PROJETO DE UMA INTERFACE DE
VISUALIZACAO GRAFICA

Carlos Norberto Fischer

Orientador:

Prof. Dr. José Hiroki Saito

Dissertação apresentada ao Instituto de
Ciências Matemáticas de São Carlos-USP, para
obtenção do título de "Mestre em Ciências de
Computação".

São Carlos

1988

AGRADECIMENTOS

Agradeço ao Prof. Dr. José Hiroki Saito pela amizade e dedicada orientação.

Agradeço aos meus familiares pelo apoio e incentivo.

A Olga Geromel pelo carinho e paciência.

Aos meus amigos de pós-graduação, aos professores e funcionários do instituto.

Ao Departamento de Computação e Estatística da UFSCar, pelo suporte para o desenvolvimento deste trabalho.

Ao CNPq e a FAPESP pelo apoio financeiro.

A Deus pela força maior.

RESUMO

Este trabalho visa descrever o projeto de uma Interface de Visualização Gráfica, capaz de gerar e apresentar traçados em um dispositivo de raios catódicos.

A Interface de Visualização Gráfica projetada está baseada em um controlador gráfico, do tipo EF9367 da Thomson Semiconductors, e, juntamente com a programação gráfica básica correspondente, é capaz de gerar e apresentar figuras e textos em um monitor de vídeo de modo rápido, a partir de informações (comandos) enviadas por um sistema de arquivos.

A implementação do protótipo foi realizada sobre um microcomputador, e a interface projetada apresenta uma resolução de 1024 x 416 "pixels" de três bits de informação.

ABSTRACT

This text describes the design of a Graphic Display Interface for generation and drawing of graphics in a cathodic ray device.

The designed interface is based on EF9367 Graphic Controller from Thomson Semiconductors. The interface and its basic graphical programming provide fast generation and visualization of pictures and texts, in a video monitor, using informations (commands) from a file system.

The implementation of the prototype was consummated using a microcomputer. The designed interface presents a three bits pixel information and 1024 x 416 resolution.

INDICE

1	INTRODUÇÃO	01
2	SISTEMAS GRAFICOS	05
2.1	Considerações Iniciais	05
2.2	Descrição de um Sistema Gráfico	06
2.3	Padronização Gráfica	07
2.3.1	GSFC - CORE ("Graphic Standards Planning Committee" - CORE)	09
2.3.2	GKS ("Graphical Kernel System")	10
2.3.3	GKS - 3D ("Graphical Kernel System" - 3D) ...	12
2.3.4	PHIGS ("Programmer's Hierarchical Interactive Graphics System")	12
2.3.5	CGI ("Computer Graphics Interface")	13
2.3.6	CGM ("Computer Graphics Metafile")	14
2.3.7	DGIS ("Direct Graphics Interface Specification")	15
2.4	Aplicações de Sistemas Gráficos	16
2.4.1	Gerenciamento de Informação	17
2.4.2	Gráficos Científicos	18
2.4.3	Comandos e Controles	18
2.4.4	Geração de Gráficos em Tempo Real	18
2.4.5	Projetos Elétricos e Mecânicos	19
3	PROJETO DE INTERFACES DE VISUALIZAÇÃO GRAFICA INTERATIVA	20
3.1	Considerações Iniciais	20

3.2	Sistemas de Visualização Interativos	21
3.2.1	O Video com Tubo de Raios Catódicos	22
3.2.2	CRT Colorido	28
3.2.3	Dispositivos Gráficos de Entrada	31
3.3	O Registrador de Deslocamento de "Pixels"	32
3.4	As VRAMs	35
3.5	A Interface de Visualização Gráfica	37
3.5.1	A Interface para o Sistema com Exploração Rastreada	37
4	IMPLEMENTAÇÃO DA INTERFACE DE VISUALIZAÇÃO GRAFICA	44
4.1	Considerações Iniciais	44
4.2	Características do Microcomputador	44
4.3	Características da Interface	45
4.4	O CI EF9367	47
4.4.1	Descrição dos Pinos	51
4.4.2	Os Registradores Internos do EF9367	53
4.5	O Projeto da Interface	58
4.5.1	O Sinal FMAT	58
4.5.2	O endereço de Escrita	58
4.5.3	Seleção do EF9367	59
4.5.4	Sincronismo do EF9367 com o Z80A	60
4.5.5	Geração dos Sinais de Temporização	64
4.5.6	Seleção dos CIs de Memória	72
4.5.7	Seleção das Cores	73
4.5.8	O Registrador de Deslocamento	74
4.5.9	Sinal de Sincronismo e a Saída RGB	75

4.5.10	Memória de quadro	75
4.5.11	Interface com o Monitor de Video	77
5	SISTEMA PROGRAMACIONAL BASICO	78
5.1	Considerações Iniciais	78
5.2	As Primitivas Gráficas	79
5.3	Observações	87
6	COMENTARIOS, CONCLUSÕES E PROPOSIÇÕES	89
6.1	Comentários	89
6.2	Conclusões	95
6.3	Proposições	97
	BIBLIOGRAFIA	99
	APENDICE	

CAPITULO 1 - INTRODUÇÃO

Existem situações em que se necessita apresentar informações gráficas em formatos fáceis e seguros de serem interpretados.

Esses formatos podem ser tabelas, histogramas, gráficos em curvas, em setores, etc., e serem utilizados em muitas áreas, como educação, economia e estatística em geral.

Outras vezes, pode-se desejar visualizar o traçado de curvas, como para análise de forma de onda e representação de funções.

Também, pode-se desejar fazer a apresentação de formas e contornos de objetos reais.

Muitas vezes, é importante que a apresentação da figura seja rápida, devido a necessidade de interação do usuário com o sistema gráfico. Embora o tempo de processamento para a geração da figura seja também muito importante, enfoca-se o presente trabalho para o aspecto da visualização gráfica, considerando-se que os dados gráficos já estejam processados e disponíveis em algum arquivo.

Portanto, o tema deste trabalho surgiu do interesse em se desenvolver uma interface de visualização e a programação básica correspondente, considerando-se a apresentação de textos e figuras, de forma rápida e com resolução relativamente alta.

A Interface de Visualização Gráfica (IVG) foi implementada para efeitos de protótipo num microcomputador, e apresenta os

resultados num dispositivo gráfico de saída, um monitor de vídeo, com bom desempenho em termos de velocidade, podendo ser usada em aplicações de texto, com caracteres de vários tamanhos e tipos de escrita, e de traçado de figuras, como formas geométricas, gráficos, mapas e representações de objetos reais, em preto e branco ou a cores.

A finalidade acadêmica do trabalho foi a de se abter conhecimentos adicionais sobre sistemas gráficos, tanto no que se refere à dispositivos gráficos, o "hardware", como à sistemas programacionais dedicados, os "softwares" gráficos, que compõem aqueles sistemas.

A IVG está baseada no controlador gráfico EF9367 da Thomson Semiconductors, justificado pelo fator econômico, a sua disponibilidade, e também por servir como base, uma introdução, ao desenvolvimento de IVGs.

Outro ponto importante é o fato do CI EF9367 ter sido desenvolvido em universidade, L'UNIVERSITE PIERRE ET MARIE CURIE, PARIS VI - França, aumentando a característica acadêmica da IVG e possibilitando o acesso à documentações referentes ao projeto deste CI [MA78], o que facilita o estudo e o desenvolvimento de arquiteturas para controle de visualização gráfica à nível deste e de outros CIs.

Quanto à forma de descrição, o presente trabalho foi dividido em capítulos, sendo que o primeiro, o presente capítulo, corresponde à introdução.

O capítulo 2 faz uma apresentação quanto a definição de Sistemas Gráficos, bem como a sua composição; também, padronização de programação ("software") gráfica e aplicações desses sistemas são temas abordados.

Devido ao fato de se desejar a implementação física, o "hardware", de uma Interface de Visualização Gráfica, o capítulo 3 traz considerações de projetos deste tipo de interface.

Naquele capítulo são vistos assuntos como: sistemas de visualização interativos, em especial, o vídeo com tubo de raios catódicos (CRT) associado a dispositivos gráficos de entrada, o registrador de deslocamento e as RAMs (memórias de acesso aleatório) de Vídeo (VRAMs).

Depois, é descrito de que maneira, basicamente, uma interface de visualização pode ser utilizada, trazendo também, para uma interface dedicada a CRTs com exploração rastreada ("raster scan"), os tipos de circuitos integrados a partir dos quais podem ser baseadas estas interfaces: Controladores de CRT, Controladores ou Processadores Gráficos.

O capítulo 4 descreve a implementação da Interface de Visualização Gráfica, apresentando-se as principais características da interface e de seu componente básico - o Controlador Gráfico EF9367.

No final do capítulo, faz-se um detalhamento do projeto da interface desenvolvida, mostrando-se as características de construção.

A programação gráfica básica, dedicada à manipulação da interface, é apresentada no capítulo 5, cujo conteúdo principal é a descrição das características das funções primitivas gráficas implementadas.

O capítulo 6 apresenta uma comparação entre três maneiras comuns de se apresentar figuras, a fim de se observar qual tem melhor característica custo x desempenho.

As conclusões obtidas com este trabalho, bem como as propostas de continuação, também são encontradas naquele capítulo.

Finalmente, apresenta-se a bibliografia utilizada e apreciada durante este trabalho.

Apresenta-se, em anexo, a documentação em diagramas esquemáticos e de posicionamento de componentes, bem como fotos das placas que compõem a IVG.

CAPITULO 2 - SISTEMAS GRAFICOS

2.1) CONSIDERAÇÕES INICIAIS

A arte de criar gráficos e figuras através do uso de um computador é uma prática que vem sendo empregada a algum tempo.

Anteriormente, isso era praticado baseado em sistemas computacionais de uso geral, tornando assim a obtenção de resultados lenta. Também, não se conseguia grandes realizações, pois a geração de figuras complexas era bastante difícil e ficava muito comprometida pela falta de condições adequadas, necessitando-se para tanto de muito tempo e conhecimento para se confeccionar programas geradores de figuras.

Assim, houve a necessidade de se construir mecanismos voltados à obtenção de figuras, os quais facilitassem a geração e também a manipulação destas; estes deveriam fazer o papel de uma conexão entre um sistema computacional e seu usuário. Esses mecanismos foram criados e receberam o nome de SISTEMAS GRAFICOS.

A seguir é feita uma descrição de sistemas gráficos, mostrando-se como estes podem ser basicamente divididos, a fim de se localizar a interface de visualização desenvolvida (IVG) dentro destes sistemas (na figura 2.1 a IVG corresponde ao bloco dos dispositivos gráficos físicos). Outro ponto importante se refere à padronização do "software" gráfico, descrevendo-se os padrões gráficos existentes, inclusive o sistema GSFC - CORE, a partir do qual foi desenvolvido o sistema programacional gráfico, apresentado no capítulo 5. No final desta seção mostra-se em quais áreas se aplicam os sistemas gráficos.

2.2) DESCRIÇÃO DE UM SISTEMA GRAFICO

Sistema gráfico é o nome genérico dado a um mecanismo que faz a troca de informações gráficas entre um sistema computacional e o seu usuário.

Incorpora-se ao sistema todos os programas de manipulação dos aspectos de visualização das figuras. Esses programas têm a incumbência de representar graficamente, em um dispositivo dedicado de visualização, dados contidos em, por exemplo, programas de aplicação.

Os dispositivos dedicados de visualização, ou dispositivos gráficos de saída, compõem um meio de comunicação entre um sistema gráfico e seu usuário, assim como os dispositivos de entrada, que auxiliam na interação do homem com a máquina, à medida que as figuras são geradas.

O sistema gráfico é composto, basicamente, de "software" e "hardware" dedicados. No "software" estão incluídos programas para a manipulação de entrada, saída e armazenamento de informações, e programas para processamento dessas informações, chamados Núcleos de Processamento Gráfico - NPG. O "hardware" compreende dispositivos de vídeo, sistemas mecânicos de canetas ("plotters"), arquivos gráficos especiais, etc., e os circuitos lógicos dedicados.

Dentre os programas de um sistema gráfico existem aqueles que fazem o interfaceamento entre os componentes desse sistema, chamados de interfaces [CA86 e MA86]. A figura 2.1 mostra um diagrama de blocos de um sistema gráfico, composto de: programa de aplicação, núcleo de processamento gráfico (NPG) e

dispositivos gráficos físicos. Mostra-se, também, as duas interfaces: a de aplicação e a de dispositivos.

Figura 2.1: Diagrama de blocos de um sistema gráfico.

Basicamente, estas interfaces podem ser descritas como [SK86]:

- Interface de Aplicação: faz o interfaceamento entre o programa de aplicação e o NPG, assegurando troca adequada de informações entre o programa do usuário, que faz uso das rotinas contidas no NPG, e o NPG, ou seja, permite a adaptação deste programa a tal núcleo [SP86]; e

- Interface de Dispositivo: faz a interface do NPG com os dispositivos gráficos físicos, dispositivos estes que apresentam ou armazenam os dados gráficos processados pelo NPG.

2.3) PADRONIZAÇÃO GRAFICA

Quando se deseja gerar figuras em um sistema computacional convencional, o programa de aplicação necessita conter instruções

e dados específicos num formato que o dispositivo gráfico de saída possa "entender"; ou seja, a geração é totalmente dependente do conhecimento que o programador possui a respeito desse dispositivo.

Com a crescente difusão dos sistemas gráficos, pode-se construir programas de aplicação sem a necessidade de se conhecer em detalhes os dispositivos de saída utilizados.

Então, os fabricantes de sistemas computacionais passaram a oferecer aos usuários programas especiais, que fazem um interfaceamento entre o dispositivo de saída e o programa de aplicação, chamados pacotes gráficos.

Esta idéia evoluiu até os dias de hoje onde o pacote gráfico, tratado aqui como núcleo de processamento gráfico (NPG), tornou-se independente do fabricante, dos tipos de dispositivos e da linguagem de programação, oferecendo um conjunto de primitivas padronizadas a nível do programa de aplicação.

A padronização gráfica preocupa-se com o desenvolvimento de interfaces de "software" gráfico independentes do "hardware" e "software" básicos, e com o desenvolvimento de núcleos de "software", visando a compatibilidade com outros núcleos, a portabilidade destes, a comunicação transparente entre os vários equipamentos de um sistema, o desenvolvimento de programas aplicativos independentes dos periféricos, e fácil manutenção e evolução.

A portabilidade proposta por um núcleo gráfico pode ser dividida em portabilidade de programa, de programador, de

dispositivo e de dados, sendo que:

- portabilidade de programa descreve o fato de se poder transportar aplicações gráficas de uma instalação para outra com mínima ou nenhuma alteração no programa aplicativo;
- portabilidade de programador é o fato de um programador se deslocar de um sistema para outro com mínimo ou nenhum treinamento adicional; e
- portabilidade de dispositivo e de dados se referem à possibilidade de se utilizar a mesma aplicação gráfica em qualquer dispositivo e à independência em relação aos dispositivos de armazenamento, respectivamente.

A padronização gráfica foi tão necessária que várias entidades internacionais criaram grupos para estudar e padronizar sistemas gráficos.

A seguir descreve-se os núcleos existentes, padronizados ou em fase de padronização.

2.3.1) GSPC - CORE ("Graphic Standards Planning Committee" - CORE)

Em 1977, o GSPC ("Graphic Standards Planning Committee") liberou a primeira versão de uma proposta de padronização gráfica, o sistema CORE da SIGGRAPH ("Special Interest Group in Computer Graphics") da ACM ("Association for Computing Machinery"), e em 1979 uma segunda versão deste sistema foi apresentada.

O sistema CORE, descrito em [MA86], é um conjunto de rotinas gráficas que faz o interfaceamento entre um programa

aplicativo e o "hardware" gráfico. Programas aplicativos usando rotinas do sistema CORE, de acordo com HARRINGTON em [HA83], possuem alta independência de dispositivos. CORE contém primitivas gráficas básicas a partir das quais rotinas gráficas mais complexas ou de propósitos especiais podem ser construídas. Os objetos podem ser manipulados por programas aplicativos em duas ou três dimensões.

CORE é o precursor de padrões gráficos formais de hoje.

2.3.2) GKS ("Graphical Kernel System")

O padrão gráfico GKS ("Graphical Kernel System"), encontrado em [HO83, MA86 e TC84], foi apresentado em 1977 pelo DIN ("Deustches Institut fur Normung"), como a versão número 1. Em 1982, a versão 7.0 foi aprovada e adotada pela ISO ("International Standards Organization") como padrão gráfico internacional.

