UNIVERSIDADE DO PORTO

A ARQUITECTURA DE SISTEMAS GRÁFICOS

FACULDADE DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

Rua dos Bragas - 4099 PORTO CODEX - PORTUGAL

Telef. 27505 - Telex 27323 FEUP P

Vítor Manuel Correia da Silva Santos

A ARQUITECTURA DE SISTEMAS GRÁFICOS

043 M 5239a

681.3(043)/SAN/ARQ cat

UNIVERSIDADE DO PORTO
Faculdade de Engenharia
BIBLIOTECA
N.º 22663
CDU
Data 02/06/1998

Dissertação apresentada à

UNIVERSIDADE DO PORTO

em preenchimento parcial dos requisitos para
obtenção do grau de Mestre em
Sistemas e Computadores Digitais

FACULDADE DE ENGENHARIA

JUNHO 1987

à minha filha, Júlia Cristina

Agradecimentos

- ao meu supervisor Professor Doutor Nunes Ferreira, pela orientação, estímulo, apoio e disponibilidade que, muito para além do que seria legítimo esperar-se, prestou à orientação deste trabalho.

- Ao Instituto Nacional de Engenharia de Sistemas e Computadores, INESC, pelas facilidades concedidas.

- Ao Conselho Científico do Instituto Superior de Engenharia do Porto, e em particular ao seu Presidente Engenheiro Carlos Ferreira, pelo apoio e estímulo concedidos.

- Aos colegas do Gabinete de Sistemas do ISEP, pelas sugestões e críticas feitas ao longo deste trabalho.

- A todos os colegas e amigos que tiveram, quotidianamente, palavras de apoio e estímulo.

I N D I C E

CAPÍTULO 1: Contexto e Objecto do trabalho 1.1

1.1 - Introdução 1.1
1.2 - Sistemas gráficos: uma classificação 1.3
1.3 - Estações gráficas 1.4
1.4 - O futuro imediato na arquitectura dos sistemas gráficos 1.6
1.5 - As últimas soluções 1.7
1.5.1 - A memória 1.7
1.5.2 - A serialização 1.8
1.5.3 - O monitor 1.9
1.6 - O panorama geral 1.11
1.7 - O objecto do trabalho 1.12

CAPÍTULO 2: A arquitectura dos sistemas gráficos 2.1

2.1 - Introdução 2.1
2.2 - Um modelo de arquitectura de sistemas gráficos 2.1
2.2.1 - O modelo geral 2.2
2.2.2 - O modelo físico 2.3
2.2.3 - Os processos e estruturas de dados 2.5
2.3 - Conceitos básicos relativos aos subsistemas gráficos 2.8
2.4 - Duas questões fundamentais 2.15
2.4.1 - O balanço hardware-firmware-software 2.15
2.4.2 - A largura de banda do barramento de memória 2.16

CAPÍTULO 3: Descrição e análise de duas implementações de uma arquitectura de terminais gráficos 3.1

3.1 - Introdução 3.1
3.2 - A primeira implementação do terminal gráfico 3.5
3.2.1 - O controlador gráfico uPD7220, da NEC 3.9
3.2.2 - A memória TMS4416 da Texas Instruments 3.18
3.2.3 - O conversor d/a VDA444TD da Intech 3.18
3.2.4 - O princípio de funcionamento 3.19
3.3 - A segunda implementação do terminal gráfico 3.24
3.3.1 - O Interface Multiprocessador TMS9650 da Texas Instruments 3.27
3.3.2 - O controlador de vídeo TMS34061 da TI 3.27
3.3.3 - A vídeo-RAM TMS4161 da TI 3.29
3.3.4 - O conversor vídeo DAC3408 da Intech 3.30
3.3.5 - O princípio de funcionamento 3.31
3.4 - Uma avaliação das arquitecturas testadas 3.34
3.5 - Conclusão 3.41
3.5.1 - O tratamento unificado de texto e imagem 3.42
3.5.2 - As características desejáveis para os processadores gráficos 3.43

CAPÍTULO 4: O estado-da-arte dos dispositivos para subsistemas gráficos 4.1

4.1 - Introdução 4.1
4.2 - O controlador gráfico 82786, da Intel 4.2
4.2.1 - Arquitectura interna do 82786 4.3
4.2.2 - O firmware do 82786 4.9
4.3 - O controlador gráfico HD63484 da Hitachi 4.12
4.3.1 - Arquitectura interna do 63484 4.13
4.3.2 - O firmware do 63484 4.17
4.4 - O controlador gráfico Am95C60, da AMD 4.18
4.4.1 - Arquitectura interna do 95C60 4.18
4.4.2 - O firmware do 95C60 4.20

4.5 - O processador gráfico TMS34010, da TI 4.21
4.5.1 - O GSP como processador de uso geral 4.22
4.5.2 - O GSP como processador gráfico 4.23
4.5.1 - A arquitectura interna do GSP 4.23
4.5.4 - O software do GSP 4.27
4.5.4 - As instruções gráficas do GSP 4.29
4.6 - Avaliação comparativa dos processadores e controladores gráficos 4.33
4.7 - A memória gráfica 4.36
4.8 - Os conversores vídeo d/a 4.39
4.8.1 - O conversor vídeo Bt451, da Brooktree 4.41
4.8.2 - Os conversores vídeo DAC3401 da Honeywell e Bt108 da Brooktree 4.43

CAPÍTULO 5: O desenvolvimento a curto prazo dos sistemas gráficos 5.1

5.1 - Introdução 5.1
5.2 - Uma "nova" classificação dos sistemas gráficos 5.2
5.3 - Esboço de projecto para um sistema gráfico usando uma arquitectuta "fácil" 5.5
5.3.1 - Os dispositivos utilizados 5.6
5.3.2 - O esquema geral do subsistema gráfico 5.7

CAPÍTULO 6: Conclusão 6.1

CAPÍTULO 1: Introdução: Contexto e Objecto do trabalho.

1.1 - Introdução

Provavelmente a nossa época virá a ser conhecida como a era da Revolução Informática e o computador verá ser-lhe atribuído um estatuto idêntico, embora temporalmente diferido, ao da máquina a vapor.
De facto se, um pouco arbitrariamente fixarmos o início da Idade Moderna do Computador nos fins da década de 40, com o aparecimento do EDSAC (Electronic Delay Storage Automatic Computer), vemos que apenas quatro décadas, o tempo médio de um conflito entre povos na História Moderna, foram suficientes para passar do estádio incipiente do cálculo logístico simples aos últimos desenvolvimentos da Inteligência Artificial, com o computador manipulando elevadíssimos volumes de informação.
Poderia pensar-se que só recentemente, e devido a estes elevados caudais de informação, se colocou ou agudizou o problema relacionado com a efectiva tansferência de informação entre o utilizador e a máquina, e que esse facto levou a que, só recentemente, os investigadores tenham entendido a necessidade de desenvolver técnicas de computação gráfica capazes de fornecer um adequado interface entre ambos.
Talvez, no entanto, não seja assim. Se fizermos um traçado dos correntemente considerados marcos da indústria gráfica (19), verificamos que o primeiro padrão se coloca a meio da década de 50, com o estabelecimento de uma ligação directa entre um computador, osciloscópios e outros dispositivo de natureza funcional semelhante, realizada no MIT. Assim, nesta focagem puramente histórica se afirma a preocupação, havida desde sempre, da utilização de dispositivos gráficos

para o diálogo homem-máquina. A capacidade de síntese de uma imagem, a riqueza de informação que é capaz de conter e o pré-histórico hábito da sua utilização na comunicação com outrem, tornavam a computação gráfica demasiado tentadora para que não fosse usada desde início.
Claro que a experiência do MIT tem o sabor das coisas antigas, deliciosas mas imprestáveis e, de facto, só na década de 70 se deram passos reais no desenvolvimento da computação gráfica, ancorados na realização, em 1974, da primeira conferência anual do SIGGRAPH. Desde essa altura o esforço efectuado em investigação nesta área pode ser indiciado pelo número de participantes na conferência: cerca de 500 em 1974, mais de 27.000 em 1985 (21). Hoje, as áreas envolvidas na utilização da computação gráfica são extensas, indo desde aplicações gerais de gestão e burótica, executadas em computadores pessoais ou em sistemas de médio desempenho, até aplicações de projecto eléctrico e mecânico assistido por computador (CAD), e de fabricação assistida por computador (CAM), executadas em sistemas de médio desempenho, ou aplicações especializadas tais como a fábrica automática, modelação de sólidos, simulação ou animação artística, necessitando de sistemas de alto desempenho.

Os factores fundamentais, determinantes da evolução verificada que conduziu ao estado de desenvolvimento actual, parecem radicar numa série de inovações havidas durante os anos de 1984/1986, quer na área de hardware quer na área de software, com elevado impacto na relação custo/desempenho e acessibilidade (25).
Foi extremamente difícil acompanhar esta evolução,marcada essencialmente pela criação de uma nova geração de processadores gráficos, alguns do quais já considerados como padrões de facto (TMS 34010, Hitachi 63484 e i82786) e pela aceitação, por parte dos integradores de sistemas gráficos, dos padrões de software para aplicações gráficas (GKS,

PHIGS, IGES, CORE, NAPLPS), incluindo o desenvolvimento de suporte em hardware para aqueles padrões.
Ainda não está reflectido, hoje, e em termos de produto final -sistemas de computação gráfica- o impacto destes avanços tecnológicos, ou pelo menos ainda não é possível efectuar o seu balanço. As próximas conferências do SIGGRAPH, mostrarão certamente os seus resultados.

1.2 - Sistemas gráficos: uma classificação

Actualmente, talvez não seja demasiado polémico considerar três classes de sistemas gráficos. A classe de mais elevado desempenho, com sistemas gráficos cuja gama de custos vai de US$20.000 a mais de US$100.000, inclui computadores multi-utilizador ou estações de trabalho mono-utilizador e são usados nas áreas da arte, simulação e CAD/CAM/CAE e, em geral, sempre que é necessária a geração realista de imagens. Normalmente dispõem de resolução superior a 1024x1024 pixels, e paletas de cor implementadas com 24 bits/pixel ou mais; a classe de médio desempenho integra aplicações de CAD/CAM/CAE menos exigentes, de burótica, de processamento sofisticado de texto e imagem, e de Inteligência Artificial, e o seu custo situa-se na gama dos US$2.000 a US$15.000, dispondo de resoluções desde 640x480 pixels até 1024x800 pixels, e paletas de cor de 256 cores seleccionáveis numa gama mais elevada; finalmente, a classe de baixo desempenho integra as aplicações executadas principalmente sobre computadores pessoais, e o seu custo estende-se desde US$200 até aos US$1.500 (17).

Esta classificação torna aparente a relação entre as necessidades da aplicação gráfica e o binómio custo/desempenho do sistema gráfico. A necessidade fundamental, nas aplicações gráficas executadas na classe superior dos sistemas gráficos, refere-se à capacidade de

manipular, com um nível de interacção adequado, grandes volumes de dados. Esta manipulação diz respeito não só à apresentação dos dados resultantes do processamento, mas também à possibilidade de modificar essa apresentação, permitindo ao utilizador mover, decompor, rodar, etc. o objecto apresentado.
Fazer isto em tempo real significa dispor de um sistema computacional de elevado desempenho, logo de elevado custo. A única solução actual, ou pelo menos até um passado recente, passava pela ligação de um terminal gráfico de tecnologia avançada a um sistema computacional de grande porte. Esta opção é extremamente desvantajosa pois que, dado que as aplicações gráficas são essencialmente interactivas, o balanço da carga do sistema é irregular. Em determinados momentos toda a potência de cálculo do sistema está em uso, em geral, excluindo outros utilizadores e, seguidamente, o subsistema gráfico fica à espera que o utilizador da aplicação gráfica inicie a próxima actividade.
Além disso, o trânsito de informação entre o computador central e o terminal, que necessariamente deve ser feito a alta velocidade, utiliza normalmente o modo DMA, o que se traduz em diminuição do desempenho do sistema central, por saturação do barramento de comunicação.

1.3 - Estações gráficas

Uma solução possível para evitar os inconvenientes relacionados com a utilização de um subsistema gráfico no computador central passa pela implementação de estações de trabalho gráficas. As estações gráficas utilizam a "display-list" existente no sistema central, a qual é uma descrição ponto a ponto de um objecto em coordenadas definidas pelo utilizador, transferem-na para a sua própria memória e executam sobre elas a computação necessária para execução

dos procedimentos gráficos. A "display list" é enviada do sistema central à estação gráfica uma única vez, no início da sessão e, eventualmente no fim da sessão, se houve modificações, é reenviada da estação gráfica para o sistema central. A estação gráfica dispõe de capacidade de processamento gráfico, que pode ser activada quer localmente, a partir dos dispositivos de entrada/saída habituais, quer remotamente por meio de uma sequência de comandos enviados pelo sistema central.

