulo, 1978.
31. Mostek, Data Book and Designers Guide, Mostek Corporation, 1979.
32. Texas, The TTL Data Book. Texas Instruments Inc., 1980.
33. Digital, VT100 User Guide. Digital Equipment Corporation, 1979.
34. Woote, D., Compatible MK4027 and MK4116 Memory System Designs. In: Memory Data Book and Designers Guide, Mostek Corporation, 1979.