O GKS tem muitos conceitos semelhantes ao CORE, pois é considerado uma evolução deste, sendo bem mais detalhado [VA86].

A descrição das figuras ocorre no espaço bidimensional, sendo os desenhos constituídos por primitivas de saída (funções gráficas) que podem ser agrupadas formando segmentos.

Segmento é uma parte, um objeto, de um quadro total a ser apresentado, possuindo identificação (nome) e atributos próprios. Atributos são elementos que descrevem características. Aos segmentos podem ser aplicadas operações, tais como: criação, transformação, renomeação, etc.

Os atributos de segmentos no GKS (visibilidade, cintilamento, etc.) podem ser alterados dinamicamente [HO83 e

TC84].

No GKS existe o conceito de Estação de Trabalho - ET ("Workstation"), que é um conjunto de dispositivos gráficos tratados como uma unidade lógica, formado por um dispositivo de saída e zero ou mais dispositivos de entrada [TC84].

A característica principal da Estação de Trabalho é que ela é específica para determinados dispositivos; por exemplo, ET constituída por um "plotter" ou um digitalizador. Essa característica faz com que se possa acessar um ou mais dispositivos gráficos a qualquer momento, e também pode-se mostrar partes separadas de uma figura total em várias ET's [H083].

Na figura 2.2 apresenta-se um exemplo de um quadro dividido em partes, mostradas em várias ET's.

Figura 2.2: Visualização de partes separadas de um quadro em várias ET's.

Através do conceito de ET o usuário tem acesso a quaisquer dispositivos gráficos de forma uniforme.

No Brasil, em 1986, a ABNT (Associação Brasileira de Normas Técnicas), através do Comitê Brasileiro de Computação e Processamento de Dados CB - 21, adotou o GKS como padrão para programação gráfica.

2.3.3) GKS - 3D ("Graphical Kernel System" - 3D)

Um terceiro núcleo, descrito em [MA86 e PU86], é o GKS-3D. Este é uma simples e bem feita extensão do GKS, que é bidimensional, para condições tridimensionais. Como enfatizado em [PU86], tentou-se assegurar completa compatibilidade com o GKS, de forma que programas GKS possam rodar sem modificações nas condições GKS-3D. Os dados manipulados por GKS-3D são tridimensionais, mesmo aqueles descritos em duas dimensões. Funções bidimensionais são convertidas para três dimensões antes de serem processadas; por exemplo, coordenadas bidimensionais são convertidas para três dimensões, fazendo-se seu componente Z igual a zero, automaticamente. Desta forma, funções GKS e GKS-3D podem ser misturadas livremente, conforme [PU86].

Além das extensões do GKS, o GKS-3D possui outras características que o tornam um sistema com grande capacidade de manipulação de figuras tridimensionais.

2.3.4) PHIGS ("Programmer's Hierarchical Interactive Graphics System")

Já o PHIGS ("Programmer's Hierarchical Interactive Graphics System"), apresentado em [MA86 e SH83], visa o interfaceamento

entre um programa de aplicação e um sistema gráfico que controla dispositivos dedicados, sendo muito útil para aplicações complexas bi- e tridimensionais em uma condição altamente interativa. PHIGS é uma interface que trabalha com dados gráficos hierárquicos (em organização hierárquica estruturas inferiores herdam atributos das superiores), fazendo com que aplicações multinível sejam facilmente tratadas por sistemas gráficos, e de maneira interativa, fornecendo facilidades que permitem modificação e interpretação dinâmicas das bases de dados gráficos.

Em PHIGS, os dados gráficos são organizados em unidades chamadas estruturas, que vêm a ser um mecanismo para se agrupar os elementos básicos (elementos gráficos padrão), como primitivas de saída, rótulos, dados de aplicação, etc. As estruturas em PHIGS, assim como os segmentos em GKS, contêm primitivas gráficas e atributos para a definição da figura.

Algumas áreas bem adequadas para o uso de PHIGS são CAD, e controle e monitoração de processos.

2.3.5) CGI ("Computer Graphics Interface")

CGI ("Computer Graphics Interface"), em [MA86 e P086], é um poderoso e complexo núcleo que, de acordo com [B086 e P086], basicamente, pode ser usado de duas maneiras: como acesso direto a um dispositivo ou como uma interface dentro de um sistema gráfico.

[P086] descreve que, quando se está trabalhando com um sistema pequeno, como um PC (computador pessoal) ou uma Estação de Trabalho com habilidade local de processamento, o CGI pode ser

implementado como um recurso disponível para trabalhar como uma conexão direta entre o usuário e o dispositivo, desta forma, sendo usado como uma interface por programadores de aplicativos.

Outra aplicação corresponde a utilizá-lo como um recurso oculto para ser usado somente por implementadores de sistemas; ou seja, quando se tem disponível uma interface de maior nível, como GKS, CGI se torna uma interface para o dispositivo, controlando a comunicação entre o sistema e esse dispositivo.

Segundo BONO [BO86], o CGI é o primeiro padrão a incorporar primitivas, atributos e funções de controle com o propósito expresso de controlar dispositivos "raster" bit-mapeados.

WILLIAMS em [WI86c] salienta que, o CGI permite o uso direto de conjuntos de "pixels" ("pixel" é a informação correspondente a cada ponto luminoso do vídeo), ou operações de transferência de blocos de "pixels". Estas são operações lógicas e de movimento de conjunto de "pixels" em torno de um mapa de bits. O CGI pode, também, definir mapas de bits dentro e fora do espaço de memória destinado a tela - uma capacidade muito importante para implementar condições de janelas (janela é uma técnica usada para selecionar partes de uma figura completa).

2.3.6) CGM ("Computer Graphics Metafile")

Outro núcleo é o CGM ("Computer Graphics Metafile"), em [HE86 e MA86], que é utilizado para padronizar o formato de armazenamento dos dados gráficos, proporcionando um mecanismo para a "captura", armazenamento e transporte destes dados gráficos, de forma independente dos dispositivos onde estes dados

serão armazenados e apresentados. É descrito em [HE86] que, CGM pode ser considerado como a composição de um gerador e de um leitor/interpretador de dados gráficos em um "metafile".

MAGALHAES, em [MA86], define "metafiles" gráficos como arquivos em formatos padrões para armazenamento a longo prazo para transferência de informação gráfica, ou entre sessões de trabalho ou entre sistemas de computação gráfica.

2.3.7) DGIS ("Direct Graphics Interface Specification")

Existe ainda um NPG chamado DGIS ("Direct Graphics Interface Specification") que, de acordo com [CL86 e WI86b], é um núcleo especialmente destinado a interfacear aplicações PC-DOS diretamente com dispositivos gráficos.

DGIS tem, como salientado em [OH86 e WI86a], um conjunto de comandos que executa controles (como limpar a tela e liga/desliga cursor), operações de traçado para primitivas gráficas (círculos, arcos, múltiplas linhas - "polyline", comandos de preenchimento e conjuntos de "pixels"), controle de atributos (tal como preenchimento com cor), e perguntas vindas do processador principal, denominado "host", para a tela.

DGIS capacita programadores a escrever programas gráficos genéricos, facilitando a comunicação dos programas aplicativos com o "hardware" dedicado.

WILLIAMS em [WI86a] observa que alguns fabricantes de CI's (circuitos integrados) gráficos estão adaptando seus produtos a DGIS, entre eles INTEL e TEXAS.

2.4) APLICAÇÕES DE SISTEMAS GRAFICOS

Os sistemas gráficos estão sendo aplicados basicamente em duas áreas distintas: O Processamento de Imagens e a Computação Gráfica.

O Processamento de Imagens consiste em analisar, manipular e apresentar dados digitalizados de informações relacionadas a objetos reais, contidos na memória dedicada de um sistema gráfico. Um mecanismo específico se incumba de captar tais informações, digitalizá-las e inseri-las na memória dedicada para posterior análise e processamento.

O processamento de imagens tem por finalidade a observação e o exame não destrutivo de informações.

Seu objetivo é enfatizar partes que estão em alteração e, em especial, identificar contornos e bordas para reconhecimento visual ou por computador.

A análise de dados coletados remotamente por satélites é uma das suas aplicações. Dentre os usuários se incluem: serviços meteorológicos, observações geológicas e aplicações na agricultura.

Outra área que tem recebido grande colaboração, em relação a aplicações de processamento de imagens, é a medicina.

A Computação Gráfica é a prática de criar, armazenar e manipular modelos de objetos e cenas através do computador. Isso é feito através de métodos geométricos (processamento geométrico).

Com o auxílio de comandos especiais, como para traçar ponto,

linha, etc., consegue-se criar figuras para posterior manipulação dos dados pelo sistema.

A computação gráfica está sendo empregada, hoje, em praticamente todas as áreas. A evolução da sua aplicação varia desde a geração de gráficos simples, mapas e tabelas, coloridos ou em preto e branco, passa pela utilização especializada em indústrias automotivas e aeroespaciais, através do processo CAD (projeto auxiliado por computador), e em seguida pelos projetos de circuitos elétricos. Hoje tem-se suas aplicações voltadas para a arte, educação, arquitetura, engenharia e tantas outras.

A computação gráfica está sendo utilizada, também, em simulações de condições reais, como, por exemplo, de voo para treinamento de pilotos e simulação de combate para aplicações militares. Também é encontrada na área de animação, como por exemplo em filmes cinematográficos, programas e propagandas de televisão.

Segundo MACHOVER, em [MA77], as cinco maiores áreas de aplicação da computação gráfica, cada uma com seus próprios requisitos para "hardware" e "software", são:

2.4.1) Gerenciamento de Informação:

Sistemas para gerenciamento de informação oferecem ajuda na tomada de decisões de acordo com informações armazenadas previamente na base de dados do computador, e que podem ser mostradas em um grande número de formatos úteis, os quais podem ser modificados quando necessário para ajudar na análise de tendências. Estes formatos podem ser mapas, gráficos, tabelas, etc.

2.4.2) Gráficos Científicos:

Os sistemas para visualização de gráficos científicos são usados em tarefas de traçados bidimensionais e tridimensionais. Exemplos destas tarefas são análise de forma de onda, ajuste (adaptação) de curvas, visualização de funções, projeto e análise interativos.

2.4.3) Comandos e Controles:

Um elemento chave na operação bem sucedida de sistemas muito grandes é um centro de controle a partir do qual subsistemas selecionados podem ser monitorados e controlados. As maiores aplicações são em utilizações elétricas, e também em sistemas militares os quais usam gráficos para aplicações tão complexas como simulações de condições de combate.

2.4.4) Geração de Gráficos em Tempo Real:

Tal sistema é utilizado em situações onde se deseja um alto grau de realismo na geração de gráficos, principalmente pelas suas alterações rápidas.

Equipamentos para treinamentos militares e aeroespaciais - particularmente para astronautas - são os maiores beneficiados com esse sistema, e praticamente os únicos que podem tê-lo, devido ao seu alto custo financeiro.

Na atualidade, porém, os custos desses sistemas têm caído e algumas empresas aéreas passaram a utilizá-los como simuladores de voo no treinamento de seus pilotos.

2.4.5) Projetos Elétricos e Mecânicos:

O sistema CAD ("Computer-Aided Design"), que auxilia nos esboços e projetos gerais, pertence a uma área da computação gráfica onde se obtém notórias diferenças de custo e tempo em relação a técnicas manuais.

Além de muito usados na indústria automobilística e aeroespacial, eles têm, também, grande emprego no projeto de circuitos impressos, circuitos integrados, áreas de produção, arquitetura, engenharia, etc.

CAPITULO 3 - PROJETO DE INTERFACES DE VISUALIZAÇÃO GRAFICA INTERATIVAS

3.1) CONSIDERAÇÕES INICIAIS

Uma Interface de Visualização Gráfica é um dos componentes de um sistema gráfico. Esta é um dispositivo que faz a conexão entre os sistemas computacional e de visualização.

A figura 3.1 mostra um possível diagrama de blocos de um dispositivo gráfico físico, onde se nota a presença da interface de visualização gráfica (IVG).

Figura 3.1: Diagrama de blocos de um dispositivo gráfico físico.

Esta interface pode ser utilizada, basicamente, como um mecanismo para, simplesmente, transferir dados contidos em uma memória dedicada a figuras, chamada Memória Gráfica, para um sistema de visualização, ou como um dispositivo para processar estes dados e, então, enviá-los ao sistema de visualização.

Os sistemas de visualização, ou dispositivos gráficos de saída, são utilizados para a apresentação das informações gráficas ao usuário, podendo fazê-lo em uma forma dinâmica, ou não, dependendo da finalidade do sistema gráfico que estes

incorporam.

Um exemplo de um dispositivo gráfico de saída dinâmico é o monitor de CRT (tubo de raios catódicos), e de um não dinâmico é o sistema mecânico de canetas ("plotter").

O projeto de uma interface de visualização gráfica depende, entre outras coisas, das condições de trabalho encontradas, e de qual será a finalidade do sistema gráfico que esta comporá.

3.2) SISTEMAS DE VISUALIZAÇÃO INTERATIVOS

Um sistema de visualização pode ser considerado como um dispositivo para traçar (mostrar) figuras.

Este pode ser permanente, ou seja, um sistema que não permite a manipulação das figuras pelo usuário; este simplesmente mostra os dados gráficos em um formato fixo, não permitindo alterações na figura descrita. Tal sistema pode ser usado quando se deseja obter a figura, relativa aos dados gráficos, em uma forma transportável.

Um dispositivo gráfico de saída dinâmico permite alterações na figura mostrada, sobre a figura total ou sobre determinadas partes desta.

Os sistemas de visualização permanentes são bastante lentos em comparação com os dinâmicos, devido às suas características mecânicas de construção.

Vários tipos de dispositivos podem compor um sistema de visualização; entre eles encontram-se: sistema mecânico de canetas, monitores de vídeo, registradores de filmes, e

projetores para telas grandes.

Pode-se ter, ainda, sistemas de visualização em forma de matriz: painel de plasma, e os constituídos por LEDs (diodos emissores de luz), cristal líquido ou filme fino.

Os sistemas mecânicos de canetas e os registradores em filmes são dispositivos primários para produzir gráficos permanentes.

Os dispositivos de visualização dinâmicos, tais como CRTs e painéis de plasma, são úteis em aplicações onde se faz necessária a amostragem de informações dinâmicas em tempo real, como em simulações de voo, observação de arquivos de dados, vigilância e sistemas industriais.

Os dispositivos de saída dinâmicos são os mais utilizados pelo fato de serem interativos, mais rápidos na amostragem dos dados gráficos e, também, em geral, pelos seus menores custos.

3.2.1) O VIDEO COM TUBO DE RAIOS CATODICOS

Uma forma de apresentação (visualização) de imagens e gráficos é através dos videos com tubos de raios catódicos (CRT) [KA80].

Um CRT é projetado para produzir, acelerar e focalizar um feixe de elétrons, e desviá-lo quando desejado.

A figura 3.2 mostra um diagrama genérico de um CRT, com os seus principais componentes.

Figura 3.2: Representação de um CRT.

O canhão de elétrons (catodo) emite um feixe de elétrons, o qual é acelerado em direção à cobertura de fósforo.

Os circuitos de deflexão, horizontal e vertical, fazem o controle de localização do ponto na qual o feixe de elétrons atinge a tela, desviando tal feixe. Quando os elétrons atingem a tela, o fósforo emite luz visível, dando origem ao ponto. A localização do ponto é proporcional às voltagens aplicadas às entradas desses circuitos.

A quantidade de luz emitida pelo fósforo decai exponencialmente com o tempo; desta forma, a figura mostrada precisa ser retracada (reavivada) muitas vezes por segundo, para que o usuário não perceba o cintilamento ("flickering") da figura.

Conforme descrito no capítulo 2, "pixel" ("picture element") é um termo usado para descrever a informação mostrada em um ponto unitário do vídeo. Este pode ser codificado em bit único ou em

múltiplos bits. O "pixel" é então decodificado em sinal analógico e enviado ao monitor de CRT através do sinal de VIDEO, que controla a intensidade do feixe de elétrons.

Pode-se encontrar, basicamente, três tipos de dispositivos de vídeo CRT: o tubo de armazenamento de visualização direta ("Direct View Storage Tube" - DVST), o CRT com feixe direcionado ("Direct Beam CRT"), e o CRT com exploração rastreada ("Raster Scan CRT").