Com este tipo de arquitectura o balanço global da carga do sistema torna-se mais favorável, dado que o subsistema gráfico solicita em menor grau o sistema central, mas tem como consequência impor a existência de capacidade de processamento gráfico potente na estação de trabalho. A utilização de processadores de uso geral, mesmo os mais avançados tais como, por exemplo, os da família 68000 da Motorola, apenas conseguem, em geral, colocar os sistemas gráficos na classe de médio desempenho, mesmo quando complementados com algum tipo de dispositivos coprocessadores ou aceleradores.
As estações gráficas da classe mais elevada usam implementações baseadas em sistemas computacionais de grande potência, por exemplo, o Cray XMP, sistemas distribuídos, ou sistemas com organização SIMD (Single Instruction Multiple Data) de processadores; por seu turno, as estações da classe inferior são actualmente implementadas à base de controladores gráficos, extensamemte desenvolvidos e comercializados nos últimos anos, e por isso bem conhecidos e utilizados em arquitecturas que vão desde os sistemas com um único controlador até aos sistemas distribuídos com vários controladores.

1.4 - O futuro imediato na arquitectura dos sistemas gráficos

A perspectiva actual parece ser a de alargar a classe intermédia dos sistemas gráficos ao processamento de aplicações até agora inseridas no segmento inferior da classe mais elevada. Tal parece ser possível com a utilização dos novos processadores gráficos já comercializados ou em vias de comercialização. Estes processadores gráficos distinguem-se dos controladores gráficos pela sua programabilidade, no sentido em que extraem e executam autonomamente as suas instruções.
Os processadores gráficos de maiores possibilidades, até agora tornados disponíveis pelos fabricantes, são o TMS34010 e o 85XX da National Semicondutor's.

Simultaneamente, dipõe-se actualmente de controladores mais potentes tais como o 82786 da Intel, o HD63484 da Hitachi e o 95C60 da AMD. Estes dispositivos gráficos têm características de programabilidade, rapidez e capacidade de processamento gráfico de base, nomeadamente a possibilidade de geração de linhas, arcos e polígonos, de enchimento de polígonos e áreas e de transferência de blocos de bits, BitBlT (bit boundary block transfer), possibilitando a transferência de uma matriz de pixels para um ponto qualquer da memória gráfica, embora difiram quanto ao nível de programabilidade, ao número de "bit planes" que podem controlar, à técnica de implementação de janelas e à implementação da operação de BitBlT.

1.5 - As últimas soluções

Os desenvolvimentos havidos na área dos controladores e processadores gráficos colocaram novas questões no projecto de arquitectura de sistemas gráficos relacionados com a obtenção de um elevado desempenho global do sistema. Essas questões dizem respeito, essencialmente, a mais três pontos chave do projecto: o sistema de memória, os dispositivos terminais de ataque ao monitor (serialização), e o monitor propriamente dito. Em todos estes campos foram propostas e vulgarizadas, nos dois últimos anos, novas soluções.

1.5.1 - A Memória

No que respeita ao sistema de memória, a par dos aumentos de densidade e do aumento de complexidade dos dispositivos de detecção de erros, verificaram-se também desenvolvimentos significativos nas velocidades disponíveis nos controladores de memória dinâmica (8). Mas o facto mais saliente foi o aparecimento de RAMs estáticas de dois portos (6), que veio solucionar de modo satisfatório um dos grandes problemas das arquitecturas de sistemas gráficos em termos de desempenho: a resolução dos problemas de arbitragem e contenção no barramento, relacionados com a pequena largura de banda dos barramentos de acesso comum do processador gráfico e do dispositivo de serialização.
O passo lógico seguinte, que fornece uma solução satisfatória pelo menos ao nível dos sistemas gráficos da classe intermédia, foi a montagem de um registo de deslocamento (shift-register) em lugar do banco de registos ou do banco FIFO da memória de duplo porto, construindo assim as designadas memórias vídeo (VRAM), das quais os

modelos que mais se têm imposto parecem ser a Am90C644 (17) da Advanced Micro Devices, a uPD41264C da Nec (17) e a TMS4161 da Texas Instruments.
A arquitectura das VRAMs, aliadas ao desenvolvimento geral da tecnologia CMOS no que respeita essencialmente à velocidade e ao consumo, veio permitir a obtenção de medianas densidades e altas velocidades, comparáveis pelo menos à da tecnologia TTL, sendo possível configurar arquitecturas de sistemas com valores de largura de banda do barramento tão elevadas quanto 5 Gbit/s (17). Assim, as memórias estáticas RAM de duplo porto e as memórias VRAM permitiram obter resultados francamente positivos relativamente a, pelo menos duas figuras de mérito na apreciação destas arquitecturas: a obtenção de resoluções elevadas sobre o monitor, e a possibilidade de interacção em "tempo real".

1.5.2 - A serialização

O segundo ponto crítico na arquitectura de sistemas gráficos diz respeito ao dispositivo de serialização dos dados em memória, e de conversão d/a, as duas funções fundamentais no andar de ataque ao monitor.
Aqui colocam-se problemas delicados de velocidade sempre que se pensa num sistema gráfico de média ou alta resolução, que assentam na dicotómica necessidade de aumentar o comprimento da palavra acedida em memória, para aumentar a largura de banda total, a necessidade de elevar a taxa de acessos à memória, para aumentar o desempenho total, e a necessidade de serializar os bits em cada palavra, de os processar e de efectuar uma conversão digital-analógica.
Classicamente, a resolução para estes problemas passava pela conversão do tipo de lógica, de TTL a ECL. Mas a utilização

de ECL em implementação discreta põe, só por si, problemas graves não só ao nível do interface e lógica de controlo dos componentes e das necessidades em termos de fonte de alimentação, mas também, e desde logo, ao nível do traçado dos circuitos na própria placa. É pois uma solução cara e difícil, embora por vezes utilizada.

Nesta área o facto saliente foi o aparecimento de dispositivos em CMOS capazes de funcionar acima dos 100 MHz, o que se traduz na possibilidade, ao menos teórica, de aceder as altas resoluções (1280x1024) com relativa facilidade.
Adicionalmente, os dispositivos de mais elevado desempenho, tal como o Bt451 da Booktree, oferecem um conjunto de funções integradas capaz de cobrir as necessidades do andar de ataque do monitor: multiplexagem, paletas de cor externamente acessíveis e conversores d/a de boas características e impedância de saída normalizada. Fica assim diminuída a complexidade do interface com lógica rápida, nomeadamente ECL, que se reduz geralmente ao oscilador e ao divisor de frequência para este oscilador, normalmente com a possibilidade adicional de ligação à tensão de alimentação do sistema (5 Volts), dispensando a utilização de uma fonte de alimentação adicional de -5,2 V.

1.5.3 - O monitor

O último ponto sobre o qual focaremos a nossa atenção diz respeito ao monitor do sistema gráfico.
Uma avaliação do que nos é oferecido mostra-nos que a generalidade dos sistemas gráficos utilizam monitores com

tubos de raios catódicos de varrimento, a cores ou preto e branco.
No caso dos monitores de cor a entrada é, em geral, do tipo analógico, o que permite obter, com facilidade, paletas de cor dispondo de um número razoável de valores.

A caracterização do monitor, do ponto de vista da utilização do sistema gráfico, pode ser efectuada usando dois parâmetros importantes, o tipo de persistência e o diâmetro de perfuração da máscara do ecrã.
O tipo de persistência está relacionado com a diminuição do efeito de cintilação: dispõe-se, actualmente, de fósforos de alta persistência, permitindo a obtenção de uma imagem com elevada estabilidade e conforto de visualização.
Quanto ao diâmetro da máscara de ecrã, os valores mínimos actualmente atingidos são da ordem dos 0,21mm, e embora já se tenha atingido, experimentalmente, 0,15mm, não parece que a médio prazo tal produto seja comercializável a preço acessível.
Estas restrições tecnológicas limitam a resolução destes tubos a cerca de 1280x1280 pontos. Resoluções mais elevadas, tipicamente da ordem de 4096x4096 pontos, são possíveis sobre tubos de raios catódicos de memória.
Se bem que de custo mais elevado, estes monitores permitem uma qualidade de imagem muito superior, isenta de cintilação com maior definição, embora com menor grau de interactividade.
Outras tecnologias estão também em desenvolvimento, nomeadamente na área dos ecrãs planos a plasma e electroluminiscentes (11) os quais, no entanto, se bem que permitindo obter com certa facilidade altas resoluções -o visor plano Touchpen de Sun-Flex permite uma resolução de 4.000x4.000 pontos (11)-, não têm ainda grande flexibilidade no que respeita à gama de cores que permitem formar, pelo que o seu uso parece estar reservado, no futuro imediato, a uma classe restrita de aplicações.

1.6 - O panorama geral

Tem algum interesse avaliar quais as características correntemente oferecidas pelos fabricantes de terminais gráficos, nomeadamente no que diz respeito à capacidade de cor, resolução, emulação, facilidades e custo.
Usando para tal uma amostra de 90 produtos de 57 fabricantes dos Estados Unidos (2) verificamos que, no domínio da cor, se oferece habitualmente uma gama de 16 ou 256 cores numa paleta de até 16,7 milhões, as resoluções mais frequentes sendo as de 640x480, 1024x780 ou 1024x1024 e 1280x1024.
As emulações normalmente oferecidas são as fixadas pela Tektronix da série 4000, que pode ser considerada quase como um padrão de facto, e a DEC VT100.
Em geral são oferecidas possibilidades de "zoom", "pan", enchimento de polígonos, geração de janelas, traçado de vectores e arcos.
Quanto à gama de preços, difícil de avaliar pela sua grande variação, parece situar-se na zona dos US$3.000 a US$9.000 para as resoluções de 640x480, em cor, US$20.000 a 50.000 para a gama dos 1280x1024, também em cor, e cerca de US$30.000 para a resolução de 4096x4096, em monitores monocromáticos de memória.

1.7 - O objecto do trabalho

O objecto deste trabalho é o estudo de arquitecturas de um terminal gráfico. Pretende-se avaliar o desempenho de arquitecturas baseadas em dispositivos correntemente oferecidos pelos fabricantes, avaliar o estado da arte destes dispositivos e, daí, inferir a possibilidade de implementação de uma arquitectura "fácil" para um terminal gráfico de média resolução (640x480 pixels a 820x640 pixels) e uma paleta limitada de cores. Pretende-se ainda traçar pistas para futuros desenvolvimentos de arquitecturas na área de terminais gráficos em cor de alta resolução (1280x1280 pixels ou mais).

Neste primeiro capítulo esboçamos o contexto geral em que este trabalho se insere, apontando os vectores de desenvolvimento que entendemos serem os principais.

O segundo capítulo do trabalho faz-nos o traço da arquitectura genérica dos terminais gráficos e justifica, de algum modo, as tecnologias utilizadas, quer no que respeita aos processadores gráficos quer no que respeita ao monitor.

O terceiro capítulo descreve duas implementações particulares da arquitectura de terminais gráficos, usando dispositivos sobre os quais já se dispunha de informação detalhada, analisa os resultados obtidos, nomeadamente as limitações encontradas, e tenta descrever, à luz da experiência havida, as características desejáveis para os dispositivos a usar neste tipo de implementação.

O quarto capítulo descreve, com base em informações dos fabricantes, os dispositivos mais avançados recentemente oferecidos no mercado, cotejando-os com as conclusões

obtidas no capítulo anterior.

O quinto capítulo sugere possibilidades de desenvovimento de arquitecturas de terminais gráficos de média e alta resolução, eventualmente baseadas nos dispositivos estudados no capítulo anterior.

O sexto capítulo sintetiza e tira conclusões do trabalho efectuado e lança pistas para o desenvolvimento do trabalho futuro.

CAPÍTULO 2: A arquitectura dos sistemas gráficos

2.1 - Introdução

No capítulo anterior referimos os motivos da escolha de uma dada arquitectura de sistemas gráficos. Esses motivos são, basicamente, dois: (1) trata-se de uma arquitectura que é implementável utilizando dispositivos que, actualmente, são acessíveis com facilidade; (2) nela se centra o vector dominante da investigação e desenvolvimento internacional nesta área.
Neste capítulo começamos por descrever um modelo desta arquitectura, que servirá como ponto de referência para os desenvolvimentos posteriores, após o que detalharemos a sua implementação, referindo os conceitos básicos em que se suporta, a natureza dos dispositivos utilizados e as questões fundamentais que importa resolver.

2.2 - Um modelo de arquitectura de sistemas gráficos

Podemos definir um sistema gráfico como um conjunto de dispositivos capaz de, a partir de uma descrição apropriada de um objecto, gerar e/ou manipular uma imagem desse objecto.
Em termos de desempenho, um sistema gráfico deve oferecer três requisitos fundamentais: permitir um elevado grau de interactividade com o utilizador, ser rápido e fornecer imagens realistas, congregando aqui as necessidades de resolução, estabilidade, cor ou gradação de tons.