O DVST

O DVST é um CRT que não necessita ser reavivado. A figura é armazenada do lado de dentro do DVST como uma distribuição de cargas sobre a superfície interna da tela. A figura 3.3 mostra um diagrama simplificado de um DVST.

Figura 3.3: Diagrama de um DVST.

O canhão de elétrons, com o feixe de escrita, cria a figura a ser mostrada, a figura armazenada, e os canhões de amostragem tornam a figura armazenada visível.

Tipicamente, o DVST possui baixa velocidade de interfaceamento e mínimas características de alterações na figura; o DVST permite o acréscimo de novos componentes à figura; mas, se for desejado fazer uma modificação parcial (retirada de algum componente) da figura armazenada, essa terá que ser totalmente apagada e retracada por inteiro, devido às características do DVST. O nível de interatividade com um DVST é mais baixo que com os outros dois tipos de CRT. O DVST apresenta, também, luminância e contraste baixos.

O CRT CALIGRAFICO

O CRT com feixe direcionado, ou CRT de exploração aleatória, ou, simplesmente, CRT caligráfico, move o feixe de elétrons em direções arbitrárias [LU78]. O feixe é movido diretamente para o ponto onde o elemento gráfico será mostrado - com o feixe desligado - e então é desviado para traçar o elemento desejado - com o feixe ligado. A figura 3.4 mostra como um traçado é feito por um CRT caligráfico, onde as setas ($\rightarrow$) indicam a sequência de movimentos do feixe de elétrons.

Figura 3.4: Traçado feito por um CRT caligráfico.

O sistema caligráfico oferece resolução alta. Por outro lado, possui capacidade de cor limitada e a informação (a figura)

mostrada não é totalmente livre de cintilamento (tem baixo "flicker-free").

Uma resolução típica para o CRT caligráfico é de 1024 X 1024 pontos visíveis.

O CRT COM EXPLORAÇÃO RASTREADA

O tipo de sistema de CRT que mais se utiliza é o com exploração rastreada ("raster scan"), o mesmo que é usado nos televisores comerciais e em muitos terminais de CRT projetados para mostrar caracteres alfanuméricos [BL82]. (A interface de visualização gráfica projetada utiliza como dispositivo de visualização um vídeo com exploração rastreada.)

Um típico monitor de CRT com exploração rastreada necessita, basicamente, de três sinais para o seu funcionamento: sincronização horizontal (HSYNC), sincronização vertical (VSYNC) e sinal de vídeo (VIDEO). Estes sinais são aplicados aos circuitos do monitor de CRT para fazer o controle das deflexões, horizontal e vertical, e de intensidade do feixe de elétrons.

Neste sistema o feixe de elétrons que varre a tela, em geral da esquerda para a direita e de cima para baixo, provoca o aparecimento de linhas de informações horizontais nesta, as quais irão formar as figuras na tela. Tais linhas são tratadas como linhas de varredura ("scan-lines"), como visto na figura 3.5.

Figura 3.5: Representação de linhas de varredura em um monitor com exploração rastreada.

As linhas tracejadas da figura 3.5 representam as chamadas porções de retraço, horizontal e vertical, do modelo com exploração rastreada, que vêm a ser os retornos do feixe de elétrons durante a varredura da tela; estas linhas não são visíveis ao observador devido à reduzida intensidade do feixe de elétrons.

Os sinais de sincronização (HSYNC e VSYNC) controlam a frequência dos movimentos de explicitação do feixe de elétrons na tela do CRT, uma vez que a explicitação é usualmente contínua, de tal forma que se possa sincronizar a apresentação de informação através do feixe de elétrons. A intensidade do feixe de elétrons é modulada pelo sinal de VIDEO.

Alguns monitores com exploração rastreada possuem uma única entrada de sinais. Este sinal de entrada, chamado VIDEO COMPOSTO, inclui HSYNC, VSYNC e VIDEO. Nesse caso, o monitor de CRT tem um circuito para separar os três sinais que compõem o sinal de VIDEO COMPOSTO. A grande vantagem do sinal de VIDEO

COMPOSTO sobre três sinais de entrada separados é que este sinal pode ser enviado através de um único cabo coaxial, conveniente onde se utiliza conexão a longa distância.

O sistema com exploração rastreada, segundo BLINN em [BL82], não apresenta uma resolução muito alta (resolução típica é de 512 X 512 pontos); em compensação oferece excelente apresentação de cores e figuras altamente livres de cintilamento (alto "flicker-free"), e, com a produção em massa, é muito mais barato que os outros tipos de monitores (DVST e caligráfico).

Tanto no sistema com exploração rastreada como no caligráfico existe a necessidade de se reavivar a tela, reescrever os dados, em intervalos constantes, pelo fato desta reter a informação apenas temporariamente.

Uma memória, chamada memória gráfica, contendo as informações que serão apresentadas, pode ser utilizada em associação ao vídeo para facilitar o reavivamento da tela.

3.2.2) CRT COLORIDO

As cores, em um sistema CRT, são conseguidas construindo-se a sua tela com fósforos vermelho, verde e azul, diferente do monocromático que apresenta apenas um tipo de fósforo.

Os pontos podem ser tríadas (conjuntos de três pontos) de fósforos, ou conseguidos a partir de faixas de fósforos, vermelho, verde e azul; quando três feixes de elétrons atingem os fósforos estes emitem luzes de suas respectivas cores.

Embora os CRTs operem com vermelho, verde e azul, uma grande variedade de outras cores podem ser reproduzidas por combinações

destas três, inclusive o branco.

Os três feixes de elétrons são desviados sincronamente e são focalizados sobre o mesmo ponto (triáda) na tela. Antes de chegar à tela, os feixes passam por uma máscara de sombra, que possui três pequenos orifícios para cada ponto. Os orifícios são precisamente alinhados com respeito ao ponto e aos canhões de elétrons, tal que cada fósforo do ponto é exposto a elétrons vindos de somente um canhão. Assim, a intensidade de elétrons em cada feixe controla a quantidade de luz vermelha, verde e azul gerada pelo ponto.

A necessidade da máscara de sombra e triádas impõe um limite sobre a resolução dos CRTs a cores.

A seguir, são descritos dois exemplos de monitores coloridos com suas respectivas resoluções, e que dão uma idéia numérica quanto a tal limite para vídeos a cores disponíveis comercialmente.

Um exemplo é o "Terminal Gráfico Colorido 4111" da Tektronics, de 19 polegadas, que tem uma resolução de 1024 X 768 pontos visíveis, com uma seleção de 16 cores de um conjunto total ("palette") de 4096. O outro é a tela colorida utilizada pela "Estação de Trabalho Gráfica 4125", também da Tektronics, de 19 polegadas, com 1280 X 1024 pontos visíveis, proporcionando uma seleção de 64 cores em 16 milhões.

Em CRTs de resolução muito alta, as triádas são colocadas em centros de aproximadamente 0,35 mm, enquanto em tubos de TVs, estas são colocadas em pontos de 0,60 mm de diâmetro, conforme

descrito em FOLEY em [F082].

Além da limitação em resolução, os vídeos coloridos podem apresentar um outro inconveniente, por utilizarem pontos com posicionamentos definidos.

Quando se utiliza monitores coloridos em aplicações gráficas, os "limites" de cada cor têm que ser bem definidos, conforme explanação seguinte.

Por exemplo, supondo-se que se deseje traçar três pontos distintos na tela: um vermelho (R), um verde (G) e o terceiro azul (B).

Para se traçar o ponto R, somente o fósforo vermelho da triada correspondente deverá ser ativado pelos elétrons; o mesmo deverá ocorrer para os outros dois pontos, com relação aos seus respectivos fósforos.

Se assim ocorrer, poderá se notar a perfeita nitidez de cada ponto, cada um com sua respectiva cor.

Isso é conseguido utilizando-se uma frequência de "pixels", ou seja, frequência de transferência de dados do sistema gráfico para o monitor de CRT, idêntica àquela que proporciona uma resolução horizontal igual à do vídeo, a frequência de triadas.

Se for utilizada uma frequência de "pixels" maior que a de triadas, poderá ocorrer alguns problemas em relação a apresentação dos três pontos.

Por exemplo, supondo-se que a frequência de "pixels" seja igual a três vezes a de triadas. Então, ao invés de se ter três pontos distintos (R, G e B), poderá surgir apenas um ponto na tela, com uma cor que deverá ser uma combinação das outras três.

3.2.3) DISPOSITIVOS GRAFICOS DE ENTRADA

Para se tornar interativo, um sistema gráfico deve fornecer recursos especiais ao usuário, de modo que este possa interagir com o sistema de visualização, alterando as figuras mostradas ou o seu posicionamento na tela.

Os mecanismos que proporcionam tal interação são conhecidos como dispositivos gráficos de entrada. Através destes dispositivos novas informações podem ser introduzidas no sistema gráfico, de modo a provocar alterações na figura mostrada.

Entre os vários dispositivos gráficos de entrada disponíveis incluem-se: teclados, teclas de funções, caneta ótica ("light pen"), digitalizador ("tablet"), "joystick", diais, "mouse" e "control ball".

Cada um desses dispositivos pode ser tratado como a realização física de uma ou mais das quatro classes de dispositivos virtuais, assim considerados por FOLEY em [F082]: seletor ("pick"), tecla ("button"), localizador ("locator") e avaliador ("valuator").

Descreve-se a seguir, em uma forma sucinta, estas quatro classes de dispositivos virtuais:

- seletor: utilizado como dispositivo de apontamento; é usado

para apontar objetos e símbolos definidos pelo usuário, como linha, resistor, curvas, etc., mostrados na tela. Como exemplo de seletor tem-se a caneta ótica ("light pen");

- tecla: utilizado como dispositivo para selecionar comandos, caracter ou opções de "menu" pré-definidos; é implementado através de teclados de funções programadas, os quais tipicamente contêm de 16 a 32 teclas. Exemplos desta classe são: teclas de funções e teclados;

- localizador: usado como dispositivo de posicionamento; serve para indicar uma localização e/ou a orientação no espaço de traçado. Exemplos de localizador são: digitalizador, "joystick", "mouse" e "control ball";

- avaliador: usado para a entrada de constantes; é um dispositivo para determinar um único valor no espaço dos números reais. Um potenciômetro é um avaliador clássico. Exemplos dessa classe são: diais de controle e potenciômetros deslizantes.

3.3) O REGISTRADOR DE DESLOCAMENTO DE "PIXELS"

Quando se trabalha com um sistema de visualização que utiliza reavivamento de tela, como é o caso daquele que trabalha com exploração rastreada, existe a necessidade de se transferir dados entre a memória gráfica e o sistema de visualização em taxas muito elevadas.

Para tanto, deve-se providenciar um mecanismo que faça tal transferência rapidamente.

Tal mecanismo poderia ser um simples registrador de

deslocamento, que receberia os dados da memória gráfica em paralelo e os enviaria serialmente para o vídeo, conforme figura 3.6.

Figura 3.6: Esquema de transferência de informação através de um registrador de deslocamento.

Esse registrador deve ser bastante rápido no envio de dados ao vídeo, de tal forma que a atualização da tela ocorra sem atrasos perceptíveis ao usuário, evitando o surgimento de cintilamento da figura mostrada.

Esse registrador de deslocamento pertenceria a uma das famílias de circuitos lógicos existentes, dependendo do tipo e da resolução do vídeo a ser utilizado.

Quanto maior for a resolução do vídeo, maior será o número de pontos da tela, e, portanto, mais rápido deverá ser o registrador de deslocamento.

Para facilitar o entendimento deste ponto, observe-se o exemplo que se segue.

Se for utilizado um monitor de CRT com exploração rastreada com uma resolução de 520 pontos por 525 linhas, com varredura entrelaçada, será usado para o sincronismo horizontal, HSYNC, um valor igual a 15.750 Hz; obtém-se, daí, um tempo de linha de $1/15.750$, que é aproximadamente igual a $63,5 \mu s$.

Supondo que o tempo de retraço horizontal ocupa aproximadamente 17% do tempo de linha, tem-se:

$$17\% \times 63,5 \approx 11 \mu s.$$

Sendo assim, o tempo de apresentação da linha está em torno de $52,5 \mu s$.

Se este monitor tem uma resolução horizontal de 520 pontos, então, o tempo necessário para apresentar um único ponto é de:

$$52,5 \mu s / 520 \text{ pontos} \approx 100 \text{ ns/ponto}.$$

Isso significa que o registrador de deslocamento deverá enviar um dado, relativo ao ponto, a cada 100 ns para o monitor.

Nesse caso, deverá ser usado um registrador de deslocamento que possa transferir um dado num tempo menor que 100 ns, por exemplo, 20 ns.

Mas, se for utilizado um monitor de CRT que possui, como única diferença em relação ao anterior, uma resolução horizontal de 5200 pontos, obtém-se um tempo de apresentação de ponto de:

$$52,5 \mu s / 5200 \text{ pontos} \approx 10 \text{ ns}.$$

Ou seja, o registrador de deslocamento deste exemplo deverá enviar um dado 10 vezes mais rápido que o do exemplo anterior.

Sendo assim, para garantir o perfeito funcionamento do sistema, deverá ser usado um registrador com tecnologia mais rápida que a anterior, uma vez que o novo tempo de apresentação de cada ponto (10 ns) é menor que o de transferência do registrador exemplificado (20 ns). Nota-se, então, que o registrador de deslocamento de "pixels" é uma parte crítica do projeto de interfaces de visualização, dependendo de sua

resolução.

3.4) AS VRAMs

Também os CIs (circuitos integrados) que compõem a memória gráfica são um outro fator de preocupação.

Quando se trabalha com um vídeo de resolução não muito alta, a memória gráfica pode ser construída com RAMs dinâmicas (DRAMs) convencionais.

Mas, com o aumento na resolução do vídeo podem surgir alguns inconvenientes ao se projetar memórias gráficas com DRAMs convencionais; os principais são: o alto custo para se obter a devida densidade de armazenamento, a baixa "bandwidth" (taxa de transferência de informações no tempo) e o gargalo de memória.

O gargalo ("bottleneck") de memória em sistemas gráficos é um problema que surge quando se deseja, ao mesmo tempo, transferir continuamente os dados da memória dedicada para o CRT, para o reavivamento da tela, e abordar os bits desta memória, para a atualização da imagem, por algum elemento processador.

Uma maneira de se eliminar o gargalo é através de memória de acesso dual (dois portos de acesso); um acesso permite a entrada de dados em paralelo para atualizar a imagem na memória, e o outro permite a leitura e consequente geração da cadeia serial de dados para reavivar a tela, conseguindo-se, assim, um significativo aumento no seu desempenho [GU86].

Salienta-se em [GU86] que, embora se obtenha uma razoável densidade de armazenamento, com baixo custo, construindo-se

memória gráfica com DRAMS convencionais para sistemas de visualização de média resolução, tais DRAMS, que são dispositivos de porto único, não proporcionam a necessária "bandwidth".

Então, recentemente criou-se elementos de memória de dois portos, chamados VRAMs ("Video RAMs"), para eliminar o gargalo, fazendo com que memórias gráficas construídas a partir destes tenham uma alta "bandwidth".

Basicamente, como descrito em [GU86], a VRAM é uma DRAM que foi modificada internamente pela adição de um grande registrador de deslocamento. A figura 3.7 é um diagrama de blocos de uma VRAM.

Figura 3.7: Diagrama de blocos de uma VRAM.

As VRAMs facilitam o projeto de memórias gráficas com alta "bandwidth", com um pequeno número de componentes, e com maior disponibilidade, aumentando a velocidade de traçado de textos "bit-mapeados"[NI88].

3.5) A INTERFACE DE VISUALIZAÇÃO GRAFICA

Como descrito anteriormente, uma interface de visualização gráfica faz a conexão entre um sistema computacional e dispositivos de visualização.

Descreveu-se, também, os dois modos básicos de utilização de uma interface: como um simples transferidor de dados da memória gráfica para o dispositivo de saída, ou como um mecanismo que processa a informação obtida na memória gráfica antes de apresentá-la para visualização.

A maneira como tal interface será utilizada e, portanto, construída, dependerá da finalidade do sistema gráfico que esta comporá, bem como de condições de trabalho e financeira existentes. Também, o tipo de sistema de visualização utilizado influenciará no projeto desta interface.

3.5.1) A INTERFACE PARA O SISTEMA COM EXPLORAÇÃO RASTREADA

Este item está relacionado a interfaces construídas para sistemas com exploração rastreada, justificado por três fatores importantes: sistema altamente interativo, oferece excelente apresentação de cores e é o mais comumente usado.