2.2.1 - O modelo geral

A arquitectura básica de um sistema gráfico pode ser esquematizada (13) tal como vemos na fig. 2.1

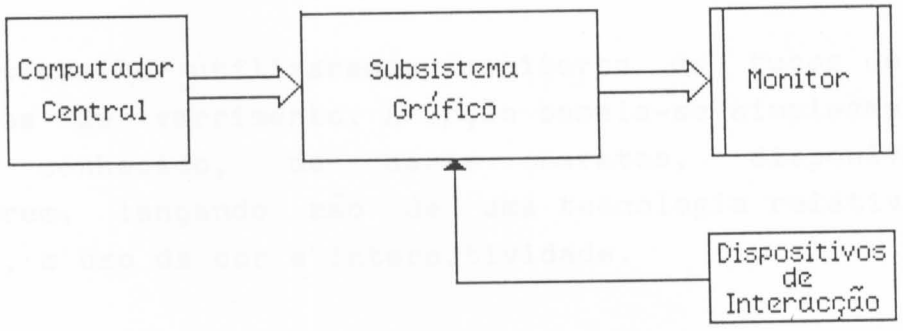


Figura 2.1

Neste esquema cabe ao computador central organizar uma representação adequada do objecto; o processador gráfico utiliza esta representação para obter outra, com organização adequada à visualização no monitor; o controlador de vídeo é o responsável pela geração da imagem, cuja sede é o monitor; os dispositivos de interacção permitem ao utilizador manipular a descrição da imagem.

Esta manipulação comporta, em geral, três classes de actividades:

- Localização: permite indicar uma posição sobre o monitor. Dispositivos de localização são, por exemplo, o rato, a caneta luminosa ou o "joystick".

- Selecção: permite seleccionar uma entidade visualizada no monitor. Um exemplo de dispositivo de selecção é a caneta luminosa ou o "touch-panel".

- Leitura: permite fornecer valores utilizáveis no processamento. Dispositivos de leitura, no sentido aqui referido, são o teclado, para a leitura de texto, comandos ou parâmetros, dispositivos potenciométricos para a marcação de valores, etc.

Neste trabalho utilizaremos monitores de tubos de raios catódicos de varrimento. A opção baseia-se simplesmente no facto, conhecido, de serem baratos, disponíveis e permitirem, lançando mão de uma tecnologia relativamente simples, o uso da cor e interactividade.

2.2.2 - O modelo físico

A utilização do monitor de raios catódicos fixa a arquitectura do sistema, nomeadamente a do processador gráfico; na figura 2.2 vemos uma representação da arquitectura básica.
Nesta concepção podemos distinguir quatro zonas:

- o processador de aplicações, geralmente um computador potente, responsável pela criação e manutenção de uma representação de alto nível da cena;

- o processador gráfico, que tem por função converter aquela representação noutra, de formato adequado à tecnologia de varrimento;

- o controlador de vídeo, cuja função é manter adequadamente a imagem no monitor;

- o monitor de varrimento.

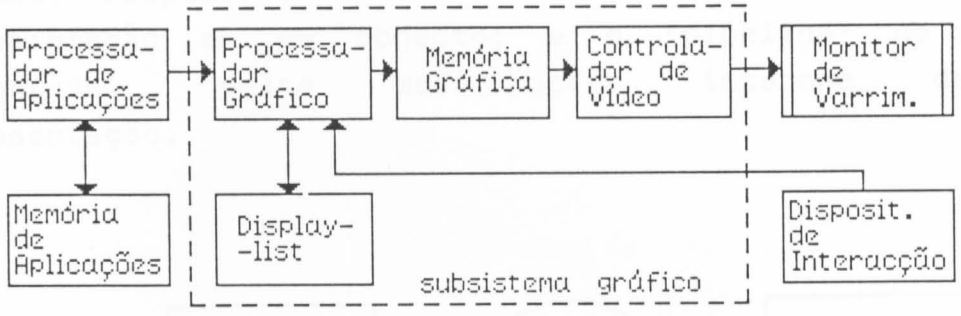


Figura 2.2

Este modelo de arquitectura básica, que não é único, mostra uma concepção funcional dita em "pipeline". Nesta concepção, funções diferenciadas, funcionando separadamente, são interligadas em série, sendo apenas necessário promover a compatibilidade na zona de interface entre cada secção do "pipeline". No modelo da fig. 2.2 podemos distinguir, a nível do subsistema gráfico, duas secções: a do processamento gráfico, que cria uma descrição da imagem na memória gráfica, e a do controlador de vídeo, que usa a descrição na memória gráfica para criar a imagem no monitor.

2.2.3 - Os processos e estruturas de dados

A este modelo de arquitectura básica, organizada funcionalmente em "pipeline", corresponde um modelo funcional de software, fig. 2.3, criado por Ingrid Carlbom (7, citado em 13), também organizado em "pipeline", no qual se distinguem essencialmente duas secções: o "pipeline" de entrada, responsável pela construção ou modificação da representação de um objecto; e o "pipeline" de saída, responsável pelas modificações internas daquela representação.

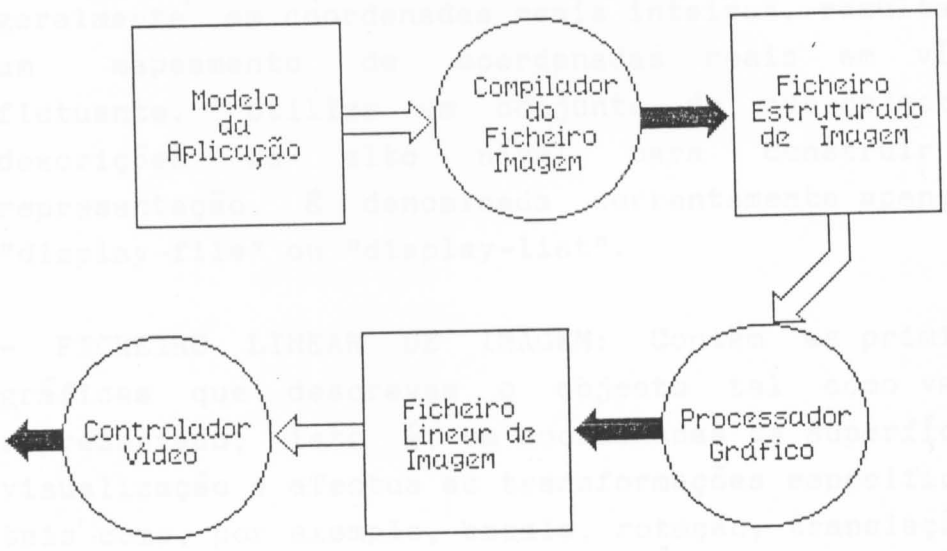


Figura 2.3

Neste modelo de software podemos distinguir, como habitualmente, os elementos de um sistema de processamento de informação: os processos, representados por círculos, que actuam sobre estruturas de dados, representadas por

rectângulos, criando outras estruturas de dados.

As três estruturas de dados são as seguintes:

- MODELO DA APLICAÇÃO: Contém uma descrição das propriedades gráficas e não gráficas do objecto. É essencialmente uma base de dados contendo a descrição do objecto, uma estrutura acessível e manejável pelo utilizador da aplicação, por exemplo, a sua descrição gráfica, numeração e descrição de componentes do objecto, propriedades dos materiais, e outras características especiais relacionadas com a aplicação.

- FICHEIRO ESTRUTURADO DE IMAGEM: Contém uma descrição hierárquica da representação gráfica do objecto, geralmente em coordenadas reais inteiras, resultado de um mapeamento de coordenadas reais em vírgula flutuante. Utiliza um conjunto de sub-objectos e descrições de alto nível para construir esta representação. É denominada correntemente apenas por "display-file" ou "display-list".

- FICHEIRO LINEAR DE IMAGEM: Contém as primitivas gráficas que descrevem o objecto tal como vai ser representado, isto é, em coordenadas da superfície de visualização e efectua as transformações especificadas, tais como, por exemplo, escala, rotação, translação, etc.

Os três processos gráficos representados podem ser definidos pela sua função. Assim:

- o COMPILADOR DO FICHEIRO DE IMAGEM, cuja entrada é o modelo da aplicação, varre sequencialmente esta estrutura e constrói outra contendo, hierarquicamente organizados, os elementos a visualizar.

- o PROCESSADOR GRÁFICO mapeia o ficheiro estruturado de imagem (display file) no ficheiro linear, o qual é, na tecnologia que estamos a tratar, uma "cópia" ponto a ponto da imagem a visualizar.

- o CONTROLADOR DE VÍDEO, que mapeia o ficheiro linear de imagem, concatenado com os respectivos atributos, na superfície de visualização, construindo e mantendo a imagem propriamente dita.

A estes três processadores lógicos não têm necessariamente de corresponder três processadores físicos distintos, podendo um determinado dispositivo executar funções adstritas a mais do que um processador lógico, ou a parte de um processador lógico. Em geral, ou melhor, a tendência actual não é essa, ao menos no que respeita ao processador gráfico e ao controlador da vídeo. De facto, o primeiro, sendo responsável pela manutenção da representação na memória gráfica, pode funcionar a velocidade diferente, menor, do que a do controlador de vídeo, que tem necessidades muito exigentes de refrescamento da imagem no monitor, sob pena de perturbação, sinónimo de má qualidade do sistema gráfico.
Eventualmente, poderão estar reunidas as funções do processador da aplicação e do processador gráfico, se bem que habitualmente a primeira função seja desempenhada por um computador central, em geral potente, e a segunda seja objecto, actualmente, de processadores VLSI dedicados.

2.3 - Conceitos básicos relativos aos subsistemas gráficos

Focaremos a nossa atenção na arquitectura de subsistemas gráficos usando monitores de varrimento, munidos de interfaces com outros dispositivos (computador central, estação gráfica, dispositivos de leitura, etc.). Representa-se na figura 2.4 um modelo geral deste subsistema que pode ser utilizado como modelo de um terminal gráfico com capacidade de cor. Outro modelo possível teria todos os dispositivos com acesso separado ao barramento geral do sistema, mas estamos a focar a nossa atenção numa arquitectura envolvendo "pipeline", por ser essa a tendência geral, e aparentemente irreversível, do desenvolvimento tecnológico actual.

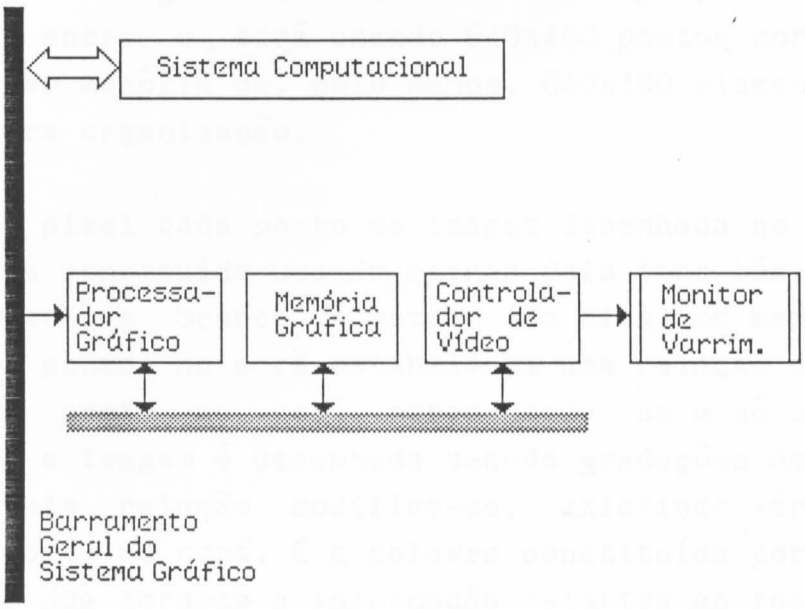


Figura 2.4

Neste modelo o processador gráfico, usando comandos externos, escreve e lê na memória gráfica. Este processador

deve poder calcular e escrever com eficiência vários tipos de figuras, nomeadamente vectores, arcos, polígonos, faces pintadas e símbolos, incluindo caracteres, eventualmente realizando operações lógicas sobre o conteúdo da memória. Deve ainda ser capaz de copiar uma área de memória para outra zona da memória.
O processador gráfico pode ser construído com base num processador de uso geral, ou pode ser um processador dedicado, discreto ou VLSI.

A memória gráfica, muitas vezes denominada, nesta arquitectura, de "bit map", é em geral constituída por um ou mais bancos de memória. O seu conteúdo é uma matriz de valores lógicos discretos, que é uma representação ponto a ponto da imagem a desenhar no ecrã, e cuja dimensão fixa a resolução da imagem desenhada. Por exemplo, a uma figura construída sobre o ecrã usando 640x480 pontos corresponde uma matriz em memória de, pelo menos, 640x480 elementos, com esta ou outra organização.

Denomina-se pixel cada ponto da imagem desenhada no ecrã. Se a imagem é construída usando apenas dois tons básicos, por exemplo preto e branco, a matriz (de bits) em memória e a matriz de pontos no ecrã estabelecem uma relação de 1 para 1: a cada ponto no ecrã corresponde um e só um bit em memória. Se a imagem é desenhada usando gradações de tons ou cores, aquela relação modifica-se, existindo vários bits para cada ponto no ecrã. É a palavra constituída por aqueles vários bits que fornece a informação relativa ao tom ou cor. Esta palavra é designada, geralmente, por profundidade do pixel.

Quando existem vários bits por pixel podemos ver a memória como um conjunto de bancos, cada banco estabelecendo uma relação de 1 para 1 com o conjunto de pixels da imagem, e o conjunto de bancos estabelecendo uma relação de n para 1 com

o conjunto de pixels da imagem.
Designa-se por "bit plane" cada um dos bancos de memória referidos. O número de "bit planes" fixa o número de cores utilizáveis simultaneamente. Assim, quatro "bit planes" permitem o uso de 16 cores simultâneas. Um sistema gráfico usando 24 "bit planes" diz-se ser um sistema de cor total.

A tecnologia vulgarmente utilizada para a implementação da memória gráfica é a das RAM dinâmicas, em MOS. Esta escolha faz-se, essencialmente, por razões de custo. Só em sistemas onde se deseja um muito elevado desempenho são utilizadas memórias RAM estáticas. O uso de DRAMs põe, no entanto, o problema da necessidade de refrescamento, dado que não se pode garantir que, no tempo disponível, todas as células de memória sejam acedidas pelo processador gráfico ou pelo controlador de vídeo. Em geral, implementa-se o refrescamento da memória de modo transparente para o utilizador e tendo o menor impacto possível no desempenho global. Note-se que a memória tem de ser acedida quer pelo processador gráfico, quer pelo controlador de vídeo, quer pelos dispositivos de refrescamento da DRAM. Estes acessos têm de ser, de algum modo, arbitrados. Por isso é corrente que o processador gráfico, nomeadamente quando é um processador dedicado, contenha facilidades específicas para efectuar o refrescamento. Alternativamente, deve usar-se um controlador de refrescamento e implementar um circuito de arbitragem do acesso à memória que garanta não existir perda de informação. Dado que os sistemas gráficos (e os sistemas computacionais em geral) devem, desejavelmente, possuir larguras de banda tão elevadas quanto possível, as DRAMs actuais, normalmente organizadas ao bit, oferecem outras organizações de modo a permitirem acessos sequenciais mais eficientes. Estas organizações modificadas são referidas como modos de funcionamento à página (page mode), ao "nible" (nible mode) e em "ripple mode"

A função do controlador de vídeo é ler a informação contida na memória gráfica e fornecer os sinais necessários para formar a imagem no monitor, com formato e temporização adequados.
Em geral é também função deste controlador a execução de algumas actividades do sistema, tais como "pan", "scroll" e janelas, geração e controlo de cursores, execução de sobreposições e transparências, etc.
O controlador de vídeo efectua a leitura sequencial de palavras na memória gráfica. O comprimento da palavra lida é definido pela largura do barramento do subsistema gráfico, sendo em geral de 16 ou mais bits. A primeira acção do controlador é converter esta entrada paralela numa saída série com temporização adequada. Esta saída série é depois processada até à conversão no sinal vídeo. Este processamento consiste, em geral, na definição do código de cor que vai afectar o pixel e que serve de índice à denominada tabela de cores (look-up-table), cuja saída é condicionada por outra informação, (cursores, símbolos, transparências), e na conversão digital-analógica, incluindo dispositivos de adaptação às impedâncias normalizadas de saída.
Normalmente, este controlador é implementado com componentes discretos, embora a tendência seja a de utilizar circuitos dedicados VLSI.

O último componente do subsistema gráfico, na arquitectura que estamos a descrever, é o monitor. A tecnologia utilizada, pelos motivos antes apontados, é o tubo de raios catódicos de varrimento. Neste tipo de TRC, o feixe electrónico executa permanentemente um varrimento sequencial do ecrã, traçando sucessivas linhas horizontais, segundo um movimento linear da esquerda para a direita, com um deslocamento vertical entre linhas. Esta figura é denominada "raster" e cada uma das linhas da figura é denominada "raster line" ou "scan line".

Algumas grandezas podem ser definidas sobre esta figura:

- retorno: no fim de cada linha o feixe electrónico tem de "voltar para trás", ao início da próxima linha; no fim do varrimento do quadro tem de se deslocar, também "para trás" desde o canto inferior direito até ao canto superior esquerdo. A estes movimentos "para trás" dá-se o nome de retorno, respectivamente, horizontal e vertical. O tempo gasto no retorno é denominado, em consequência, tempo de retorno horizontal e vertical. Durante o retorno o feixe é apagado (blanked); durante o varrimento o feixe diz-se activo.

- taxa de refrescamento do ecrã: é o número de imagens escritas no ecrã na unidade de tempo. A duração de cada imagem é denominada "frame time" e é o inverso da taxa de refrescamento.

- entrelaçamento: existem essencialmente dois tipos de imagens.
O primeiro corresponde ao varrimento sequencial de todas as linhas do quadro; o segundo, originário das técnicas de televisão, faz o traçado das linhas ímpares (quadro ímpar) e, no fim, intercala o traçado das linhas pares (quadro par). O primeiro sistema é dito não-entrelaçado e o segundo entrelaçado.
Por exemplo, a TV europeia usa 25 imagens/s e 50 quadros/s (entrelaçado); um terminal de computador pode usar 50 imagens/s e 50 quadros/s (não entrelaçado).

- cintilação: é o fenómeno observado na imagem quando a taxa de refrescamento do ecrã não é suficiente para uma observação permanente apoiada na persistência da excitação da retina. Em geral estima-se que, em sistemas não-entrelaçados, esta taxa, para não existir cintilação apreciável, deve ser superior a 50

imagens/segundo, sendo menor para sistemas entrelaçados.

- sinais vídeo normalizados: a configuração do sinal vídeo a entregar ao TRC de varrimento é objecto de, pelo menos, duas normas, ambas estabelecidas para televisão. Não estando ainda definida uma norma equivalente para sistemas gráficos, utiliza-se em geral uma daquelas.

A norma RS-170-A da Electronic Industries Association define o sinal para TV a preto-e-branco ou cor, tornando-a aplicável a sistemas de 525 linhas e 30 imagens/segundo, entrelaçadas. A norma RS-343-A aplica-se a sistemas de TV de alta resolução, tais como 1024 linhas e 30 quadros/segundo, não-entrelaçadas. Esta norma, que é geralmente a utilizada, especifica para o tempo de retorno vertical 1250 useg+/-150 useg e para o tempo de retorno horizontal 7 useg+/-0,20 useg.

- frequência de pixel: é a frequência máxima, fixada pelas características dos componentes do sistema, à qual se podem gerar pontos sobre o ecrã. O seu inverso é denominado tempo de pixel. É um parâmetro importante no projecto de sistemas gráficos.

As definições que referimos permitem estabelecer uma fórmula para o cálculo do tempo de pixel para um dado monitor

Assim, convencionando:

Tq	tempo total de um quadro
Tvr	tempo total do retorno vertical
Thr	tempo total do retorno horizontal
NL	número de linhas visíveis de um quadro
taq	tempo activo de um quadro

tvr tempo de retorno vertical
tl tempo de uma linha
NP número de pixels visíveis numa linha
tp tempo de pixel
R taxa de refrescamento

teremos:

tempo total de um quadro Tq=taq+Tvr+Thr

mas Tvr=tvr
Thr=NL*thr
taq=NL*tl
e ainda tl=NP*tp

logo Tq=NL*NP*tp+tvr+NL*thr

donde tp=1/NL*NP(Tq-tvr-NL*thr)

atendendo a que R=1/Tq, teremos

tp=(((1/R-tvr)/NL)-thr)/NP

Por curiosidade, calculemos o tempo de pixel para um sistema com resolução de 1280x1024, 50Hz, não-entrelaçado, RS-343-A.

Teremos:

tp<=(((1/50-(1250+150)E-6)/1024)-(7+0,25)E-6)/1280

obtendo tp<=8,5 ns.

O inverso de tp seria a frequência máxima a que deveria funcionar o controlador de vídeo, para obter o máximo desempenho, supondo o monitor tecnologicamente adequado a estes valores.

2.4 - Duas questões fundamentais

No modelo de arquitectura de sistemas gráficos a que nos estamos a referir podem ser identificadas duas questões básicas, que detalharemos posteriormente. A primeira relaciona-se com o balanço dos vários componentes do sistema e a segunda com a utilização do barramento da memória.

2.4.1 - O balanço hardware-firmware-software

Neste modelo está bem patente o grau de compatibilidade e inter-relação entre o sistema de hardware e o sistema de software. É no modo como esta inter-relação é projectada que se decide, geralmente, o desempenho global do sistema.
Um bom exemplo desta afirmação está referido em (5), onde se comparam as arquitecturas dos terminais 4113 e 4115 da Tektronix.
A diferença fundamental nos projectos destes modelos refere-se à passagem de uma arquitectura em que estavam reunidas as funções do processador gráfico, neste caso baseado no processador 8086, suportando uma alta carga de processamento, para outra arquitectura utilizando um processador "bit slice" para a execução de algumas das funções antes desempenhadas pelo 8086, nomeadamente as que respeitam ao varrimento da "display-list" e à execução dos comandos de transformação de imagem. Os resultados, em termos de aumento de desempenho, foram notáveis: cerca de 50 vezes na velocidade de escrita de pequenos vectores e de 1000 vezes na velocidade de escrita de faces.

2.4.2 - A largura de banda do barramento de memória

A questão fundamental posta por este tipo de arquitectura diz respeito à saturação dos acessos à memória gráfica. De facto, esta memória tem de ser acedida para execução de três funções diferentes, qualquer delas especificando limites apertados para esse acesso. Por um lado, o controlador gráfico tem de, continuamente, varrer a memória gráfica, lendo o seu conteúdo para o monitor; uma paragem neste acesso significa a existência de perturbação, em maior ou menor grau, na visualização da imagem.
Segundo, o processador gráfico tem de aceder, sem qualquer restrição, à memória gráfica, para poder efectuar, rapidamente, actualizações do seu conteúdo; a imposição de restrições a este nível implica uma menor interactividade. Por último, é necessário efectuar o refrescamento da memória gráfica (supondo sempre que, como é usual, é implementada em DRAM), dentro dos limites impostos pela sua tecnologia; restrições aqui implicam a possibilidade de perda do conteúdo da memória gráfica.

Como estas três classes de acesso não podem ser simultâneas, é necessário estabelecer um protocolo de acesso à memória gráfica que garanta o correcto funcionamento do sistema. Uma solução simples é estabelecer uma classe de prioridades que, como não podia deixar de ser, atribui maior prioridade aos acessos efectuados pelo controlador de vídeo do que ao processador gráfico, para assim poder dispor de uma imagem de qualidade no ecrã. Deste modo, o processador gráfico vê os seus acessos preteridos durante intervalos de tempo significativos. Esta questão torna-se mais grave à medida que a resolução ou o número de cores pretendidos aumentam, aumentando em consequência a dimensão da memória gráfica.

Este problema foi em parte resolvido com a utilização de vídeo-RAMs mas, por agora, centremos a nossa atenção no uso de DRAMs, dado que foi essa a tecnologia usada na implementação que descreveremos no capítulo seguinte.

A técnica mais simples para prevenir a possibilidade de acesso simultâneo à memória consiste em permitir que o processador gráfico lhe tenha acesso apenas durante os tempos de retorno do feixe que não sejam utilizados para efectuar o refrescamento da memória, normalmente durante o tempo de retorno vertical.
Obviamente que este tipo de arbitragem pode limitar severamente a velocidade de actualização da imagem, ou, o que é o mesmo, diminuir drasticamente o grau de interactividade. Righter (citado em 23), menciona algumas técnicas para reduzir os inconvenientes da solução, a mais vulgarizada parecendo ser a adição de uma FIFO ao sistema. O processador gráfico, que é o único dispositivo podendo aceder a este "buffer", vai-o carregando com os resultados da computação, organizando o "buffer" de modo semelhante à organização da memória gráfica. Quando se torna possível o acesso do processador à memória gráfica, é feita uma descarga deste "buffer", previamente preparado, usando um dos processos de aumento da largura de banda do barramento mencionados anteriormente: efectua-se a transferência ao "niblle", à página ou em modo "ripple". Esta técnica de "bufferização" do processador gráfico, conquanto permita um melhor desempenho quando a actividade do processamento é o cálculo e carga da memória gráfica, não melhora esse desempenho sempre que sejam necessárias actividades que impliquem a leitura, modificação e re-escrita da memória.

Outra técnica utilizada para prevenir a possibilidade de acesso simultâneo à memória é a duplicação da memória gráfica. É no entanto uma solução cara e, como veremos, não completamente satisfatória.

Neste caso, um dos "buffers" de memória está sempre acessível ao processador gráfico para leitura e escrita e o segundo "buffer" está acessível ao controlador de vídeo para leitura. Quando termina um varrimento total deste "buffer", ambos podem trocar de função.

A utilização de um duplo "buffer" resolve alguns problemas da técnica, já referida, de escrita durante o retorno vertical, principalmente o que se relaciona com a transição suave entre imagens. Neste caso, quando o cálculo e escrita de uma nova imagem não pode ser feita durante o tempo de retorno vertical, a transição entre imagens sucessivas não tem um aspecto realista. É o caso de aplicações onde um objecto se move no ecrã, caso em que é praticamente obrigatório o uso de "buffer" duplo.

Existem duas organizações possíveis quando se implementa a memória gráfica usando o duplo "buffer". Na primeira, que chamaremos "bufferização em profundidade", a memória é organizada com o dobro dos "bit planes" necessários para a gama de cores desejada, formando dois bancos diferenciados. Estes dois bancos podem ser acessíveis por endereçamento comum ou separado. No caso de ser comum é necessária lógica adicional de protecção do acesso. Esta organização contém, intrínseca, a vantagem de facilmente se poder dispor de dois tipos de organização: a memória gráfica pode ser tratada como um duplo "buffer" para uma gama restrita de cores, em sistemas dispondo de bom desempenho e transição suave de imagem, ou pode ser tratada como um "buffer" simples, com menor desempenho mas uma larga gama de cores. No caso usual do processador gráfico trabalhar sobre um barramento de 16 bits consegue-se, por esta via, com facilidade, um sistema de cor total.