INTERFACE PARA CONTROLE DE CRTs

Se a interface a ser construída simplesmente controlará a apresentação de dados num vídeo, esta deverá fornecer sinais, alguns vitais, ao sistema, de modo a tornar eficaz e rápida a transferência de informações a serem visualizadas.

Entre os sinais a serem fornecidos, alguns são indispensáveis ao funcionamento correto do sistema com exploração

rastreada; entre estes encontram-se os de sincronização, horizontal e vertical, e o sinal de vídeo. Também o sinal "blanking" se faz necessário, uma vez que este alimenta a entrada de vídeo para desligar o feixe de elétrons durante os retraços horizontal e vertical do mesmo.

Um recurso que se deve fornecer, também, é um circuito endereçador da memória gráfica, para que o devido dado seja acessado e enviado para o vídeo.

Cada um destes sinais pode ser fornecido por circuitos eletrônicos independentes.

Estes sinais podem ser gerados por um único dispositivo, um circuito lógico integrado, denominado CONTROLADOR de CRT. Os circuitos geradores de tais sinais estão integrados dentro do controlador de CRT.

Com a utilização de um tal controlador como base para se construir a interface, obtém-se grande facilidade no seu projeto, um reduzido tamanho e, principalmente, um menor custo.

Além daqueles, um controlador de CRT traz, integrados a esse, outros recursos e sinais.

Entre estes recursos pode-se encontrar:

- Gerador de Caracteres: que facilita e agiliza a escrita de caracteres no vídeo; é constituído por uma ROM, que contém caracteres padrões, um registrador de deslocamento (com entrada paralela e saída serial) e um endereçador para a ROM, para selecionar o caractere desejado;

- Gerador de Cursor: lógica para gerar e mover um cursor, que é um sinal visível, no vídeo e, que facilita a manipulação de dados

no mesmo;

- Temporizador de Ponto: o sinal emitido por este é utilizado como um relógio para o registrador de deslocamento do gerador de caracteres, para transferir informações dos pontos deste registrador para o CRT;
- Contador de Linhas de Varredura: este trabalha em conjunto com o endereçador para a ROM do gerador de caracteres;
- Caneta Ótica: entrada para caneta ótica ("light pen"), que é capaz de captar o efeito luminoso na sua extremidade; a caneta ótica é utilizada como uma maneira de se coordenar a entrada de dados a um local desejado na tela;
- "Scrolling": lógica para implementar a operação "scrolling", a qual movimenta verticalmente os dados na tela; e
- Registradores Programáveis: dentre os registradores que podem ser incluídos aqui encontram-se: o do cursor, de "status" e da caneta ótica.

Dentro deste estilo de controladores de CRT pode-se citar o DP8350 da National Semiconductor, 8275 da Intel, 6845 da Motorola, 6545 da Synertek e o 5027 da SMC Microsystems Corporation [KA80].

INTERFACE QUE MANIPULA INFORMAÇÕES

Se a interface desejada também manipulará dados, esta deverá fazer mais do que simplesmente gerar sinais e recursos, como os citados anteriormente, para o perfeito funcionamento do sistema

com exploração rastreada. Esta deverá, também, fornecer recursos para a geração e manipulação, rápida e eficiente, de figuras no vídeo.

Tais recursos podem vir embutidos fisicamente, em "hardware", o que tornaria a interface mais veloz do que se implementados por programação, em "software".

Alguns CIs trazem integradas outras funções além das encontradas em controladores de CRT. Por exemplo, apresentam um controle total sobre a memória gráfica e outras partes de um sistema gráfico.

Estes CIs são controladores de CRT que passaram por modificações, e a esses foram incluídas algumas funções para o traçado de primitivas gráficas, tais como ponto e linha.

Estes CIs são comumente denominados CONTROLADORES GRAFICOS.

Dentre os controladores gráficos pode-se citar o EF9367 da Thomson Semiconductors, HD63484 da Hitachi America, 82720 da Intel, 7220 da NEC e o TMS34061 da Texas.

Os controladores gráficos executam tarefas dedicadas através de conjuntos de comandos fixos, que são algoritmos para traçado de ponto, linha e outras figuras geométricas, mais rapidamente que o processador principal, o "host". Desta forma, eles tiram do "host" a responsabilidade de executar tarefas gráficas de baixo nível, denominadas de primitivas gráficas [DE83, IN83, NE82, OL83].

Pelo fato das arquiteturas de alguns controladores gráficos terem muitas funções integradas, consegue-se grandes facilidades de interfaceamento para a memória gráfica, processador "host", sistema de visualização, etc., obtendo-se, assim, uma sensível redução no tamanho total do "hardware" da interface.

Um outro tipo de CI que pode servir como base para a construção de uma interface de visualização gráfica deste tipo é o chamado PROCESSADOR GRAFICO.

Um processador gráfico vem a ser um controlador gráfico que recebeu modificações na sua arquitetura a fim de melhorar algumas de suas características e acrescentar outras.

Em relação ao controlador gráfico, um processador gráfico possui, principalmente, maior programabilidade e flexibilidade, e é mais rápido, seguro e com maior capacidade de processamento.

Os CIs considerados como processadores gráficos são o TMS34010 da Texas, o 82786 da Intel e o Am95c60 da Advanced Micro Devices [AS86, JA87, MO88, NI87, RA86, RA86a, SH86, WI87, WI88].

O processador gráfico possui um conjunto de instruções gráficas (as vezes, adicionado de instruções de propósito geral) bem maior que o controlador gráfico, e é capaz de fazer a busca ("fetch") de instruções, colocadas em uma memória dedicada, e executá-las; isso faz com que a sua programabilidade seja muito alta, e que trabalhe quase independentemente do "host", livrando este de muitas de suas responsabilidades para gerenciamento gráfico.

Alguns circuitos de interfaceamento com outros subsistemas (visualização, memória gráfica e "host") do "hardware" são

integrados aos processadores gráficos. Isso faz com que a interface de visualização gráfica, construída a partir de processadores gráficos, tenha um número reduzido de componentes, diminuindo os custos no seu projeto, e tornando-a de grande desempenho.

Outra característica muito importante, encontrada nos processadores gráficos, é o fato desses trabalharem tanto com RAMs dinâmicas convencionais como com RAMs de Video, descritas anteriormente.

Também, os processadores gráficos são capazes de manipular blocos de "pixels" ou bits rapidamente, através de operadores PixBlT ("Pixel Block Transfer") e BitBlT ("Bit Block Transfer"), respectivamente, que são procedimentos para copiar, mover ou combinar conjuntos retangulares de "pixels" ou de bits, usando algumas das familiares operações "booleanas", tornando muito práticas as aplicações com texto.

Através dessas operações, os processadores gráficos atuam sobre conceitos muito importantes, como por exemplo o de janela e recorte.

Alguns processadores gráficos podem servir como processador principal. Por exemplo, o 34010 permite a execução de até 6 milhões de instruções de propósito geral e de instruções especiais de processamento de "pixel" por segundo. O sistema que o acompanha possui um compilador C e ferramentas de desenvolvimento, habilitando-o a servir como o processador principal para um computador pequeno. As ferramentas de

desenvolvimento do sistema vêm na forma de um "assembler", um "linker", um simulador, uma biblioteca de funções, "software runtime", um cartão de desenvolvimento para PC, e um emulador de tempo real [AS86, TE86, TE86a].

Sendo assim, uma interface de visualização gráfica que manipula informações pode ser construída baseada em controladores ou processadores gráficos; a decisão de qual CI utilizar vai depender das características do sistema gráfico, de sua utilização, e também das condições financeiras encontradas.

CAPITULO 4 - IMPLEMENTAÇÃO DA INTERFACE DE VISUALIZAÇÃO GRAFICA

4.1) CONSIDERAÇÕES INICIAIS

O trabalho proposto para o mestrado tem por finalidade dotar um microcomputador com recursos gráficos.

Para tornar este microcomputador do tipo gráfico, desenvolveu-se uma Interface de Visualização Gráfica e a programação básica correspondente.

Esta interface é baseada em um Controlador Gráfico, do tipo EF9367 da Thomson Semiconductors, e faz o elo de ligação entre o computador e o sistema de visualização, que é um monitor de video.

A programação básica, que é descrita no capítulo seguinte, é composta de primitivas gráficas, de modo que outras funções de nível mais alto possam ser criadas pelo usuário do sistema.

4.2) CARACTERÍSTICAS DO MICROCOMPUTADOR

O sistema básico do microcomputador é modular, utilizando cartões de 44 pinos no barramento.

O módulo principal de controle é o cartão UCP com um microprocessador Z80A, a 4 MHz, com programa básico de controle em EPROM 2732 de 4 Kbytes.

A memória principal, que constitui um outro módulo, utiliza 8 CIs (circuitos integrados) de RAMs dinâmicas de 64 K X 1 bits, constituindo 64 K palavras de 8 bits.

Outro módulo é uma interface serial, RS232C, utilizada para

a conexão com um terminal de video alfanumérico.

E finalmente, este possui uma interface para o "drive" de discos flexíveis.

4.3) CARACTERISTICAS DA INTERFACE

A interface de visualização gráfica (IVG) implementada apresenta as seguintes características:

1. varredura para video com exploração rastreada;
2. geração de figuras com resolução alta (1024 X 416 pontos visíveis);
3. o projeto da interface é modular;
4. o projeto está dividido em duas placas: uma placa do controlador gráfico e a outra da memória gráfica;
5. os módulos foram inseridos no barramento do microcomputador.

As seguintes considerações devem ser feitas em relação a estas características.

Uma vez que o projeto da interface é modular, pode-se utilizar várias páginas de memória, com a finalidade de viabilizar a variação no nível de intensidade ou cores do gráfico.

A placa do controlador gráfico contém, além de tal controlador (o EF9367), os circuitos de seleção da placa e geradores de sinais para acesso à memória dedicada.

Os sistemas gráficos para videos com exploração rastreada possuem, geralmente, uma memória dedicada de leitura e escrita

que armazena as informações gráficas a serem mostradas no vídeo. A cada ponto luminoso do vídeo, denominado "pixel", deve ser gerada uma informação através do conteúdo desta memória.

Tal informação pode estar em forma final, ou seja, em correspondência direta com os pontos da tela, caso em que o armazenamento é denominado memória de quadro, ou então, pode estar em forma de instruções a serem processadas para a obtenção das informações finais de figuras no vídeo.

Optou-se pela utilização das memórias de quadro, uma vez que esta forma permite a modularidade em termos de níveis de intensidade ou cor.

Cada página de memória contém oito CIs de RAMs do tipo MCM6665 de 64 K X 1 bits. A memória de quadro é constituída de três páginas de memória, sendo que cada uma permite a obtenção de um nível de intensidade ou uma das três cores fundamentais (vermelho, verde ou azul). Combinando-se estes três níveis ou cores obtém-se um total de oito níveis de intensidade, numa situação monocromática, ou oito cores, para uma aplicação colorida.

Desta forma, pode-se utilizar um monitor em preto e branco ou a cores. Foi utilizado um a cores.

Os acessos para leitura e escrita (modificação do conteúdo) na memória de quadro são feitos pelo EF9367.

A figura 4.1 mostra o diagrama de blocos da arquitetura do sistema gráfico implementado.

Figura 4.1: Diagrama de blocos do sistema gráfico.

4.4) O CI EF9367

O EF9367, em [TH83 e TH83a], é um controlador para vídeo com exploração rastreada, entrelaçado ou não, compatível com padrão 525 ou 625 linhas, programável por um microprocessador.

Além das funções lógicas de tempo necessárias para gerar sinais de sincronismo e "blanking" do vídeo, o EF9367 possui dois processadores de visualização integrados a este: um gerador de vetor e um de caractere.

O EF9367 possui as seguintes características, que são

relevantes ao projeto desenvolvido:

- resolução utilizada: 1024 X 416 pontos;
- geração de vetores em velocidade muito alta (os 1024 pontos diagonais podem ser traçados em menos de 1,4 ms);
- gerador interno de caracteres ASCII (96 caracteres), com tamanhos e orientações programáveis;
- capacidade de aplicação multipágina;
- interfaceamento direto com o monitor através dos sinais SYNC (sincronismo composto) e BLK ("blanking");
- geração automática de ciclos de reavivamento, escrita, visualização e carregamento para a memória de quadro;
- entrada/saída compatível com lógica TTL;
- alimentação única em 5V.

O EF9367 tem, também, entrada para circuito de caneta ótica ("light pen"), fornecendo registradores e sinais de controle para esta (a caneta ótica não está sendo utilizada na interface).

Além da resolução citada, e utilizada no projeto (1024 X 416 pontos), pode-se programar o controlador gráfico para fornecer as seguintes resoluções:

- vertical: monitor de 525 linhas (208 ou 416 mostradas) e 625 linhas (256 ou 512 mostradas);
- horizontal: 256, 320*, 384*, 512, 640*, 768*, sendo que as assinaladas com * são conseguidas através de PROM externa.

O EF9367 apresenta uma configuração de 40 pinos, cujo diagrama de blocos e descrição funcional são mostrados através das figuras 4.2 e 4.3.

Figura 4.3 - Descrição funcional dos pinos do EF9367.

4.4.1) DESCRIÇÃO DOS PINOS

Apresenta-se, a seguir, uma descrição dos pinos do EF9367, agrupados conforme suas funções.

a) Alimentação, relógio e parâmetros de operação:

Vss - alimentação - terra;

Vcc - alimentação - +5V;

CK - relógio do CI - todos os estados internos do CI são modificados com a borda de descida deste sinal; também utilizado como sinal de multiplexação de endereços de memória, DAD;

FMAT - formato - este pino conectado a Vcc, Vss, CK ou $\overline{CK}$ fornece o número de linhas do monitor e da imagem:

Vcc - monitor de 625 linhas, entrelaçado, com 512 linhas mostradas;

CK - monitor de 525 linhas, entrelaçado, com 416 linhas mostradas;

$\overline{CK}$ - monitor de 525 linhas, não-entrelaçado, com 208 linhas mostradas;

Vss - monitor de 625 linhas, não-entrelaçado, com 256 linhas mostradas;

WO - somente escrita - quando WO é alto, não existem reavivamento da memória e visualização. Os processadores de escrita internos podem operar sem serem interrompidos.

b) Sinais de sincronização e "blanking":

SYNC - sinal de sincronização de linha e imagem do monitor de vídeo;

BLK - "blanking" - este sinal é alto quando em não-visualização (escrita ou reavivamento);

VB - "blanking" vertical - é alto durante retraço vertical.

c) Sinais de endereçamento da memória de quadro:

DAD0 a DAD6 - endereços de visualização - são multiplexados pelo sinal CK; usados para reavivamento automático das memórias dinâmicas de 16 K ou 64 K;

X9 - endereço de memória - usado para extensão horizontal em operações de escrita (para resolução horizontal maior que 512 pontos);

MSL0 a MSL3 - seleção de memória - utilizados para seleção e escrita de "pixels";

ALL - acesso a todos os CIs de memória - permite distinguir entre acesso coletivo (ALL = 0), para visualização, reavivamento e apagamento, e acesso a um único "pixel" (ALL = 1), para escrita de vetor ou caractere.

d) Sinais de controle da memória de quadro:

DIN - saída de dados para a memória - utilizado diretamente para monitor em preto e branco (um bit por "pixel");

DW - sinal de escrita na memória de quadro - ativo quando baixo;

MW - memória disponível - ativo em nível baixo (usado também como controle da caneta ótica).

e) Sinais do barramento do microprocessador:

D0 a D7 - barramento de dados - "buffer" de entrada e saída;
é controlado por E, e a direção por R/W;
A0 a A3 - barramento de endereço - endereço dos registradores
envolvidos no acesso do microprocessador;
R/W - sinal de leitura/escrita - escrita em nível baixo;
E - habilitação - sinal de habilitação e sincronismo do CI;
IRQ - pedidos de interrupções - interrupções enviadas ao
microprocessador; programável através do registrador
interno CTRL1.

f) Sinal de operação da caneta ótica:

LPCK - entrada da caneta ótica - quando o mecanismo está
ligado, uma borda de descida carrega os registradores
XLP e YLP com os endereços de visualização do momento
e ativa os bits menos significativos do registrador
XLP para alto.

4.4.2) OS REGISTRADORES INTERNOS DO EF9367

O EF9367 possui 11 registradores internos, acessíveis
através de 16 endereços consecutivos, por meio dos quais este se
torna programável. Esta programação é feita pelo
microprocessador; mas os registradores podem ser modificados
pelos processadores internos do controlador enquanto um comando
ainda estiver sendo executado.

Tais registradores são descritos a seguir:

- registradores X e Y:

Estes são registradores de 12 bits de leitura e escrita; são
utilizados para indicar a posição do próximo ponto a ser escrito

na memória de quadro.

Estes são incrementados ou decrementados antes de cada operação de escrita na memória de quadro pelos geradores de vetor e de caractere internos, ou podem ser diretamente modificados pelo microprocessador.

Somente uma parte dos bits dos registradores X e Y são usados (os menos significativos) para proporcionar uma resolução máxima de 1024 X 512 pontos (a menos que o bit 3 do registrador CTRL1 esteja em nível alto, como será descrito a seguir);

- registradores DELTAX e DELTAY:

São registradores de 8 bits de leitura e escrita, e que indicam ao gerador de vetores as projeções do próximo vetor a ser traçado, sobre os eixos X e Y, respectivamente. O traçado do vetor é comandado pelo registrador de comando CMD;

- registrador CSIZE:

Registrador de 8 bits de leitura e escrita, e que indica o fator de multiplicação dos registradores X e Y para os símbolos e caracteres. São gerados 98 caracteres a partir de uma matriz de 5 X 8 pontos definidos em uma ROM interna.

Este registrador é definido da seguinte forma:

$$\begin{array}{cc} \underline{bbbb} & \underline{bbbb} \\ P & Q \end{array}$$

onde P e Q são os fatores de multiplicação de modo a fazer com que o caractere a ser escrito seja alterado para P(X) e Q(Y). Estes fatores variam de 1 a 16. A escrita do caractere é comandada pelo registrador CMD;

- registrador CTRL1:

Registrador de 7 bits de leitura e escrita, através do qual a operação geral do circuito pode ser realizada com os parâmetros desejados.

Cada bit tem seu próprio significado:

- bit 0: quando alto habilita a escrita na memória de quadro; quando baixo, inibe. Este bit controla a saída $\overline{DW}$;
- bit 1: quando alto permite o traçado na tela; quando baixo, inibe. Controla a saída DIN;
- bit 2: quando baixo seleciona modo de escrita normal na memória de quadro (com períodos de visualização e reavivamento de armazenamento dinâmico); quando alto seleciona modo de escrita em alta velocidade (elimina os períodos de visualização);
- bit 3: quando baixo indica que o espaço de 4096 X 4096 pontos está sendo usado (todos os 12 bits de ambos os registradores X e Y são significativos); quando alto, o espaço é igual ao desejado para a tela;
- bit 4: quando alto este habilita a interrupção disparada pelo circuito da caneta ótica; quando baixo, inibe;
- bit 5: quando alto habilita a interrupção provocada pelo "blanking" vertical; quando baixo, inibe;
- bit 6: quando alto habilita a interrupção indicando que o sistema está pronto para novo comando; quando baixo, inibe;
- bit 7: não é usado; sempre baixo em modo de leitura.

- registrador CTRL2:

Registrador de 8 bits de leitura e escrita, através do qual o traçado de vetores pode ser denotado por parâmetros.

O significado de cada um dos bits é dado a seguir:

bits 0 e 1: estes dois bits definem 4 tipos de linhas:

- 00 - contínua;
- 01 - pontilhada;
- 10 - tracejada;
- 11 - traço-ponto;

bit 2: quando baixo define escrita normal; quando alto define escrita de caractere inclinado;

bit 3: quando baixo define escrita ao longo de uma linha horizontal; quando alto, vertical;

bits 4, 5, 6, 7: não são usados; sempre baixo em modo de leitura;

- registrador de comando CMD:

Registrador de 8 bits somente para escrita. Cada operação de escrita neste registrador provoca a execução de um comando.

Vários tipos de comandos são disponíveis:

- traçado de vetor e caractere;
- apagamento ("erase") da tela;
- habilitação para o circuito da caneta ótica;
- acesso à memória de quadro através de um circuito externo;
- modificação indireta de outros registradores;

- registrador de estado ("status"):

Registrador de 8 bits somente para leitura. É usado para monitorar os estados das declarações em execução inseridas no circuito, e, mais especificamente, para evitar a modificação de

um registrador que está sendo usado por algum comando em execução.

Seus bits têm os seguintes significados:

- bit 0: quando baixo indica que uma sequência da caneta ótica está em execução;
- bit 1: este é alto durante "blanking" vertical; é uma cópia do sinal VB;
- bit 2: quando baixo indica que um comando está em execução; quando alto indica que o circuito está pronto para um novo comando;
- bit 3: quando baixo indica que os registradores X e Y estão apontando dentro do espaço de memória assumido;
- bit 4: quando alto indica que uma interrupção foi iniciada pela conclusão de uma sequência da caneta ótica;
- bit 5: quando alto indica que uma interrupção foi iniciada pelo "blanking" vertical;
- bit 6: quando alto indica que uma interrupção foi iniciada pela conclusão de execução de um comando;
- bit 7: quando alto indica que uma interrupção foi iniciada; este é uma lógica OU entre os bits 4, 5 e 6 do registrador de estados. O estado da saída $\overline{\text{IRQ}}$ é sempre o oposto do estado deste bit;

- registradores XLP e YLP:

Registradores de 7 e 8 bits, respectivamente, somente para leitura. Usados durante uma sequência da caneta ótica para indicar os endereços de visualização apontados por esta.

4.5) O PROJETO DA INTERFACE

Neste item, descreve-se as características do projeto da interface de visualização gráfica desenvolvida.

4.5.1) O SINAL FMAT

Existe a necessidade de uma definição em relação ao número de linhas do monitor e da imagem; para tanto, como descrito anteriormente, deve-se conectar o pino FMAT (formato), conforme se faça necessário, a um dos seguintes pinos: Vcc (+5V), Vss (0V), CK (relógio) ou $\overline{CK}$.

Uma vez que a resolução utilizada é de 1024 X 416 pontos, ou seja, 416 linhas mostradas, em um monitor de 525 linhas em 60 Hz, o pino FMAT do EF9367 foi conectado ao CK, também do controlador.

4.5.2) O ENDEREÇO DE ESCRITA

O endereço de escrita na memória de quadro é constituído dos bits menos significativos dos registradores X e Y, que são de 12 bits:

- X : x0, x1, x2, ..., x9
- Y : y0, y1, y2, ..., y8.

Esses endereços são representados pelos sinais DAD, MSL e X9 do EF9367.

Quando se trabalha com resolução horizontal até 512 pontos, utiliza-se para o endereço horizontal os bits de x0 a x8 ($2^9 = 512$) do registrador X.

Quando se trabalha, numa operação de escrita, com resolução horizontal maior que 512 pontos, há a necessidade de se utilizar o bit x9 do registrador X; para se conseguir isso, tem-se a saída

(pino) X9 do EF9367, o qual é uma representação física do bit x9.

Então, para o projeto em questão utilizou-se o pino X9, uma vez que a resolução horizontal é superior a 512 pontos (1024).

4.5.3) SELEÇÃO DO EF9367

O EF9367 é habilitado, e portanto selecionado, através de sua entrada $\bar{E}$.

O circuito lógico utilizado para tal seleção é mostrado na figura 4.4.

Figura 4.4: Circuito para seleção do EF9367.

A seleção acontecerá quando $\bar{E}$ estiver em nível baixo. Isso ocorrerá quando o sinal de decodificação estiver em nível baixo e $\overline{IORQ}$ (fornecido pelo Z80A) também. Na primeira borda de subida do sinal ϕ (relógio do Z80A) após $\overline{IORQ}$ se tornar baixo, o sinal $\bar{E}$ tornar-se-á baixo, e tornar-se-á alto ao mesmo tempo que $\overline{IORQ}$.

O sinal de decodificação é conseguido a partir do decodificador LS266.

A figura 4.5 mostra o esquema de decodificação.

Figura 4.5: Esquema utilizado para o sinal de decodificação.

Quando o Z80A emite um endereço de 8 bits do tipo Bx, em hexadecimal, todas as saídas do CI LS266 estarão em nível alto e, então, após passar pelo LS00, e assim ser invertido, o sinal de decodificação estará em nível baixo. No endereço emitido pelo Z80A, B engloba os quatro bits mais significativos (A7, A6, A5 e A4), que serão decodificados, e x representa o endereço de um dos onze registradores internos do EF9367, ou a seleção de cor (que será descrita adiante), correspondendo aos quatro bits menos significativos.

O sinal $\bar{E}$ controla também o "buffer" de entrada e saída de dados, para as linhas D0 a D7.

4.5.4) SINCRONISMO DO EF9367 COM O Z80A

O microprocessador Z80A apresenta a seguinte especificação de tempo, quanto ao sinal $\overline{IORQ}$:

O Z80A tem um período de tempo t_{CLK} .

Como descrito anteriormente, o sinal $\overline{\text{E}}$ para o EF9367 estará em nível baixo na primeira borda de subida do sinal ϕ (relógio do Z80A) após $\overline{\text{IORQ}}$ se tornar baixo, e tornar-se-á alto ao mesmo tempo em que $\overline{\text{IORQ}}$ se tornar alto, o que acontecerá na descida do período $T3$ do sinal ϕ .

Sendo assim, o EF9367 tem a seguinte especificação de tempo para o sinal $\overline{\text{E}}$, em relação aos sinais ϕ e $\overline{\text{IORQ}}$ do Z80A:

sendo t_{EL} a largura do pulso $\overline{\text{E}}$ quando em nível baixo.

O EF9367 tem uma especificação mínima para a largura do pulso $\overline{\text{E}}$ igual a 450 ns.

Pelo diagrama de tempo e pelo circuito lógico para a entrada $\overline{\text{E}}$, pode-se tirar a seguinte expressão:

$$t_{EL} = 3/2 t_{CLK} - t_{PHL} (LS74) - t_{PHL} (LS32)$$

$$t_{EL} = 3/2 t_{CLK} - 62 \text{ ns.}$$

sendo $t_{EL} = 450 \text{ ns}$, no mínimo, obtém-se $t_{CLK} = 340 \text{ ns}$ no mínimo.

Isso significa que a frequência do relógio (sinal ϕ) do Z80A não pode exceder a 2,9 MHz.

Para se utilizar frequências do sinal ϕ maiores que este limite, é necessário gerar períodos TW adicionais, de modo a desacelerar o ciclo do Z80A.

Isso pode ser conseguido fazendo com que o sinal $\overline{WAIT}$ do Z80A permaneça em nível baixo durante uma quantidade de tempo suficiente para retardar o surgimento do período T3, retardando a subida do sinal $\bar{E}$.

O esquema mostrado na figura 4.6 foi utilizado para gerar o sinal $\overline{WAIT}$, através do monoestável LS123, para o Z80A, a fim de fornecer um período TW adicional.

Figura 4.6: Esquema usado para gerar o sinal $\overline{WAIT}$.

Com a inclusão de mais um período TW, a nova especificação de tempo para o EF9367, em relação aos sinais ϕ e $\overline{IORQ}$ do Z80A é:

onde $t_s(wt) = 70 \text{ ns}$, valor mínimo, é conseguido através de $R_{ext} = 4,7 \text{ K}\Omega$ e de $C_{ext} = 150 \text{ pF}$, ligados ao LS123.

Através do diagrama de tempo e do circuito lógico para a entrada $\bar{E}$, pode-se tirar uma nova expressão:

$$t_{EL} = 5/2 t_{CLK} - 62 \text{ ns}$$

sendo $t_{EL} = 450 \text{ ns}$, no mínimo, obtém-se $t_{CLK} = 200 \text{ ns}$ no mínimo.

Isso significa que a frequência máxima do sinal ϕ do Z80A é de 5 MHz.

Assim, a frequência de 4 MHz do Z80A utilizado é totalmente aceitável, uma vez que o valor de t_{CLK} é igual a 250 ns, proporcionando um tempo t_{EL} igual a 560 ns, maior, portanto, que o mínimo necessário de 450 ns.

Também o sinal $R/\bar{W}$ do controlador gráfico deve ser gerado em sincronismo com os sinais ϕ e $\bar{WR}$ do Z80A.

Para o EF9367, $R/\bar{W}$ estará em nível baixo quando $\bar{WR}$ também estiver, e passará a alto na primeira borda de subida do sinal ϕ após $\bar{WR}$ alcançar o nível alto.

Isso pode ser conseguido através do esquema mostrado na figura 4.7:

Figura 4.7: Esquema usado para a geração do sinal R/W.

Esse esquema produz a seguinte especificação de tempo para R/W do EF9367, em relação aos sinais ϕ e $\overline{WR}$ do Z80A:

O sinal R/W também controla a direção do "buffer" de entrada e saída de dados, para as linhas D0 a D7.

4.5.5) GERAÇÃO DOS SINAIS DE TEMPORIZAÇÃO

O relógio do EF9367 tem uma frequência de 1,5 MHz. Tanto este sinal como alguns outros utilizados durante um ciclo de memória são gerados por uma PROM do tipo 74S288.

Os sinais gerados pela PROM são os seguintes:

- CK : relógio para o EF9367;
- $\overline{\text{LOAD}}$: carregamento dos registradores de deslocamento associados às memórias (duplo por ciclo do EF9367);
- $\overline{\text{RAS}}$: sinal de multiplexação do endereço de linha;
- $\overline{\text{CASD}}$: sinal de multiplexação do endereço de coluna durante visualização;
- $\overline{\text{CASW}}$: sinal de multiplexação do endereço de coluna durante escrita;
- STR : carregamento dos sinais DAD;
- $\overline{\text{L/M}}$: sinal de controle para a seleção de um byte dentro de um segmento;
- MUX : sinal de controle de multiplexação de endereços.

A PROM é endereçada pelo contador de 4 bits LS163A que é disparado ("clocked") pelo relógio mestre do sistema, denominado HCK, de 23,916 MHz obtido a partir de um cristal.

O sinal CK

Além de servir como relógio para o EF9367, o sinal CK é utilizado para definir o número de linhas do monitor e da imagem, pois está ligado a FMAT, e é usado como sinal de habilitação do CI LS373, que mantém os sinais de seleção das RAMs da memória em um nível estável.

O sinal STR

O sinal STR controla um "latch", cuja função é manter os sinais DAD, gerados pelo EF9367, em um estado estável, enquanto são multiplexados pelos sinais $\overline{\text{RAS}}$ e $\overline{\text{CAS}}$.

Os sinais CASD e CASW

A seguir, descreve-se as aplicações dos sinais $\overline{\text{CASD}}$ e $\overline{\text{CASW}}$.

Assumir a existência de uma resolução de $V \times H$ "pixels" (V linhas por H pontos), numa aplicação monocromática.

Pode-se dividir a linha de H pontos em h segmentos adjacentes de n pontos cada um (n bits numa aplicação monocromática), pontos estes lidos ao mesmo tempo (em paralelo) na memória de quadro, e em seguida convertidos para a forma serial, por meio de um registrador de deslocamento, para produzir o sinal de vídeo.

Sendo assim, são necessários h acessos de memória por linha, e cada acesso carrega o registrador de deslocamento, que tem entrada paralela e saída serial, com n bits.

O EF9367 é projetado para os seguintes formatos de imagem armazenada:

$V = 512$ ou 256 (50 Hz) ou 416 ou 208 (60 Hz)

$H = 1024$ ou múltiplos de 64, menores que 1024

$h = 64$ (sempre)

$n = 16, 8, 4, 2$ ou 1 .

O projeto desenvolvido apresenta os seguintes formatos:

$V = 416$

$H = 1024$

$h = 64$

e, então, $n = 16$ ($1024/64$).

Isso significa que tem-se por linha 64 segmentos de 16 bits cada, e desta forma é necessário um registrador de deslocamento

de 16 bits; portanto, deve-se utilizar 16 CIs de memória RAM.

O maior inconveniente associado com este tipo de aplicação é o número de CIs de memória utilizado. Consequentemente, preço e tamanho são as principais considerações as quais levam a explorar outras soluções alternativas mais compactas e adequadas.

Pode-se, entretanto, baseado nos parâmetros anteriores, utilizar apenas 8 CIs de memória ao invés dos 16 anteriormente exigidos.

Para tanto, deve-se executar, em modo de visualização, dois ciclos de leitura da memória durante um ciclo do EF9367. O segmento de 16 pontos é lido em dois passos de oito pontos cada.

Então, 8 CIs de memória de 64 K X 1 bits são suficientes.