Noutra organização possível, que denominamos "bufferização em superfície", o número de "bit planes" implementados

corresponde à profundidade de pixel desejada, sendo a memória total dividida num número múltiplo da memória gráfica necessária.
Estes bancos de memória assim definidos podem ter endereçamento adjacente ou separado. No primeiro caso a escolha de um dado banco para acesso do controlador de vídeo faz-se pela técnica de janela, sendo possível, com alguma lógica adicional, uma acção de "pan" e "scroll"sobre a memória total; no caso de endereçamento separado estas facilidades ficam reduzidas, mas a lógica adicional é menos complexa.

CAPÍTULO 3: Descrição e análise de duas implementações de uma arquitectura de terminais gráficos.

3.1 - Introdução

Uma primeira questão que se coloca ao projectar uma implementação do modelo de subsistema gráfico que referimos em 2.2, é a da possibilidade de escrita, combinada ou não, de texto e desenho.
De facto, as necessidades para estas funções são bastante diferenciadas e o seu tratamento uniforme cria alguns problemas, sobretudo se desejarmos dispor do desempenho habitualmente oferecido em cada um deles.
Normalmente o utilizador de um sistema gráfico deseja poder obter desenho sobre o ecrã e incluir nele caracteres alfanuméricos de variados tipos, dimensão e posição. Tal consegue-se, em geral, usando o procedimento gráfico disponível no sistema. Mas o utilizador de um sistema gráfico, que nele investiu um valor apreciável, deseja certamente poder dispor de capacidade de texto, com as facilidades usuais, nomeadamente as de edição. A solução convencional é utilizar processadores separados para texto e gráficos, e arbitrar, em cada momento, a saída efectiva para o monitor. Este tipo de solução é amplamente referenciada na literatura especializada (9, 14), existindo vários sistemas comercializados que a implementam. A figura 3.1 mostra a solução preconizada.

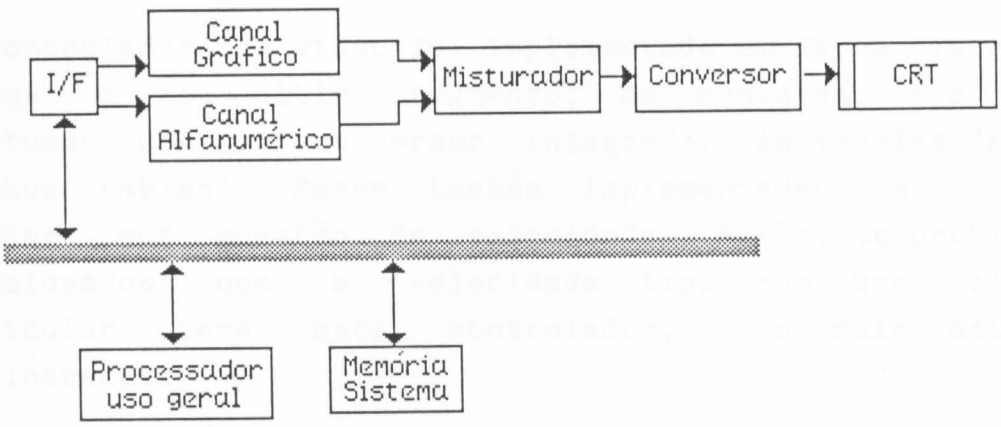


Figura 3.1

O primeiro projecto de implementação que realizámos baseia-se nesta arquitectura. O barramento geral do sistema central é um barramento específico do sistema ELENA desenvolvido no INESC (4), mas tal detalhe não é significativo de momento; o subsistema gráfico projectado podia ser desenhado com um interface comunicando com qualquer tipo de sistema central.

O controlador gráfico utilizado foi o uPD7220 da NEC (graphics display controller). Este controlador foi escolhido por ser, na data de início do trabalho, quase um padrão de facto, amplamente utilizado na indústria, não obstante actualmente poder ser considerado, de certo modo, se não obsoleto, pelo menos bastante limitado, utilizável apenas nas implementações de sistemas da classe mais baixa, facto que se deve à intensa evolução tecnológica havida durante o ano de 1986.

O subsistema de memória foi implementado com a DRAM TMS4416E, que acabava de ser comercializada no nosso país.

O controlador de vídeo foi implementado em forma discreta: apenas o seu último segmento, de conversão d/a, era efectuado com um conversor integrado. As tabelas de cor (lookup tables) foram também implementadas em memória bipolar, por questão de velocidade. Aliás, os problemas relacionados com a velocidade impuseram uma solução particular para este controlador, que mais adiante detalharemos.

O segundo projecto de implementação utiliza o mesmo modelo, embora a implementação física seja substancialmente diferente:

- O barramento geral do sistema continua a ser o mesmo, mas agora o sistema gráfico dispõe de um interface que permite ao processador central o acesso total à memória gráfica.

- O processador gráfico usado foi o processador de uso geral iAPX186, o qual é responsável quer pela manutenção da "display-list", quer pela sua execução.

- O subsistema de memória foi totalmente implementado com as VRAMs TMS4161E, embora mantendo separados o processamento gráfico e de texto.

- Foi usado um controlador de vídeo VLSI, o TMS34061.

- Os conversores d/a continham, integradas, as tabelas de cor.

- Praticamente toda a lógica de controlo foi implementada em PAL.

A figura 3.2 mostra o esquema básico da solução adoptada nestas duas implementações.

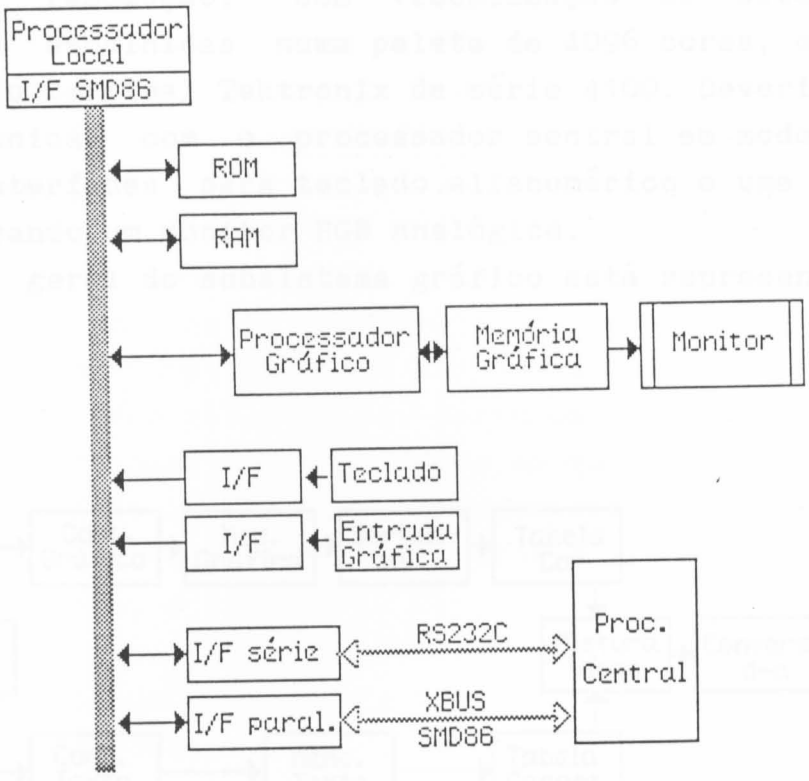


Figura 3.2

Como veremos, os elementos distintivos mais salientes entre as duas implementações são, de facto, o tipo de processador gráfico usado e o nível de integração conseguido.

3.2 - A primeira implementação do terminal gráfico.

Nesta implementação fixou-se como objectivo geral estabelecer a arquitectura e implementar um terminal gráfico de média resolução, com visualização de oito cores simultâneas escolhidas numa paleta de 4096 cores, capaz de emulação do terminal Tektronix da série 4100. Deveria ainda poder comunicar com o processador central em modo série, possuir interfaces para teclado alfanumérico e uma entrada gráfica, usando um monitor RGB analógico.
O esquema geral do subsistema gráfico está representado na figura 3.3

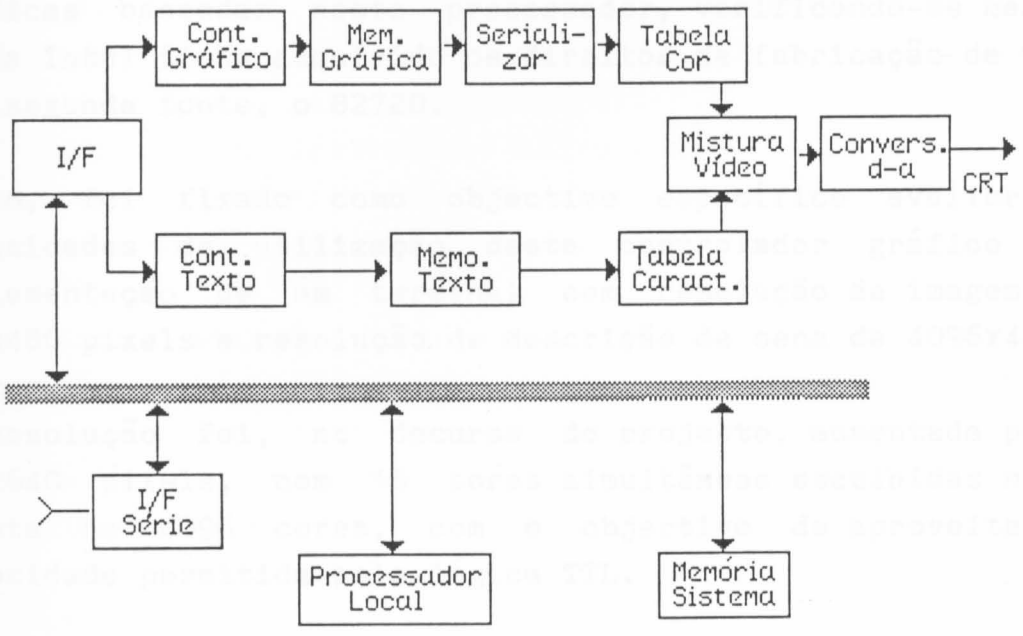


Figura 3.3

Neste esquema, o processador local era um sistema, implementado no INESC, baseado no microprocessador i8086, com 96 Kbytes de memória ROM, 32 Kbytes de memória RAM localmente extensível a 128 Kbytes, 64 Kbytes de mapa I/O e portos de entrada/saída.

O processador gráfico escolhido foi o uPG7220 da NEC, sendo o banco de memória gráfica, do tipo "bit map", implementado com as TMS4416E, e o andar de conversão com o conversor d/a VDAC444TD, da Intech.
A escolha do processador uPG7220 radica em duas ordens de razões: secundariamente, verificava-se a sua disponibilidade no mercado nacional; fundamentalmente, a análise da literatura mostrava que era um dispositivo que tinha encontrado boa receptividade por parte da indústria, havendo já, comercializadas, implementações de terminais e estações gráficas baseadas neste processador, verificando-se mesmo que a Intel tinha adquirido os direitos de fabricação de uma sua segunda fonte, o 82720.

Assim, foi fixado como objectivo específico avaliar as capacidades de utilização deste controlador gráfico na implementação de um terminal com resolução de imagem de 640x480 pixels e resolução de descrição de cena de 4096x4096 pontos.
A resolução foi, no decurso do projecto, aumentada para 820x640 pixels, com 16 cores simultâneas escolhidas numa paleta de 4096 cores, com o objectivo de aproveitar a velocidade permitida pela lógica TTL.

As soluções possíveis

Da análise das características do controlador gráfico 7220 parecia possível seguir uma de três soluções: utilizar o controlador gráfico para calcular e actualizar a memória gráfica, actualizando sucessivamente cada um dos planos de

cor do sistema; estabelecer um sistema distribuído, com um controlador actuando sobre cada plano de cor, numa arquitectura que aproveitava as facilidades de sincronização de vários controladores (escravos) a partir de um deles (mestre); ou usar um único controlador e estabelecer, separadamente, os meios capazes de implementar uma estrutura de paralelismo capaz de tratar, simultaneamente, os vários planos de cor.

A primeira solução, sendo a mais simples, parecia não conduzir a um sistema de desempenho adequado, dada a necessidade de escrever, sequencialmente, quatro palavras, uma em cada plano de cor. De facto, indicando o fabricante um tempo de escrita (obrigatoriamente com execução de um ciclo de leitura-modificação-escrita) de 800 ns, a escrita de um pixel, com esta arquitectura, necessitaria de um tempo mínimo de 3,2 us. Assim, a frequência máxima teórica de actualização do ecrã era da ordem dos 300.000 pixels/seg, devendo na prática ser bastante inferior.
Este valor parece francamente baixo quando comparado com os oferecidos por vários sistemas comercializados, razão pela qual esta arquitectura não foi utilizada.