Este tipo de aplicação possibilita a operação em modo página. Este modo permite a execução de vários ciclos de leitura tendo idênticos endereços de linha ($\overline{\text{RAS}}$), como mostra o diagrama seguinte:

Na operação de leitura, com o sinal $\overline{\text{CAS}}$ em modo página, duas porções de um segmento de 16 pontos são lidos em dois passos sucessivos de 8, e, então, aplicados sucessivamente ao mesmo registrador de deslocamento de 8 bits para conversão paralelo

para serial. Esta operação é executada primeiro sobre os 8 pontos posicionados a esquerda ($x_3x_2x_1x_0 = 0000$ a $x_3x_2x_1x_0 = 0111$, portanto $x_3 = 0$), e então sobre os 8 pontos da direita ($x_3x_2x_1x_0 = 1000$ a $x_3x_2x_1x_0 = 1111$, portanto $x_3 = 1$). A maneira utilizada para se conseguir o valor do bit x_3 do registrador de endereço X é descrita adiante.

O sinal $\overline{\text{CAS}}$ em modo página é representado, no projeto da interface, pelo sinal $\overline{\text{CASD}}$, gerado pela PROM.

Em modo de escrita, um bit é escrito por vez, e o ciclo de memória usado é o normal.

O sinal $\overline{\text{CAS}}$ em modo normal é representado pelo sinal $\overline{\text{CASW}}$, também gerado pela PROM.

Obtenção de x_3

O valor do bit x_3 do registrador X para visualização e reavivamento, e para escrita é obtido da seguinte forma.

Os contadores internos, os quais endereçam a memória de quadro, são constituídos de:

- 6 bits de endereço horizontal: h_0 a h_5 , para o endereçamento de 64 segmentos de 16 pontos;
- 9 bits de endereço vertical: t , v_0 , v_1 , ..., v_7 , o que proporciona acesso a 512 pontos verticais.

O EF9367 produz os sinais de endereços MSL, DAD e X9 conforme a tabela 1.

Tabela 1 - Endereços produzidos pelo EF9367.

ALL CK		MSL				X9	DAD							
		0	1	2	3		0	1	2	3	4	5	6	
leitura	0	0	x0	x1	x2	v1	x9	h5	h4	h3	h2	h1	h0	v0
	0	1						v7	v6	v5	v4	v3	v2	t
escrita	1	0	x0	x1	x2	y2	x9	x8	x7	x6	x5	x4	x3	y1
	1	1						y8	y7	y6	y5	y4	y3	y0

Durante a primeira fase de seu ciclo (CK = 0), o EF9367 emite os seguintes sinais nas saídas DAD, conforme tabela 2.

Tabela 2 - Sinais DAD do EF9367 quando CK = 0.

visualização/reavivamento

DAD5	h0
DAD4	h1
DAD3	h2
DAD2	h3
DAD1	h4
DAD0	h5

escrita

x3	DAD5
x4	DAD4
x5	DAD3
x6	DAD2
x7	DAD1
x8	DAD0
x9	X9

Quando em modo de visualização/reavivamento e de escrita, os seguintes sinais são fornecidos às linhas de endereçamento (A0 a A7) dos CIs de memória, conforme tabelas 3 e 4.

TABELA 3 - EM MODO DE ESCRITA (BLK = 1)

$\overline{\text{RAS}}$	$\overline{\text{CAS}}$	Endereço
X9 (x9)	DAD0 (y8)	A7
DAD0 (x8)	DAD1 (y7)	A6
DAD1 (x7)	DAD2 (y6)	A5
DAD2 (x6)	DAD3 (y5)	A4
DAD3 (x5)	DAD4 (y4)	A3
DAD6 (y1)	DAD6 (y0)	A2
MSL3 (y2)	DAD5 (y3)	A1
DAD4 (x4)	DAD5 pulsado (x3)	A0

TABELA 4 - MODO DE VISUALIZAÇÃO (BLK = 0)

$\overline{\text{RAS}}$	$\overline{\text{CAS1}}$	$\overline{\text{CAS2}}$	Endereço
DAD0 (h5)	DAD0 (v7)	DAD0 (v7)	A7
DAD1 (h4)	DAD1 (v6)	DAD1 (v6)	A6
DAD2 (h3)	DAD2 (v5)	DAD2 (v5)	A5
DAD3 (h2)	DAD3 (v4)	DAD3 (v4)	A4
DAD4 (h1)	DAD4 (v3)	DAD4 (v3)	A3
DAD6 (v0)	DAD6 (t)	DAD6 (t)	A2
MSL3 (v1)	DAD5 (v2)	DAD5 (v2)	A1
DAD5 (h0)	0	1	A0

Em modo de escrita, o bit x3 é fornecido juntamente com o endereço de linha, pelo sinal DAD5, durante a primeira fase do ciclo (CK = 0). Então, este valor precisa ser armazenado em um "flip-flop" (um LS74), conforme figura 4.8, e, subsequentemente,

aplicado junto com o endereço de coluna para os CIs de memória, durante a segunda fase do ciclo ($CK = 1$) (este sinal é chamado DAD5 pulsado).

Figura 4.8: Lógica para a obtenção de DAD5 pulsado (x3).

Em modo de leitura, pode-se obter o valor equivalente do bit x3 através do sinal $\bar{L}/M$ gerado pela PROM, que é igual a ZERO durante o primeiro $\overline{CAS}$, e igual a UM durante o segundo.

Através deste sinal consegue-se selecionar, em sucessão, a porção da esquerda ($\bar{L}/M = 0$) e a da direita ($\bar{L}/M = 1$) de um segmento de 16 pontos. Abaixo, mostra-se a forma de onda de $\bar{L}/M$.

A figura 4.9 mostra o arranjo de multiplexação usado para gerar os sinais de endereço da memória:

Figura 4.9: Multiplexação dos sinais de endereço gerados pelo EF9367.

4.5.6) SELEÇÃO DOS CIs DE MEMÓRIA

Como descrito anteriormente, para uma resolução de 1024 X 416 pontos, como a do projeto, em uma aplicação monocromática, 8 CIs de memória RAM de 64 K X 1 bits são suficientes.

Para endereçar, e portanto selecionar, estes 8 CIs utilizou-se os sinais MSL0, MSL1, MSL2, $\overline{ALL}$ e $\overline{DW}$ gerados pelo EF9367, mais a lógica composta pelos integrados LS138, LS08 e LS32.

O LS138 é utilizado para decodificar os sinais MSL0, MSL1 e MSL2 em oito sinais (decodificador 3 para 8). O sinal resultante

de uma lógica OU (LS32), entre os sinais ALL e DW, é combinado com cada um dos oito sinais decodificados anteriormente, através de uma lógica E (LS08), e assim enviados para as entradas $\bar{W}$ dos CIs de memória, para selecioná-los, individual ou coletivamente, dependendo dos níveis de $\overline{ALL}$ e $\overline{DW}$.

Se $\overline{ALL}$ for zero, o acesso será coletivo, caso contrário, individual. Se $\overline{DW}$ for zero, a operação a ser executada será a de escrita.

4.5.7) SELEÇÃO DAS CORES

Tem-se descrito, até aqui, um projeto aplicado a uma situação monocromática, em que foram utilizados 8 CIs de memória RAM de 64 K X 1 bits para satisfazer uma resolução de 1024 X 416 pontos.

Para uma aplicação colorida, somente é necessário montar, ao invés de um, três conjuntos de CIs, a fim de se conseguir três páginas de memória; cada uma dessas páginas irá controlar uma das três cores fundamentais (vermelho, verde e azul).

O projeto desenvolvido tem uma aplicação colorida. Dessa forma, foram construídas três páginas de memória, e assim, por combinação das três cores fundamentais, se consegue aplicações com até oito cores.

Cada página de memória utiliza uma linha de dados do Z80A; o dado D0 para o vermelho, o D1 para o verde e o D2 para o azul.

A seleção das cores é feita por programação, e controlada para escrita pelo endereço B4 em hexadecimal (o endereço 04 não é utilizado pelos registradores internos do EF9367), decodificado pelo LS42, em conjunto com o sinal $\bar{E}$. O sinal resultante de uma

lógica OU entre estes dois sinais é utilizado como habilitação para o LS373, que tem como entrada as três linhas de dados D0, D1 e D2.

Essas três linhas de dados são aplicadas, a seguir, ao CI LS244, que é controlado pelo sinal $\overline{DW}$, de escrita na memória de quadro, para serem posteriormente utilizadas como as entradas de dados dos CIs de cada página de memória.

A lógica utilizada para a seleção das cores é mostrada na figura 4.10.

Figura 4.10: Lógica usada para a seleção das cores.

4.5.8) O REGISTRADOR DE DESLOCAMENTO

A saída de cada um dos 8 CIs de memória está ligada a um registrador de deslocamento do tipo LS166, que recebe os oito dados em paralelo e os serializa para apresentação ao monitor.

Esses registradores são disparados pelo relógio mestre HCK em 23,916 MHz, que é a frequência de apresentação dos pontos ao

monitor, e têm as entradas $S/\bar{L}$, de deslocamento/carregamento dos dados, controladas por uma lógica OU entre os sinais $\overline{LOAD}$, gerado pela PROM, e BLK, gerado pelo EF9367.

A figura 4.11 mostra o esquema utilizado no projeto da interface para os registradores de deslocamento.

Figura 4.11: Lógica usada para o registrador de deslocamento.

4.5.9) SINAL DE SINCRONISMO E A SAÍDA RGB

O sinal de sincronismo SYNC, que é gerado pelo EF9367, é aplicado diretamente ao monitor, tendo apenas um potenciômetro (de 10 K Ω) para ajuste do nível de sincronismo.

Também as linhas de cor, que são aplicadas ao controle RGB do monitor, possuem potenciômetros que são utilizados para controlar os níveis de saturação e de intensidade das cores.

4.5.10) MEMÓRIA DE QUADRO

A memória de quadro contém informações gráficas, cuja correspondência com os sinais gerados no vídeo é biunívoca, isto é, a cada bit de memória associa-se um ponto luminoso da tela.

Basicamente, existem três tipos de acessos a esta memória.

Um desses é o acesso para leitura com a finalidade de visualização. Como o tempo nominal desse acesso para a memória dinâmica utilizada é de 150 ns, e o tempo de apresentação de um ponto ("pixel") ao vídeo é da ordem de 45 ns, é necessário utilizar-se um registrador de deslocamento com carregamento paralelo para esse fim.

Sendo assim, utilizou-se um registrador de deslocamento de 8 bits, obtendo-se um tempo de acesso para leitura na memória da ordem de 360 ns (8×45 ns), que é suficiente para a memória utilizada.

Nesse esquema, para cada acesso à memória são lidas informações referentes a oito pontos luminosos do vídeo, conforme mostra-se na figura 4.12.

Figura 4.12: Esquema utilizado para a memória de quadro.

Um outro tipo de acesso à memória de quadro é para a escrita de informação gráfica gerada pelo EF9367.

A escrita tem menor prioridade em relação à leitura, devido a

este último acesso necessitar sincronizar-se com o tempo de varredura do monitor de vídeo. Portanto, a escrita é feita, normalmente, durante o tempo de retraço horizontal ou vertical de varredura.

O terceiro tipo de acesso corresponde ao de reavivamento do conteúdo da memória, devido ao fato de se estar utilizando CIs de memória do tipo dinâmico. Esse tipo de acesso coincide com o de leitura, pois a geração de visualização garante a leitura total da memória, de forma síncrona e num espaço de tempo suficiente para o reavivamento.

4.5.11) INTERFACE COM O MONITOR DE VIDEO

A interface do monitor de vídeo utilizado com os módulos gráficos é realizada através de um conector DELTA de nove pinos, cuja configuração dos pinos foi definida de forma a compatibilizar-se com os demais microcomputadores. Dessa forma, os sinais utilizados são os seguintes:

pino	significado
1	terra
2	terra
3	sinal vermelho
4	sinal verde
5	sinal azul
6	-----
7	-----
8	sincronismo
9	-----

CAPITULO 5 - SISTEMA PROGRAMACIONAL BASICO

5.1) CONSIDERAÇÕES INICIAIS

E sempre desejável que uma interface de visualização gráfica trabalhe corretamente e em alta velocidade. Para colaborar nesse sentido, faz-se necessário que tal interface seja manipulada por um sistema programacional ("software" gráfico) dedicado, ou seja, um sistema criado conforme o tipo de interface.

O sistema programacional deve conter um conjunto adequado de funções gráficas, de modo a fazer com que a interface tenha um desempenho eficiente na criação e apresentação de figuras.

As funções que são utilizadas para manipular uma interface de visualização gráfica podem ser divididas, basicamente, em dois grupos: as funções primitivas gráficas e as funções de alto nível.

As funções primitivas gráficas são as funções que devem ser desenvolvidas baseadas nos circuitos da interface existente, ou seja, são dependentes da maneira como a interface foi desenvolvida e dos elementos que a compõem.

Entre as funções primitivas gráficas encontram-se:

- limpar a tela;
- traçar ponto;
- traçar segmento de reta;
- traçar caractere;
- posicionar cursor.

As funções de alto nível são assim denominadas por não dependerem diretamente dos circuitos da interface.

Por outro lado, estas são dependentes das primitivas, pois podem ser construídas a partir destas, ou de outras funções ou parâmetros, que dependem das primitivas ou dos circuitos componentes da interface.

Entre as funções de alto nível encontram-se:

- traçado de círculo, polígono, etc.;
- transformação de rotação, translação, etc.;
- aquelas usadas em técnicas de janela e recorte.

5.2) AS PRIMITIVAS GRAFICAS

Além da interface de visualização criada, foi, também, implementado um conjunto básico de funções primitivas gráficas, para que, através dessas, outras funções de nível mais alto possam ser desenvolvidas.

Uma vez que a interface desenvolvida está baseada no controlador gráfico EF9367, o conjunto básico de funções foi implementado segundo os comandos e parâmetros aceitos por tal controlador.

Apresenta-se, a seguir, uma descrição das primitivas gráficas implementadas. Tal descrição é feita em forma de "pseudo-código", para facilitar o entendimento das características de cada primitiva.

Estas primitivas são:

- COLOR (cor)

escreve cor no registrador de cor
retorno.

Esta função é utilizada para a definição da cor desejada, tanto na execução de uma operação de escrita como na de limpeza da tela. Esta escreve o parâmetro da cor desejada (cor) no registrador de cor (sendo utilizado no projeto como registrador de cor o CI LS373), através do endereço 0B4 H.

Este parâmetro varia, inicialmente, de 0 a 7, devido às características da interface.

- LIMPA

escreve código 04 H no registrador CMD
retorno.

E usada para executar a limpeza da tela; escreve o comando para limpar a tela (04 H) no registrador interno CMD, endereçado por 0B0 H, o qual limpa a memória de quadro imediatamente.

- CURSOR (x,y)

escreve x no registrador X
escreve y no registrador Y
retorno.

E usada para colocar o cursor interno do EF9367 numa posição desejada da tela (x,y), ou seja, determinar a posição inicial de escrita na tela. Esta função escreve o valor do parâmetro x no

registrador X, endereçado por 0B8 H e 0B9 H, e o valor do parâmetro y no registrador interno Y, endereçado por 0BA H e 0BB H.

- DC (cod)

escreve cod no registrador CMD
retorno.

Esta função executa o traçado na tela de um dos caracteres definidos na ROM interna do EF9367. Esta escreve o código do caractere desejado (cod) no registrador CMD do controlador, endereçado por 0B0 H, o que faz com que tal caractere seja escrito imediatamente na memória de quadro.

- DP

escreve código 80 H no registrador CMD
retorno.

Função utilizada para executar o traçado de um ponto na tela; escreve o código 80 H no registrador CMD do EF9367, endereçado por 0B0 H, para que seja escrito um ponto na memória de quadro.

Quando um código desse tipo (80 H) é escrito no registrador CMD, o EF9367 reconhece tal código como um comando para a geração de pequenos vetores.

Uma informação, utilizada pelo EF9367 para a geração de pequenos vetores, tem o seguinte formato:

b7 b6b5 b4b3 b2b1b0

1 x y direção

onde:

- bits 0, 1, 2 : dão a direção de traçado do vetor;
- bits 3, 4 : dão o valor da projeção no eixo y;
- bits 5, 6 : dão o valor da projeção no eixo x;
- bit 7 : sempre igual a um para pequenos vetores.