A segunda solução, uma arquitectura de controladores distribuídos permite, logo à partida, e ainda numa avaliação grosseira, aumentar a taxa de actualização do ecrã. Tem a vantagem de ser uma solução para a qual a arquitectura do controlador gráfico aponta, pela implementação da capacidade de sincronismo entre vários controladores, existindo alguns sistemas utilizando esta arquitectura (20).
As vantagens não parecem, no entanto, ser tão óbvias: o estreitamento que existia na taxa de escrita na memória gráfica passa para a escrita nos "buffers" de entrada dos controladores. De facto, agora, é o processador central que necessita de tempo para a passagem dos comandos adequados, sequencialmente, a cada controlador, incluindo o controlo do

estado do "buffer".
Adicionalmente, o mecanismo de sincronização dos controladores escravos pelo controlador mestre não é isento de alguns problemas, impondo em geral a utilização de lógica externa (20).
Por último, a situação agrava-se substancialmente se desejarmos aumentar o número de planos de cor.
São decerto estes os motivos que levam a que as implementações usando esta arquitectura de controladores distribuídos se situem na gama das resoluções médias, tipicamente 640x480 pontos.

A terceira solução foi a adoptada no projecto que descrevemos.
Nesta concepção, o controlador gráfico não trabalha directamente sobre um plano de cor, mas apenas manipula máscaras, as quais determinam a escrita simultânea, em paralelo, dos vários componentes da profundidade do pixel.
De facto, para a escrita de um pixel, é inicialmente fornecida ao controlador gráfico uma máscara fixa de leitura, e a programação do controlador deve prescrever as modificações a efectuar a esta máscara para obter o resultado pretendido. Constitui-se assim uma máscara de escrita na memória, a qual, convenientemente tratada, gera a informação a ser gravada.

O ciclo de leitura-modificação-escrita do controlador actua pois sobre as máscaras de leitura e escrita, e estas máscaras têm uma dimensão que depende apenas da largura do barramento de acesso à memória (16 bits), e não da profundidade desejada para o pixel. Teoricamente, pelo menos, o sistema é aberto do ponto de vista da profundidade do pixel, que pode ser fixada em qualquer dimensão sem afectação do desempenho do sistema, dado que todos os componentes do pixel são tratados em paralelo, com base na informação da máscara de escrita.

Nesta arquitectura parece ser de esperar que o desempenho se situe no máximo permitido pelo controlador, como se de um sistema com um único plano (preto-e-branco) se tratasse.
A figura 3.4 mostra o esquema básico desta arquitectura.

Podemos distinguir, neste esquema, os três blocos básicos do "pipeline" do sistema gráfico: a primeira secção com o controlador gráfico, a memória gráfica e a lógica de endereçamento e controlo, incluindo os dispositivos específicos desta arquitectura, a saber, as máscaras de leitura e escrita e o dispositivo de tradução constituído por um "buffer" de leitura, lógica de controlo específico, multiplexadores de dados e um banco de registos; a segunda secção, de serialização, constituída pelos registos de deslocamento (shift registers) e a lógica associada; a terceira secção, de conversão, constituída pelos registos da tabela de cores, incluindo um "buffer" específico e pelos conversores d/a e ainda a lógica de sincronização da conversão e do monitor.

3.2.1 - O controlador gráfico uPD7220 (graphics display controller), da NEC.

Trata-se de um controlador implementado em VLSI, em tecnologia NMOS, integrando um conjunto de actividades específicas do processamento gráfico. De um modo geral essas actividades são:

- Leitura e escrita da memória gráfica: o GDC, durante a escrita de figuras, calcula o endereço de cada pixel da figura, actualizando a memória de acordo com o resultado do cálculo.

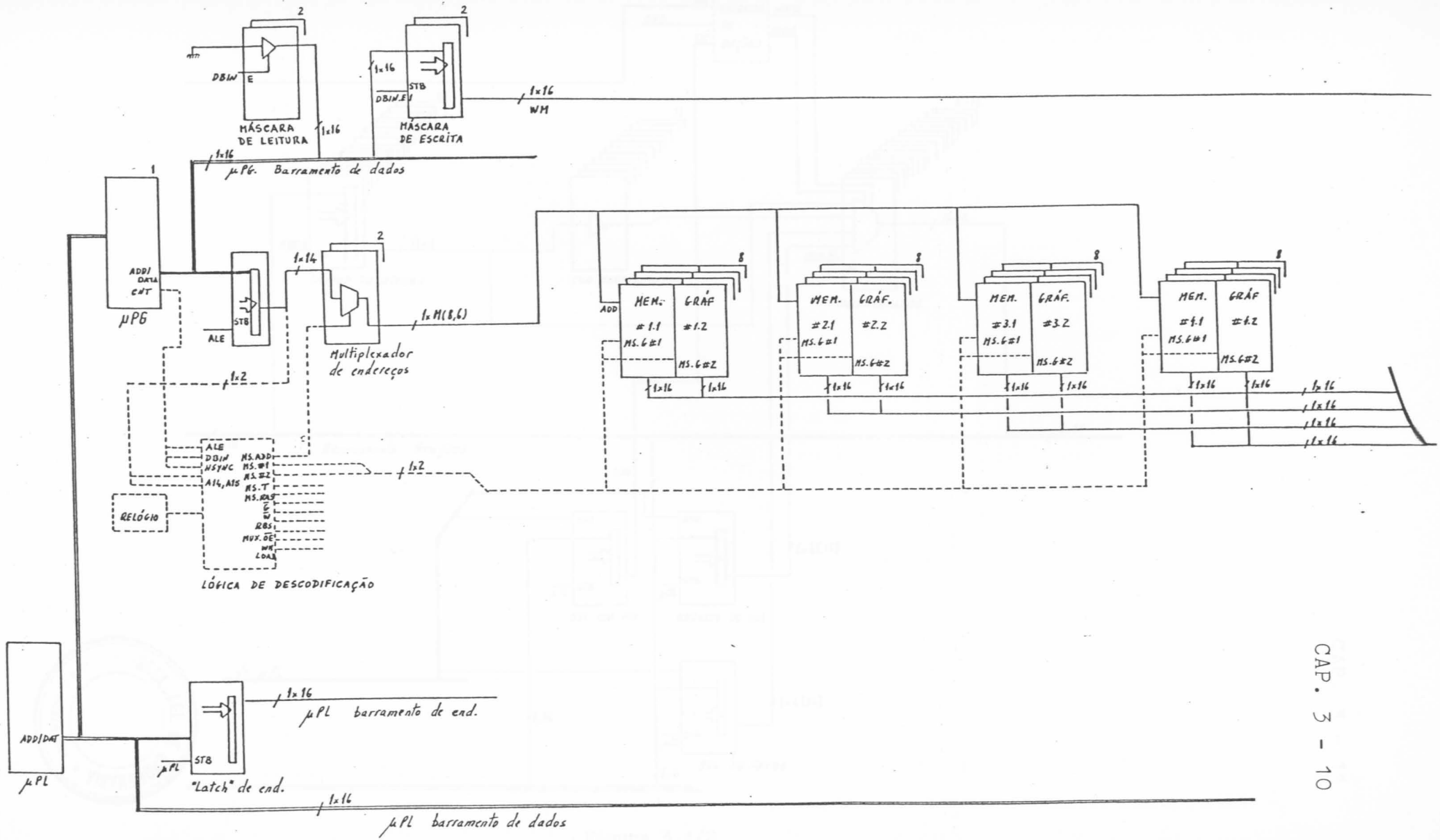

MÁSCARA DE LEITURA
MÁSCARA DE ESCRITA
WM
µPB. Barramento de dados
µPB
ALE
STB
Multiplexador de endereços
MEM.
GRÁF.
LÓGICA DE DESCODIFICAÇÃO
RELÓGIO
µPL
"Latch" de end.
µPL barramento de end.
µPL barramento de dados

Figura 3.4/1

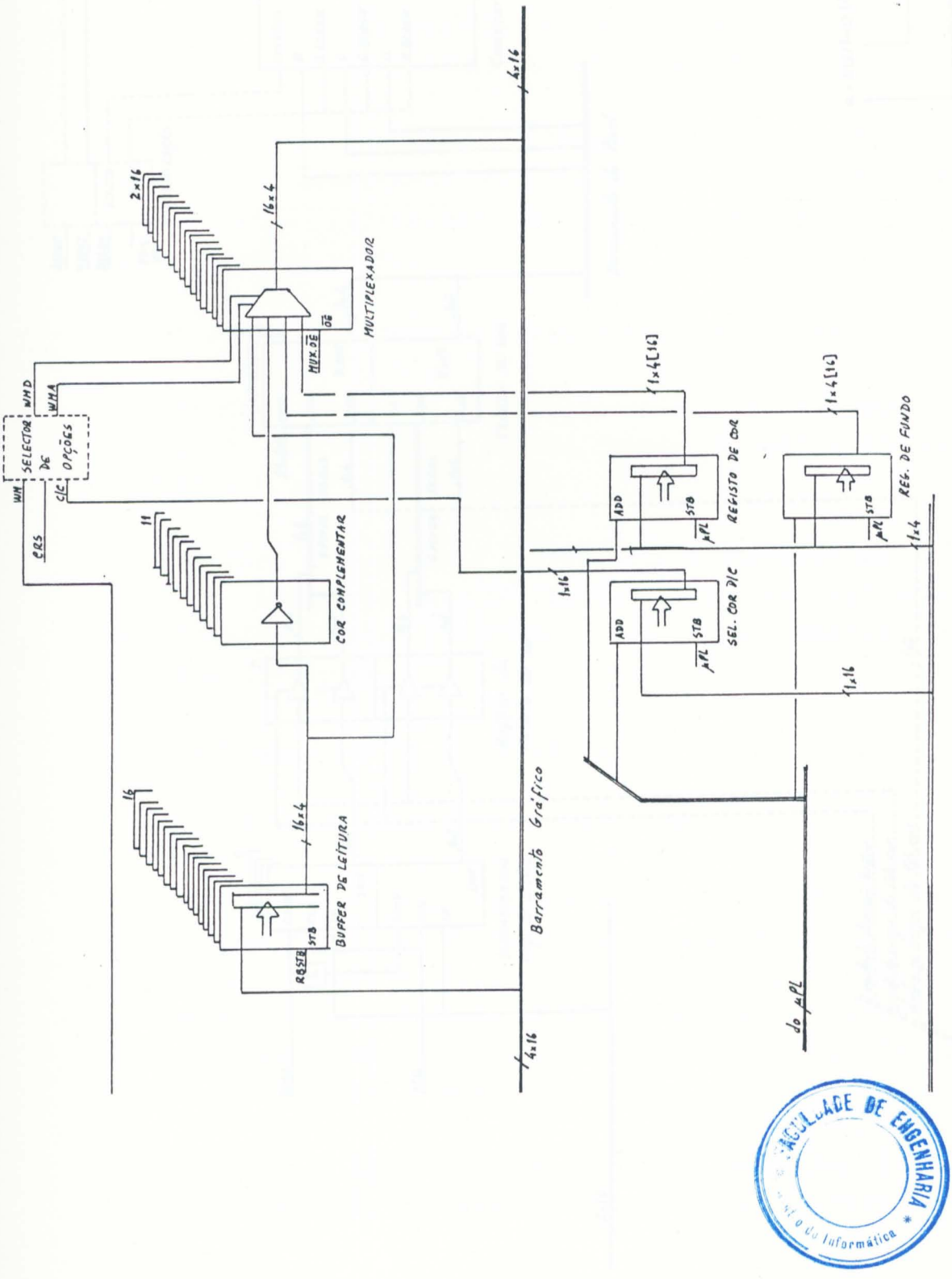


Figura 3.4/2

FACULDADE DE ENGENHARIA
Centro de Informática

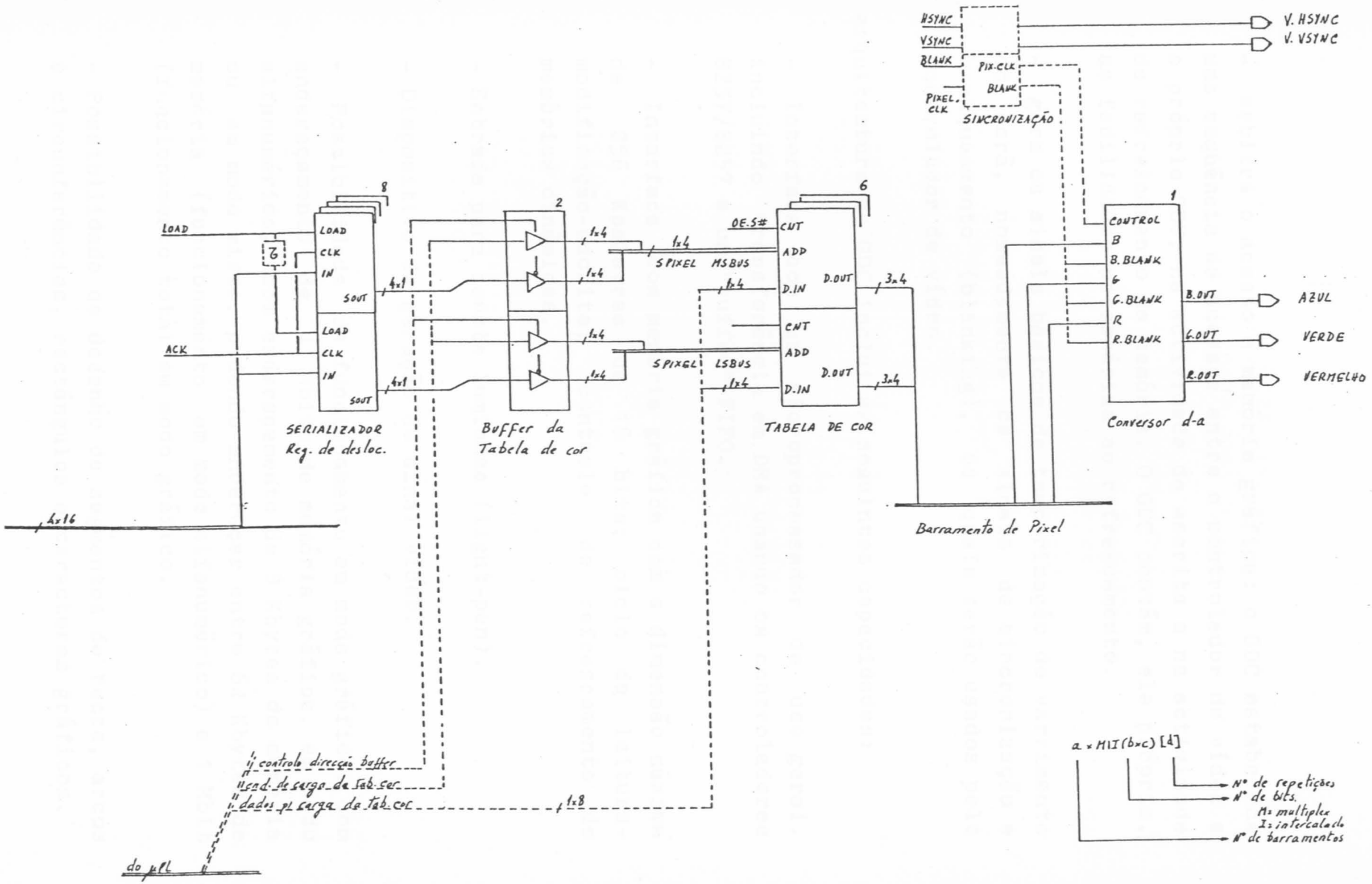

HSYNC
VSYNC
BLANK
PIXEL CLK
PIX.CLK
BLANK
SINCRONIZAÇÃO
V. HSYNC
V. VSYNC
LOAD
ACK
LOAD
CLK
IN
SOUT
SERIALIZADOR
Reg. de desloc.
Buffer da
Tabela de cor
OE.S#
CNT
ADD
D.IN
D.OUT
SPIXEL MSBUS
SPIXEL LSBUS
TABELA DE COR
CONTROL
B
B. BLANK
G
G. BLANK
R
R. BLANK
B.OUT
G.OUT
R.OUT
AZUL
VERDE
VERMELHO
Conversor d-a
Barramento de Pixel
4x16
controlo direcção buffer
cod. de carga da tab. ser.
dados p/ carga da tab. cor
do µP
a x MII (b×c) [d]
Nº de repetições
Nº de bits
M: multiplex
I: intercalado
Nº de barramentos

Figura 3.4/3

- arbitra o acesso à memória gráfica: o GDC estabelece uma sequência de acessos entre o controlador de vídeo e o próprio GDC, na actividade de escrita e na actividade de refrescamento da memória. O GDC contém, ele próprio, as facilidades necessárias ao refrescamento.

- gera os sinais básicos de temporização do varrimento do ecrã, nomeadamente os sinais de sincronização e branqueamento (blanking), os quais serão usados pelo controlador de vídeo.

A arquitectura do GDC inclui as seguintes capacidades:

- Interface com um microprocessador de uso geral, incluindo transferência em DMA usando os controladores 8237/8257 e um "buffer" FIFO.

- Interface com memória gráfica com a dimensão máxima de 256 Kpalavras de 16 bits; ciclo de leitura-modificação-escrita; controlo de refrescamento de memórias dinâmicas.

- Entrada para caneta luminosa (light-pen).

- Dispositivo de geração do sinal vídeo.

- Possibilidade de funcionamento em modo gráfico, com endereçamento de 4 Mbits de memória gráfica, em modo alfanumérico, com endereçamento de 8 Kbytes de memória ou em modo misto, podendo endereçar entre 64 Kbytes de memória (funcionamento em modo alfanumérico) e 1 Mbit (funcionamento total em modo gráfico.

- Possibilidade de desenho de segmentos de recta, arcos e circunferências, rectângulos e caracteres gráficos.

- Resolução de 1024x1024 pixels, em modo gráfico, com quatro planos; duas áreas com "scrool" independente.

- Resolução de 100 linhas de 256 caracteres, em modo alfanumérico; possibilidade de programação do avanço e altura do cursor, e da definição, sobre o ecrã, de quatro áreas com "scroll" independente.

- Possibilidade de "zoom", com factores de 1 a 16, e "panning".

- Possibilidade de iniciação do GDC através de comandos.

- Capacidade de DMA, com transferência de bytes ou palavras.

A figura 3.5 mostra o esquema de blocos do GDC.

A comunicação entre o processador central e o GDC é efectuada através de um barramento bidireccional de oito bits. O diálogo estabelece-se usando um conjunto de comandos e parâmetros enviados ou recebidos do GDC através do "buffer" FIFO. O GDC interpreta o conteúdo do "buffer" FIFO, que é executado pelo processador de desenho.

O "buffer" FIFO é usado, simultaneamente, pelo processador central e pelo processador de desenho do GDC; logo foi estabelecido um protocolo de acesso que atribui sempre prioridade à comunicação com o processador central, se bem que este tenha o seu acesso condicionado pela terminação da execução do comando actual.

O controlador gráfico pode gerar dois tipos diferentes de ciclos de acesso à memória: um ciclo de leitura, para efeito

de refrescamento do ecrã, e um ciclo de leitura-modificação-escrita, para actualização do conteúdo da memória. É o próprio GDC que determina, tendo em conta vários parâmetros, nomeadamente os comandos enviados pelo processador central e o estado interno do GDC, qual é o tipo do próximo ciclo de memória, de modo a obter um funcionamento suave sobre o ecrã. Um certo grau de decisão é deixado ao projectista do sistema, que pode programar o GDC de modo a que os ciclos de escrita na memória gráfica se efectuem apenas durante o tempo de retorno: embora a velocidade de escrita diminua, a imagem sobre o ecrã fica isenta de qualquer perturbação.

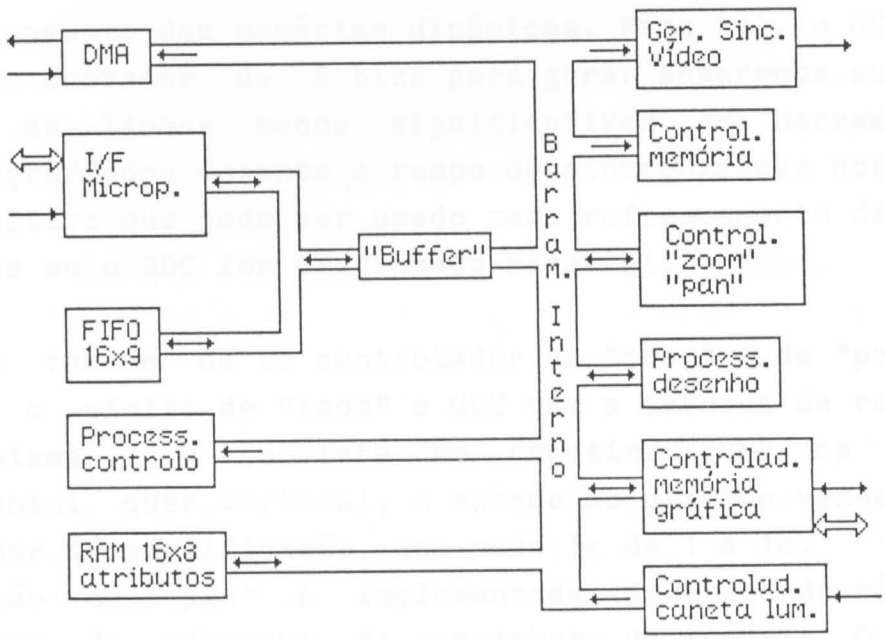


Figura 3.5

A memória gráfica deve ser organizada como um vector de palavras de 16 bits, dado que cada ciclo de leitura acede a uma palavra desta largura. É no entanto possível implementar três tipos diferentes para esta organização: no primeiro, os 16 bits de saída são colocados num barramento de 16 bits, e tornados acessíveis ao controlador gráfico como uma palavra de um plano, caso em que a informação disponível é de 1 bit por pixel; no segundo, o controlador trata separadamente dois ou mais grupos de bits formando palavras de menor comprimento, caso em que existem vários bits de informação por cada pixel; o terceiro tipo de organização combina ambas as técnicas, eventualmente utilizando algum hardware exterior.
O tipo de organização escolhido é programável sobre o GDC.

É também o controlador gráfico o gerador dos sinais de refrescamento das memórias dinâmicas. Para tal, o GDC dispõe de um contador de 8 bits para gerar endereços sucessivos sobre as linhas menos significativas do barramento de endereços/dados durante o tempo de sincronização horizontal, dispositivo que pode ser usado para refrescamento da memória gráfica se o GDC for programado para tal.

Dispõe também de um controlador de "zoom" e de "pan". Para obter o efeito de "zoom" o GDC usa a técnica de replicação de pixel que consiste em repetir, quer na direcção horizontal quer vertical, o estado do pixel n vezes sendo n o factor de amplificação, que pode ir de 1 a 16.
A acção de "pan" é implementada através do cálculo do endereço de arranque do varrimento da memória. O GDC pode ser programado para este efeito, mas o endereço de início está limitado aos extremos iniciais de cada palavra de 16 bits. Uma acção de "pan" suave, com início em qualquer posição de memória pode ser implementada usando alguma lógica adicional, mas tal leva a perder-se a capacidade de "zoom".

O gerador de sincronismo de vídeo tem como função gerar os vários sinais de temporização do monitor, nomeadamente os sinais de sincronismo e "blanking". Estes sinais são expressos em função do ciclo de memória básico (dois ciclos de relógio) e os seus parâmetros podem ser especificados pelo projectista em função do monitor usado, do formato de ecrã desejado e do modo de funcionamento do GDC.
O gerador pode ser programado para funcionar em modo não entrelaçado ou em modo entrelaçado com uma relação de 2:1.

A sintaxe dos comandos para o GDC especifica um formato constituído por uma cabeça e uma cauda. A cabeça é um byte, com significado específico, e a cauda é um conjunto de bytes que são parâmetros de especificação do comando.
A gramática de comandos inclui cinco classes:

- Comandos de desenho: incluem os comandos necessários para a utilização de primitivas do GDC tais como o desenho de segmentos de recta, arcos e circunferências, rectângulos e enchimento de figuras;

- Comandos de leitura: permitem a leitura da memória gráfica, do endereço da caneta luminosa e da posição do cursor;

- Comandos de controlo do vídeo: permitem a iniciação do GDC e a parametrização do sinal de vídeo, nomeadamente o número de linhas, o número de pixels por linha, o modo de funcionamento, etc.

- Comandos de controlo do sinal de saída: permitem o início do varrimento da memória, da especificação do factor de "zoom" e posição do cursor e estabelecimento da partição do ecrã;

- Comandos de controlo de DMA: permitem a parametrização e sincronização da transferência de dados em modo DMA entre o GDC e o processador central.

3.2.2 - A memória TMS4416 da Texas Instruments

Trata-se de um circuito híbrido contendo, sob suporte cerâmico, quatro memórias dinâmicas 4416, com organização 16kx4 bits, interligadas de modo a obter um módulo de 30 pinos com organização 16Kx16 bits.
O tempo de acesso máximo destas memórias é de 200 ns.

3.2.3 - O conversor d/a VDAC444TD da Intech

O conversor digital-analógico previsto foi o VDAC444TD. Trata-se de um conversor vídeo de três canais separados, cada canal constituído por uma "latch" de entrada, ligada em série com um conversor d/a de 4 bits de resolução e 25 ns de tempo de conversão. As saídas do conversor são compatíveis TTL e podem atacar directamente as entradas RGB de um monitor de varrimento, com frequências de pixel até 40 MHz. As entradas das "latch" podem ser ligadas directamente sobre um barramento de dados, a carga dos registos de entrada sendo efectuada por activação de um sinal STROBE. Adicionalmente, poderá ser entregue e activado um sinal de sincronismo, caso em que um dos canais (verde), tem saída composta.

A figura 3.6 mostra-nos o esquema de blocos do conversor utilizado.