Para as projeções nos eixos x e y, tem-se:

- 00 - 0 passo;
- 01 - 1 passo;
- 10 - 2 passos;
- 11 - 3 passos.

Quanto a direção, tem-se o seguinte:

Para o código utilizado, 80 H, tem-se:

80 H = 1 00 00 000

que significa:

- geração de pequenos vetores;
- projeção em x = 00, ou seja, 0 passo;
- projeção em y = 00, ou seja, 0 passo;
- direção = 000, ou seja, a positiva do eixo x.

Desta forma, verifica-se a formação de um ponto.

Quando o comando é definido neste formato, e portanto, para fins de traçado de pequenos vetores, os registradores internos DELTAX e DELTAY são ignorados.

- ESTILO (est)

escreve est no registrador CTRL2

retorno.

Esta função é usada para a definição do estilo de traçado de segmentos de reta; escreve o valor est no registrador CTRL2, endereçado por 0B2 H, o qual define o tipo de traçado a ser utilizado.

O parâmetro est pode assumir um dos seguintes valores:

00 - para traçado contínuo;

01 - para pontilhado;

02 - para tracejado;

03 - para traço-ponto.

- ORIENTAÇÃO (dir)

escreve dir no registrador CTRL2

retorno.

Usada para definir se o caractere será escrito reto ou inclinado e, se no sentido horizontal ou vertical; escreve o valor dir no registrador CTRL2 do EF9367, endereçado por 0B2 H, o qual define a maneira como um caractere será traçado.

O parâmetro dir pode assumir um dos seguintes valores:

- 00 - escrita ao longo de uma linha horizontal, com caracteres retos;
- 04 - escrita ao longo de uma linha horizontal, com caracteres inclinados;
- 08 - escrita ao longo de uma linha vertical, com caracteres retos;
- 0C - escrita ao longo de uma linha vertical, com caracteres inclinados.

- TAMANHO (tam)

escreve tam no registrador CSIZE
retorno.

Função usada para a definição do tamanho do caractere a ser traçado; escreve o valor tam no registrador CSIZE do EF9367, endereçado por 0B3 H. O registrador CSIZE é um registrador de 8 bits que indica os fatores de multiplicação para a direção x (os 4 bits mais significativos) e para a direção y (os 4 bits menos significativos) de uma matriz de 5 x 8 "pixels" para a geração dos caracteres, definidos em ROM interna ao EF9367.

Cada caractere pode ser aumentado por um fator $P(x)$ e $Q(y)$, sendo que P e Q podem variar de 1 a 16.

- DIMENSOES (dx,dy)

escreve dx no registrador DELTAX
escreve dy no registrador DELTAY
retorno.

Esta função é usada para a definição das projeções, do segmento de linha a ser traçado, nas direções x e y; escreve o valor dx no registrador DELTAX, endereçado por 0B5 H, e o valor dy no registrador DELTAY, endereçado por 0B7 H, ambos internos ao EF9367, para a definição do comprimento do segmento de linha.

- TR (cd)

escreve cd no registrador CMD

retorno.

Função usada para executar o traçado de um segmento de linha na tela; escreve o código cd H no registrador CMD do EF9367, endereçado por 0B0 H, para que seja traçado um segmento de linha na memória de quadro. O código cd, que fornece o sentido de traçado do segmento de linha, assume valores idênticos aos da direção de traçado de pequenos vetores (00 a 07), como descrito para a função DP.

- LINHA (x,y,x1,y1)

calcula o valor de cd

calcula os valores de dx e dy

chama CURSOR (x,y)

chama DIMENSOES (dx,dy)

chama TR (cd)

retorno.

Ativa os parâmetros para o traçado de um segmento de linha.

Esta função encontra o valor do parâmetro cd, de acordo com as coordenadas dos pontos inicial (x,y) e final (x1,y1), para

indicar o sentido de traçado do segmento de linha. Calcula, a seguir, os valores dos parâmetros dx, que é o valor absoluto de (x-x1), e dy, que é o valor absoluto de (y-y1), para determinar as projeções do segmento de linha nas direções x (dx) e y (dy).

Em seguida, chama as funções CURSOR, para o posicionamento inicial de traçado (x,y), e DIMENSOES, para ativar os parâmetros dx e dy.

Finalmente, chama a função TR para o traçado do segmento de linha, conforme o parâmetro cd, encontrado anteriormente.

- LIMPATELA (cor)

chama COLOR (cor)

chama LIMPA

retorno.

Função que ativa os parâmetros para a limpeza da tela; esta chama as funções COLOR, para ativar o parâmetro cor da cor desejada para o fundo de tela, e LIMPA, para executar a limpeza da tela.

- PONTO (x,y)

chama CURSOR (x,y)

chama DP

retorno.

Ativa os parâmetros para o traçado de um ponto na tela; esta chama as funções CURSOR, para o posicionamento no local de escrita (x,y), e DP, para executar a escrita de tal ponto na memória de quadro.

- texto (x,y,txt)
 chama CURSOR (x,y)
 verifica os códigos ASCII do texto txt (cod)
 chama DC (cod)
 retorno.

Função que ativa os parâmetros para o traçado de um texto na tela. Esta chama a função CURSOR, para o posicionamento inicial de escrita (x,y); em seguida, verifica os códigos ASCII (cod) dos caracteres que compõem a cadeia txt, e chama DC, para cada um dos caracteres, utilizando como parâmetro cod.

5.3) OBSERVAÇÕES

As funções primitivas gráficas, descritas anteriormente em "pseudo-código", foram implementadas em linguagem "PASCAL", uma linguagem de alto nível que fornece recursos e condições suficientes para a realização de tal evento.

As primitivas gráficas foram desenvolvidas baseadas no sistema GSPC - CORE, descrito no item 2.3.1 do capítulo 2, um núcleo de processamento gráfico capaz de interfacear um programa aplicativo diretamente com o "hardware" gráfico.

Essas funções abrangem todas as características encontradas no controlador gráfico EF9367, exceto aquelas referentes à caneta ótica ("light-pen"), uma vez que os circuitos relacionados a tal dispositivo não estão presentes na interface de visualização desenvolvida.

Acredita-se que, através dessas funções consegue-se um bom domínio das características oferecidas pela interface, bem como a possibilidade de desenvolvimento de outras funções de mais alto nível, como o que ocorre com as rotinas do sistema CORE.

CAPITULO 6 - COMENTARIOS, CONCLUSOES E PROPOSIÇÕES

6.1) COMENTARIOS

E possível fazer-se uma comparação entre algumas técnicas, em relação a quantidade de palavras de informação, necessárias a geração de figuras, ao tempo total de apresentação das mesmas e quanto as suas resoluções, tamanho da memória de quadro, a fim de se demonstrar a vantagem do sistema projetado, quanto ao aspecto de velocidade de visualização a partir de um arquivo gráfico.

Utiliza-se, para esse fim, um gráfico, conforme a figura 6.1, que apresenta como coordenadas o eixo horizontal (tempo) e o eixo vertical (média) e três segmentos interligando pontos, cujo comprimento total é aproximadamente o eixo horizontal. Foi escolhida tal figura por apresentar textos e segmentos de reta, que geralmente compõem os gráficos, e por servir para o levantamento da ordem de grandeza dos parâmetros que serão analisados.

Figura 6.1 - Gráfico de média versus tempo.

As técnicas comparadas são referentes ao (a) terminal alfanumérico, onde o gráfico apresentado é composto de caracteres em forma alfanumérica e o arquivo correspondente é uma sequência de palavras de código ASCII, (b) dados compactos, onde o gráfico mostrado no vídeo é uma aplicação um a um da memória de quadro, porém, o arquivo se apresenta em forma compacta em relação ao conteúdo dessa memória, registrando apenas as variações (derivações) dos valores correspondentes aos "pixels" e (c) primitivas gráficas, onde o gráfico é mostrado no vídeo de modo idêntico ao item anterior, porém, o arquivo se apresenta em forma de sequência de comandos para a geração de primitivas gráficas (reta, ponto e caractere).

No quadro comparativo é considerado que (a) as informações estejam inicialmente armazenadas num arquivo ("metafile") junto a algum sistema computacional, em formato apropriado para cada caso. (b) Essas informações são enviadas à interface de visualização em forma de "bytes", (c) que por sua vez faz um processamento, também apropriado, modificando a memória de quadro. (d) Após essa modificação o sistema com exploração rastreada faz com que o conteúdo da memória seja mostrado no monitor de vídeo. Essa parte é considerada igual para as três técnicas. A figura 6.2 mostra em diagrama de blocos as partes envolvidas.

Figura 6.2 - Diagrama de blocos de um sistema para visualização, a partir de um arquivo.

O quadro comparativo é o seguinte:

TECNICA	FORMATO DO ARQUIVO	INFORMAÇÕES EM BYTES	TEMPO DE APRESENTAÇÃO	RESOLUÇÃO
TERMINAL ALFANUMERICO	ASCII	190	$190 tT + tP$	80 x 25 CARACTERES
DADOS COMPACTOS	VARIAÇÕES DA MEMORIA DE QUADRO	1680	$1680 tT + tP$	1024 x 416 "PIXELS"
PRIMITIVAS GRAFICAS	COMANDOS DE PRIMITIVAS GRAFICAS	174	$174 tT + tP$	1024 x 416 "PIXELS"

O tempo de apresentação refere-se ao tempo necessário para a transferência de informações do arquivo e para a modificação da memória de quadro, uma vez que a exploração rastreada apresenta de forma imediata o gráfico no vídeo.

O parâmetro tT é o tempo de transferência de um "byte" de informação do arquivo para a interface de visualização. Nota-se, também, que o tempo de apresentação é diretamente proporcional à

quantidade de informações, caso o tempo necessário ("overhead") para processamento, t_P , na modificação da memória, se torne desprezível, conforme as considerações seguintes:

Na técnica utilizando um terminal de video alfanumérico de formato 8 x 10 pontos por matriz de caractere, tem-se 10 caracteres no eixo horizontal, 8 caracteres no vertical, 10 caracteres de segmentos e 10 das palavras de rótulo (média e tempo), no total de 38 caracteres. Considerando-se que, no pior caso, todos esse caracteres devam ser posicionados um a um, necessita-se de comando de posicionamento (um a dois "bytes"), coordenadas da posição (dois "bytes") e a palavra de código do caractere a ser escrito, ou seja, 5 "bytes" por caractere, portanto 190 "bytes".

Para a técnica que trabalha com informações em forma compacta as considerações são:

- os caracteres são matrizes de 7 x 9 pontos, sendo que na média utiliza-se, para a escrita de cada um, 32 pontos, que correspondem ao seu contorno. Tem-se, portanto, 10 caracteres de 32 pontos, ou seja, 320 pontos variáveis (a serem derivados).
- os segmentos de reta têm os seguintes comprimentos:
 - eixo horizontal : 80 pontos;
 - eixo vertical : 80 pontos;
 - segmentos : 80 pontos.

Tem-se, portanto, no total 560 pontos variáveis em toda a figura que devem ser derivados. Considerando-se o pior caso, em que cada ponto representa uma variação e portanto um item no

arquivo, e que cada variação é representada pela sua posição (2 "bytes") e seu valor, ou seja, a cor do "pixel", (um "byte"), tem-se 3 "bytes" por ponto, ou seja, 1680 "bytes" no total do arquivo.

Para o método usando primitivas gráficas, o cálculo do número de palavras, necessárias para o traçado do gráfico, é feito baseado na interface de visualização desenvolvida e descrita neste trabalho. Sendo assim, para se encontrar esse número basta saber quantas vezes os registradores internos do controlador gráfico EF9367, que compõe a interface, deverão ser programados.

Então, para se traçar o referido gráfico, serão utilizadas 174 palavras de informação, considerando-se o pior caso, em que todos os caracteres e segmentos são posicionados individualmente.

Para essas técnicas pode-se considerar que o tempo de apresentação é igual ao de transferência das palavras de informação, desprezando-se o tempo de geração das figuras pelas interfaces de visualização, ou seja, o tempo de processamento para a modificação da memória de quadro, $t_P = 0$.

Isso porque o tempo t_P só é considerável para o sistema de primitivas gráficas, uma vez que, nesse caso, as primitivas são traçadas após o recebimento dos comandos vindos do arquivo.

Por exemplo, na figura 6.2, que é composta de 560 pontos, conforme cálculo anterior, considerando-se que o controlador gráfico EF9367 seja capaz de traçar um único ponto em torno de $1,3 \mu s$, o tempo total de geração do gráfico é de aproximadamente

0,7 ms.

Dependendo do valor de tT , o tempo de geração encontrado deve ou não ser considerado.

Por exemplo, se o valor do parâmetro tT for de 1 ms, valor frequentemente encontrado na prática, o tempo de geração do gráfico por este método (0,7 ms) também tornar-se-á desprezível em relação ao de transmissão das palavras (174 ms), como foi considerado aqui.

Sendo assim, o tempo de apresentação do gráfico gerado através do método que usa primitivas gráficas é de $174 tT$.

Baseado neste quadro, podem ser tiradas algumas conclusões, que são descritas a seguir.

O custo total do sistema a ser utilizado varia, basicamente, com o tamanho da memória de quadro, ou seja, com a resolução das figuras a serem mostradas, uma vez que os demais componentes das interfaces têm custo de mesma ordem.

Sendo assim, a técnica que apresenta menor custo é a que utiliza um terminal de vídeo alfanumérico; quanto a isso, as outras duas técnicas se equivalem.

Mas, a técnica utilizando um terminal de vídeo alfanumérico tem um grande inconveniente que é a falta de precisão na apresentação de figuras, um problema que não ocorre nas demais.

Ainda, pelo quadro pode-se verificar que a quantidade de palavras de informação que deve ser transferida do arquivo para a

interface de visualização é bem menor no método que usa primitivas gráficas, em relação a técnica que trabalha com dados compactos, tornando, assim, menor o tempo de apresentação do gráfico com aquele método.

Com base nestas observações, pode-se concluir que a técnica que utiliza primitivas gráficas é a que apresenta melhor característica custo x desempenho.

6.2) CONCLUSOES

Como conclusão deste trabalho descreve-se que, o sistema da Interface de Visualização Gráfica desenvolvido está testado e em funcionamento, tanto no que se refere a implementação física, o "hardware", como na parte da programação básica, o "software".

Caracterizando sua implementação física, o capítulo 4 apresenta o detalhamento do projeto da interface.

Descreve-se, neste capítulo, que esta pode gerar figuras com resolução de 1024 x 416 pontos visíveis, resolução esta considerada alta e que proporciona uma boa qualidade da imagem mostrada.

Ainda, pelo fato do projeto da interface ser modular, esta foi implementada usando três páginas de memória, o que possibilita a obtenção de oito níveis de intensidade ou cores, para aplicações monocromáticas ou coloridas.

Em geral, esse número é suficiente para muitas aplicações práticas; mas, se for desejado, o controlador gráfico EF9367 permite um acréscimo do número de páginas, viabilizando um

aumento do número de níveis de intensidade ou de cores, devendo, obviamente, fazer-se pequenas alterações no projeto inicial da interface.

Alguns sistemas gráficos comerciais para computadores pessoais (PCs) apresentam resolução média de 320 x 200 pontos visíveis para 16 cores e alta de 640 x 200 pontos para uma cor. Assim, pode-se observar que a IVG desenvolvida proporciona uma característica resolução x cores bem melhor que os sistemas citados, pois aquela possui resolução de 1024 x 416 pontos e a possibilidade de se obter um número muito grande de níveis de intensidade ou cores, através de expansão do número de páginas de memória, obtendo-se, ainda, uma imagem sem distorções. Isso não é conseguido com os sistemas comerciais devido às suas características não-modulares.

Com relação à programação, como descrito no capítulo 5, foi implementado um sistema programacional básico.

O sistema utiliza as primitivas implementadas em "hardware" do CI EF9367, liberando o processador "host" para outras tarefas em paralelo aumentando o desempenho global do sistema, diferente do que ocorre com os sistemas comerciais citados que realizam as primitivas por "software".

Esse conjunto de funções gráficas dedicadas permite que se utilize todas as características fornecidas pela interface, fazendo com que esta trabalhe rápido e corretamente, além de possibilitar o desenvolvimento de outras funções a partir desse conjunto inicial.

A interface de visualização e a programação básica correspondente permitem a geração e a apresentação de textos e figuras muito rapidamente. Esta interface utiliza um monitor de vídeo como dispositivo de saída, o que contribui bastante para agilizar o processo de visualização de figuras.

Baseado no quadro comparativo apresentado no item anterior, e nas conclusões obtidas, pode-se descrever que a interface de visualização desenvolvida apresenta um bom desempenho no traçado de figuras.

6.3) PROPOSIÇÕES

Certamente podem ser feitas alterações na interface de visualização, de modo a proporcionar uma melhoria no seu desempenho.

Existem algumas proposições para a continuação deste trabalho, que consistem em se fazer alterações na interface, e que são descritas a seguir.

Inicialmente, pretende-se aumentar o número de páginas de memória, a fim de incrementar o número de níveis de intensidade ou cores, atualmente igual a oito.

Com isso, a interface poderá ser utilizada em aplicações mais sofisticadas, como apresentação de formas com aspecto de cenas reais, por exemplo sombreamento.

O computador utilizado no projeto inicial tem como processador principal o Z80A, um componente que trabalha com palavras de 8 bits. A fim de possibilitar a utilização da

interface em outros ambientes de trabalho, permitindo maior flexibilidade e compatibilidade com outros sistemas, pretende-se, também, adaptá-la a trabalhar com sistemas que operam com palavras de 16 bits.

Pode-se tentar, ainda, levando-se em conta uma análise de custo x desempenho utilizar outros componentes que agilizem o processo de recebimento e interpretação de informações, e o consequente traçado e apresentação de figuras.

Uma forma é utilizar, como componente base da interface, um outro circuito integrado (CI), de modo que este faça, além da interpretação, a busca ("fetch") da informação numa memória dedicada, liberando o processador principal ("host") do sistema da tarefa de controlar a interface.

Um exemplo deste tipo de CI é o TMS34010 da TEXAS, um componente bastante poderoso e, que pode, também, atuar sozinho no processamento de informações e no traçado das formas correspondentes.

Assim, uma outra proposta de continuação do trabalho é a troca do elemento base da interface de visualização, o controlador gráfico EF9367, por outro CI, por exemplo um processador gráfico, a fim de se verificar até que ponto esta substituição é viável e quais alterações isso provoca em termos de desempenho da interface.

Ainda, poderão ser promovidas outras alterações no projeto da interface a fim de permitir a utilização de outros dispositivos gráficos, tanto para entrada como para saída de informações.

BIBLIOGRAFIA

- AS86 ASAL, MIKE; SHORT, GRAHAM; PRESTON, TOM; SIMPSON, RICHARD;
ROSKELL, DEREK; GUTTAG, KARL - THE TEXAS INSTRUMENTS 34010
GRAPHICS SYSTEM PROCESSOR - IEEE Computer Graphics &
Applications, USA, Outubro 1986, Vol. 6 - N. 10, p. 24 - 39.
- BL82 BLINN, JAMES F. - RASTER GRAPHICS - IEEE TUTORIAL: Computer
Graphics por Kellogg S. Booth, USA, 1982, p. 207 - 213.
- BO86 BONO, PETER R. - GUEST EDITOR'S INTRODUCTION GRAPHICS
STANDARDS - IEEE Computer Graphics & Applications, USA,
Agosto 1986, Vol. 6 - N. 8, p. 12 - 16.
- CA86 CARSON, GEORGE S.; MCGINNIS, EILEEN - THE REFERENCE MODEL
FOR COMPUTER GRAPHICS - IEEE Computer Graphics &
Applications, USA, Agosto 1986, Vol. 6 - N. 8, p. 17 - 23.
- CL86 CLARKSON, THOMAS B. - THE IMPACT OF HIGH-PERFORMANCE
GRAPHICS TECHNOLOGY - Computer Design, USA, Dezembro 1986,
Vol. 25 - N. 22, p. 131.
- DE83 DePALMA, GARY; OLSON, MARK; JOLLIS, ROGER - DEDICATED VLSI
CHIP LIGHTENS GRAPHICS DISPLAY DESIGN LOAD - Electronic
Design, USA, Janeiro 1983, Vol. 31 - N. 2, p. 131 - 139.
- FO82 FOLEY, J.D.; Van Dan, A. - FUNDAMENTALS OF INTERACTIVE
COMPUTERS GRAPHICS - Addison - Wesley Publishing Company,
USA, 1982.

- GU86 GUTTAG, KARL; VAN AKEN, JERRY; ASAL, MICHAEL - REQUIREMENTS FOR A VLSI GRAPHICS PROCESSOR - IEEE Computer Graphics & Applications, USA, Janeiro 1986, Vol. 6 - N. 1, p. 32 - 47.
- HA83 HARRINGTON, STEVEN - COMPUTERS GRAPHICS - A Programming Approach - McGraw - Hill, USA, 1983.
- HE86 HENDERSON, LOFTON; JOURNEY, MARGARET; OSLAND, CHRIS - THE COMPUTER GRAPHICS METAFILE - IEEE Computer Graphics & Applications, USA, Agosto 1986, Vol. 6 - N. 8, p. 24 - 32.
- HO83 HOPGOOD, F.R.A.; DUCE, D.A.; GALLOP, R.J.; SUTCLIFFE, D.C. - INTRODUCTION TO THE GRAPHICAL KERNEL SYSTEM (GKS) - Academic Press, Great Britain, 1983.
- IN83 MANUAL INTEL - 82720 GRAPHICS DISPLAY CONTROLLER - INTEL Corporation, USA, Junho 1983.
- JA87 JAMES, J.M. - THE THIRD GENERATION OF PC GRAPHICS CONTROLLERS - IEEE Computer Graphics & Applications, USA, Outubro 1987, Vol. 7 - N. 10, p. 24 - 27.
- KA80 KANE, GERRY - THE CRT CONTROLLER HANDBOOK - Osborn / McGraw - Hill, USA, 1980.
- LU78 LUCIDO, ANTHONY P. - AN OVERVIEW OF DIRECTED BEAM GRAPHICS DISPLAY HARDWARE - Computer - IEEE, USA, Novembro 1978, Vol. 11 - N. 11, p. 29 - 36.
- MA77 MACHOVER, CARL; NEIGHBORS, MICHAEL; STUART, CHARLES - GRAPHICS DISPLAYS - IEEE Spectrum, USA, Agosto 1977, Vol. 14 - N. 8, p. 24 - 32.

- MA78 MATHERAT, PHILIPPE - Tese de Doutorado Terceiro Ciclo junto à "L'UNIVERSITE PIERRE ET MARIE CURIE, PARIS VI" - França, 19 Maio 1978.
- MA86 MAGALHAES, LEO PINI - COMPUTAÇÃO GRAFICA - INTERFACES EM SISTEMAS DE COMPUTAÇÃO GRAFICA - Editora da UNICAMP / Editora Papirus, Brasil, 1986.
- MO88 MOKHOFF, NICOLAS - GRAPHICS CHIPS FORGE HIGH-RES BOARDS FOR PCs, WORKSTATIONS - Electronic Design, USA, 17 Março 1988, Vol.36 - N. 6, p. 62 - 72.
- NE82 MANUAL NEC - mPD7220 GDC DESIGN MANUAL - Version 3 - NEC ELECTRONICS USA Inc, USA, 1982.
- NI87 NICHOLLS, BILL - INSIDE THE 82786 GRAPHICS CHIP - Byte, USA, Agosto 1987, Vol. 12 - N. 9, p. 135 - 141.
- NI88 NICOUD, J.D. - VIDEO RAMs: STRUCTURE AND APPLICATIONS - IEEE Micro, USA, Fevereiro 1988, Vol. 8 - N. 1, p. 8 - 27.
- OH86 OHR, STEPHAN - HOW SILICON ICs ARE RESHAPING THE GRAPHICS PICTURE - Electronic Design, USA, 26 Junho 1986, Vol. 34 - N.15, p. 72 - 79.
- OL83 OLSON, MARK; MAY, BRAD - GRAPHICS CHIP MAKES LOW-COST, HIGH RESOLUTION COLOR DISPLAYS POSSIBLE - Digital Design, USA, Abril 1983.

- PO86 POWERS, THOMAS; FRANKEL, ANDREA; ARNOLD, DAVID - THE
COMPUTER GRAPHICS VIRTUAL DEVICE INTERFACE - IEEE Computer
Graphics & Applications, USA, Agosto 1986, Vol. 6 - N. 8,
p. 33 - 41.
- PU86 PUK, RICHARD F.; McCONNELL - GKS - 3D: A THREE-DIMENSIONAL
EXTENSION TO THE GRAPHICAL KERNEL SYSTEM - IEEE Computer
Graphics & Applications, USA, Agosto 1986, Vol. 6 - N. 8,
p. 42 - 49.
- RA86 RANDALL, MARTIN - GRAPHICS IC SIMPLIFIES DESIGN, BOOSTS
PERFORMANCE - Computer Design, USA, 1 Junho 1986,
Vol. 25 - N.11, p. 73 - 77.
- RA86a RANDALL, MARTIN; JOHARY, ARUN - GRAPHICS COPROCESSOR CHIP
GIVES SMALL SYSTEMS A LOOK AT THE BIG PICTURE - Electronic
Design, USA, 29 Maio 1986, Vol. 34 - N.12, p. 159 - 166.
- SH83 SHUEY, DAVID; BAILEY, DAVID; MORRISSEY, THOMAS P. - PHIGS:
A STANDARD, DYNAMIC, INTERACTIVE GRAPHICS INTERFACE - IEEE
Computer Graphics & Applications, USA, Agosto 1983,
Vol. 6 - N. 8, p. 50 - 57.
- SH86 SHIRES, GLEN - A NEW VLSI GRAPHICS COPROCESSOR - THE INTEL
82786 - IEEE Computer Graphics & Applications, USA, Outubro
1986, Vol. 6 - N. 10, p. 49 - 55.
- SK86 SKALL, MARK W. - NBS's ROLE IN COMPUTER GRAPHICS STANDARDS
- IEEE Computer Graphics & Applications, USA, Agosto 1986,
Vol. 6 - N. 8, p. 66 - 70.

- SP86 SPARKS, MADELEINE R.; GALLOP, JULIAN R. - LANGUAGE BINDINGS FOR COMPUTER GRAPHICS STANDARDS - IEEE Computer Graphics & Applications, USA, Agosto 1986, Vol. 6 - N. 8, p. 58 - 65.
- TC84 TECHNICAL COMMITTEE X3H3 - COMPUTER GRAPHICS - INFORMATION PROCESSING SYSTEMS' COMPUTER GRAPHICS - GRAPHICAL KERNEL SYSTEM, X3H3/83 - USA, 1984.
- TE86 MANUAL TEXAS - TMS34010 GRAPHICS SYSTEM PROCESSOR - TEXAS Instruments, USA, Janeiro 1986.
- TE86a TEXAS INSTRUMENTS, SEMICONDUCTOR GROUP - 32-BIT GRAPHICS PROCESSOR FEATURES ON-CHIP DISPLAY CONTROL - Computer Design, USA, Dezembro 1986, Vol. 25 - N. 22, p. 140.
- TH83 MANUAIS THOMSON SEMICONDUCTORS - APPLICATION NOTE - INTERFACING EF9365, EF9366 AND EF9367 GRAPHIC DISPLAY PROCESSOR TO Z80A, 8085A, 6502 AND 2650A-1 MICROPROCESSORS;
 - EF9365 - EF9367 EXTERNAL SYNCHRONIZATION;
 - USING 64 KBIT DYNAMIC RAMs WITH EF9365, EF9366, EF9367 GRAPHIC DISPLAY PROCESSOR;
 - EF9367 GRAPHIC DISPLAY PROCESSOR APPLICATION PRINCIPLES;
 France, 1983.
- TH83a MANUAL THOMSON SEMICONDUCTORS - GRAPHIC DISPLAY PROCESSOR - EF9367 ADVANCE INFORMATION - France, 1983.
- TI86 TITUS, JON - VERSATILE ADD-IN BOARDS ACQUIRE AND PROCESS VIDEO IMAGES AND GRAPHICS IN REAL TIME - EDN, USA, 2 Agosto 1986, p. 103 - 110.

- VA86 VAZ, MARIA FERNANDA RODRIGUES - PADRONIZAÇÃO DA INTERFACE COM DISPOSITIVOS GRAFICOS - Boletim SCOPUS - número 101, Brasil, Novembro 1986.
- WI86 WIENTJES, BRENT; GUTTAG, KARL; ROSKELL, DEREK - FIRST GRAPHICS PROCESSOR TAKES COMPLEX ORDERS TO RUN BIT-MAPPED DISPLAYS - Eletronic Design, USA, 23 Janeiro 1986, Vol. 34 - N. 2, p. 73 - 81.
- WI86a WILLIAMS, TOM - DGIS LINKS PC-DOS TO HIGH-PERFORMANCE GRAPHICS - Computer Design, USA, 15 Maio 1986, Vol. 25 - N. 10, p. 32 - 33.
- WI86b WILLIAMS, TOM - GRAPHICS DEVELOPMENTS GEARED TOWARD SPEED AND CAPACITY - Computer Design, USA, Dezembro 1986, Vol. 25 - N. 22, p. 127 - 134.
- WI86c WILLIAMS, TOM - HARDWARE ADVANCES FUEL GRAPHICS STANDARDS CONTROVERSY - Computer Design, USA, 15 Agosto 1986, Vol. 25 - N. 15, p. 28 - 32.
- WI87 WILLIAMS, TOM - GRAPHICS ICs OFFER NEW SOLUTIONS FOR SPEED AND FLEXIBILITY - Computer Design, USA, 1 Junho 1987, Vol. 26 - N. 11, p. 40 - 46.
- WI88 WILSON, RON - CRT CONTROLLER CHIP TAKES ON WINDOW-MANAGEMENT OVERHEAD - Computer Design, USA, 15 Junho 1988, Vol. 27 - N. 12, p. 27 - 28.

APENDICE

PROJETO : CONTROLADOR GRÁFICO
PLACA CONTROLADOR GRÁFICO
PROJETISTA: CARLOS NORBERTO FISCHER

LISTA DE MATERIAL

COMPONENTES

SD = 74LS42
P1,P2,P3 = 74LS08
MM = 74LS123
F1, F2 = 74LS74
B1, B2 = 74LS244
DE = 74LS138
P4, P5 = 74LS32
CO = 74LS163A
TB = 74LS245
L1,L2,L3 = 74LS373
PR = 74LS288
CG = EF9367
RE = 74LS374
SQ = 74LS266
S1,S2,S3,S4 = 74LS157
P6 = 74LS04
P7 = 74LS00
XI = CRISTAL DE 23.916 MHz
CE = CAPACITOR DE ENTRADA
IE = INDUTOR DE ENTRADA
R1,R2,R3,R4 = RESISTORES DE 5,6 KΩ
R5 = RESISTOR DE 4,7 KΩ
R6,R7,R8,R10 = RESISTORES DE 1 KΩ
R9 = RESISTOR DE 100 Ω
C1 = CAPACITOR DE 150 pF
C2,C3 = CAPACITORES DE 10 pF
CH1 A CH4 = CHAVES

SOQUETES

1 WW DE 40 PINOS
7 WW DE 20 PINOS
11 WW DE 16 PINOS
8 WW DE 14 PINOS

PROJETO : CONTROLADOR GRÁFICO
DISTRIBUIÇÃO DOS COMPONENTES
PLACA CONTROLADOR GRÁFICO
PROJETISTA: CARLOS NORBERTO FISCHER

PROJETO : CONTROLADOR GRÁFICO

PLACA MEMÓRIA DE QUADRO

PROJETISTA: CARLOS NORBERTO FISCHER

LISTA DE MATERIAL

COMPONENTES

M1 A M24 = MCM 6665
D1, D2, D3 = 74LS165
RE = 74LS175
PN = 74LS25
CE = CAPACITOR DE ENTRADA
IE = INDUTOR DE ENTRADA
R1, R3, R5, R8 = RESISTORES DE 10KΩ
R2, R4, R6 = RESISTORES DE 560Ω
R7, R9 = RESISTORES DE 1KΩ
P1, P2, P3, P7 = POTENCIOMETROS DE 10KΩ
P4, P5, P6 = POTENCIOMETROS DE 4,7KΩ
T1 A T7 = TRANSISTORES TIPO 2N2369

SOQUETES

1WW DE 14 PINOS
28WW+ DE 16 PINOS

PROJETO : CONTROLADOR GRÁFICO

DISTRIBUIÇÃO DOS COMPONENTES

PLACA MEMÓRIA DE QUADRO

PROJETISTA : CARLOS NORBERTO FISCHER

PLACA DA MEMORIA GRAFICA.