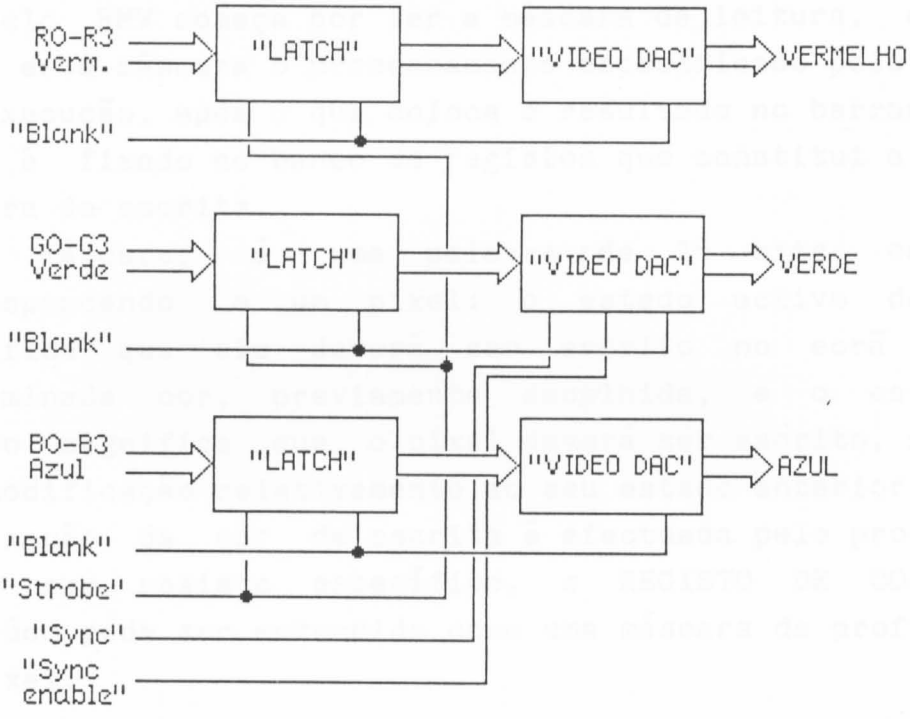


Figura 3.6

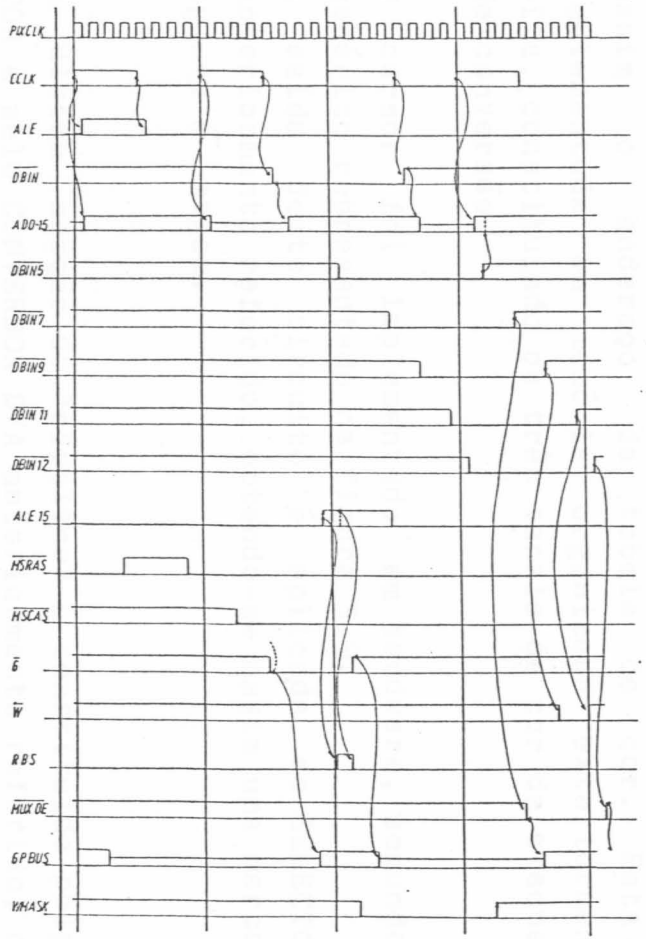
3.2.4 - O princípio de funcionamento

É ao processador local que compete a iniciação do sistema, com o envio dos comandos adequados, seguido do comando para o início de uma actividade específica

conducente à obtenção de uma imagem no ecrã. Em cada momento, apenas um de dois tipos de ciclo pode existir: ou um ciclo RMW, de escrita na memória, figura 3.7, ou um ciclo DISPLAY, de refrescamento do ecrã, figura 3.8.
Como vimos no início deste capítulo, é o controlador que arbitra a execução concorrente destes dois ciclos.

O ciclo RMW começa por ler a máscara de leitura, efectuar sobre esta máscara o processamento especificado pelo comando em execução, após o que coloca o resultado no barramento, o qual é fixado no banco de registos que constitui a sede da máscara de escrita.
Esta máscara, é uma palavra de 16 bits, cada bit correspondendo a um pixel: o estado activo do pixel significa que ele deverá ser escrito no ecrã com uma determinada cor, previamente escolhida, e o estado não activo significa que o pixel deverá ser escrito, no ecrã, sem modificação relativamente ao seu estado anterior.
A fixação da cor de escrita é efectuada pelo processador local num registo específico, o REGISTO DE COR, cujo conteúdo pode ser entendido como uma máscara da profundidade do pixel.

Simultaneamente com a leitura da máscara é feita a leitura da memória gráfica no endereço especificado, a qual é aplicada a uma das quatro entradas de um multiplexador; as restantes entradas contêm, uma, o complemento do BUFFER DE LEITURA, obtido por simples inversão; as outras duas, respectivamente, a máscara do REGISTO DE COR e a máscara do REGISTO DE CURSOR, registos que são carregados directamente pelo processador local.
A saída deste multiplexador é, em cada momento, definida pelo SELECTOR DE OPÇÕES, que fornece a palavra de controlo do multiplexador a qual é determinada pela composição lógica do bit correspondente da máscara de escrita, do estado do REGISTO DE CURSOR (que permite inserir uma marca de cursor



ciclo RMW

Figura 3.7 - ciclo RMW

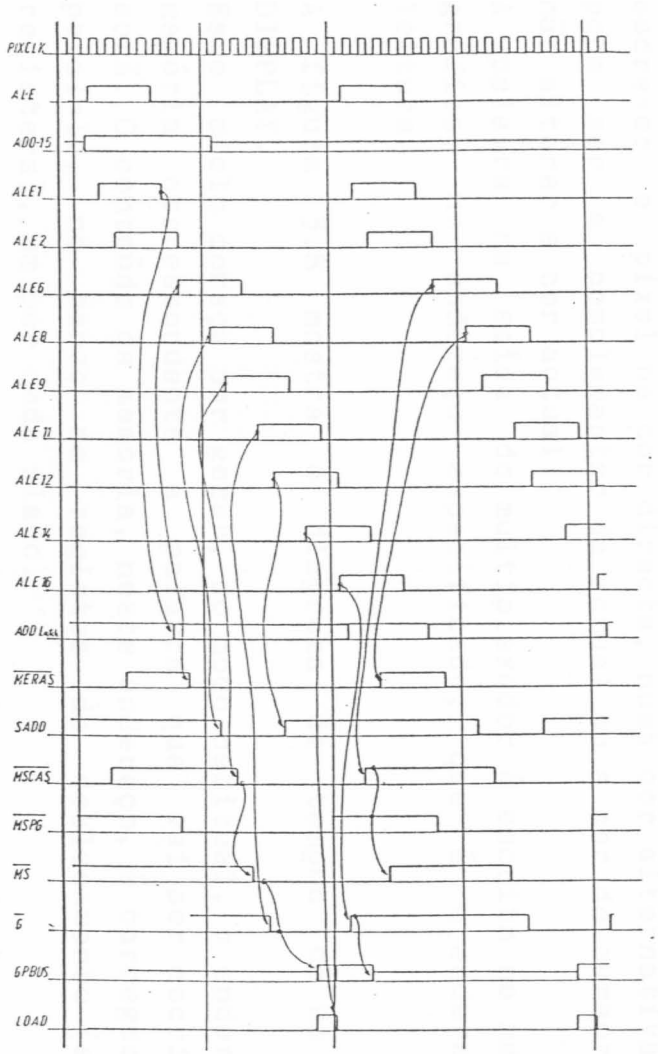


Figura 3.8 - ciclo DISPLAY

em qualquer ponto do ecrã), e de um REGISTO DE SELECÇÃO, carregado pelo processador local, que permite escolher entre escrever o pixel na cor directa, numa cor alternativa (que pode ser a complementar da actual ou a cor do cursor), ou não alterar a cor actual.
A palavra na saída do multiplexador é escrita na memória gráfica, no endereço especificado, que é o endereço de leitura.

A figura 3.8 mostra o diagrama de tempos de um ciclo DISPLAY.
Este ciclo começa por gerar, do modo habitual, o endereço de memória correspondente à palavra que vai ser escrita no ecrã. O conteúdo da memória, neste endereço, é carregado, em paralelo, no banco de registos de deslocamento (shift-registers), um por cada plano.
A palavra carregada em cada registo de deslocamento é serializada á frequência do relógio de pixel construindo-se assim o endereço da tabela de cor. Esta tabela é implementada em memória organizada exteriormente em 3x4 bits, constituindo os três canais de cor de ataque da secção de conversão.

O cursor foi implementado em hardware, segundo o esquema genérico representado na figura 3.9.
A saída deste circuito é aplicada ao SELECTOR DE OPÇÕES anteriormente referido, obtendo-se assim uma marca de cursor aposta à imagem.

A última secção do "pipeline" é constituída pela TABELA DE COR e pelo CONVERSOR D/A anteriormente referido, que gera os sinais de cor fundamentais a entregar ao monitor. A lógica de sincronização gera, do modo habitual, os sinais de sincronismo vertical e horizontal e os sinais de controlo do

conversor.

A marca fundamental do funcionamento do subsistema gráfico como um "pipeline" de três secções é dada pelo sinal LOAD,

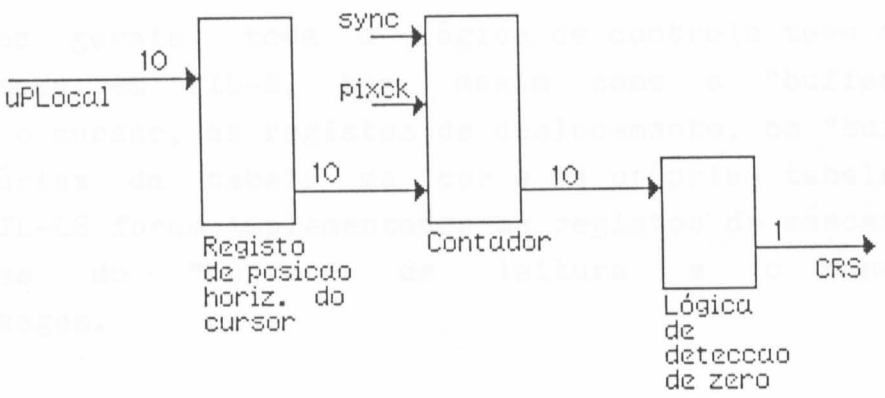


Figura 3.9

de carga de uma palavra nos registos de deslocamento, no ciclo DISLAY.
Esta palavra é processada na secção de serialização durante um intervalo de tempo máximo

$$t = 16 \times tpixel$$

Ao fim deste tempo o bit mais significativo da palavra carregada no registo de deslocamento está a abandonar a secção, sendo a sua imagem, produzida pela tabela de cor, entregue ao conversor d/a.
Este é pois o tempo de trânsito em cada secção: enquanto uma dada palavra está a ser serializada, a palavra seguinte está a ser processada pelo controlador gráfico, eventualmente na secção de processamento gráfico, e a palavra anterior está a ser convertida em sinal analógico e enviada ao monitor.

A fixação do tempo de pixel, isto é, da frequência de relógio de pixel, fixa as características dos dispositivos a utilizar em cada secção, nomeadamente a tecnologia dos circuitos integrados digitais, do conversor d/a e do monitor.

Em termos gerais, toda a lógica de controlo teve de ser implementada em TTL-S, bem assim como o "buffer" de leitura, o cursor, os registos de deslocamento, os "buffers" das memórias da tabela de cor e as próprias tabelas; em lógica TTL-LS foram implementados os registos de máscara, os inversores do "buffer" de leitura e o banco de multiplexagem.

3.3 - A segunda implementação do terminal gráfico

Esta segunda implementação, também realizada no INESC, no âmbito dos trabalhos do projecto ELENA, foi concebida para permitir uma resolução máxima de 960x720 pixels e 16 cores simultâneas, escolhidas numa paleta de 4096 cores, podendo também funcionar em média resolução com 680x512 pixels, e 256 cores simultâneas.
O sistema permite visualizar simultaneamente gráficos e texto, possuindo janelas separadas, uma para gráficos e outra para texto, podendo-se definir a dimensão e localização no ecrã de cada uma das janelas. Permite ainda fixar o número de "bit planes" adstritos à janela gráfica, obtendo-se por essa via a possibilidade de escrita selectiva nos vários planos.
Relativamente à visualização de texto, a escrita alfanumérica pode ser feita em duas cores, uma para fundo e outra para o carácter, qualquer que seja a resolução utilizada no canal gráfico. Quer em modo gráfico quer em

modo texto, a paleta disponível é de 4096 cores.
O sistema dispõe ainda, para comunicação com o exterior, de um interface paralelo, dois interfaces série RS232C e de um canal DMA de 2 Mbytes/s.

A figura 3.10 contém uma descrição da arquitectura global escolhida para esta implementação. Como vemos, é ainda uma arquitectura que trata de modo separado gráficos e texto, embora modificada relativamente ao modelo clássico de arquitectura, escolhido para ambas as implementações.

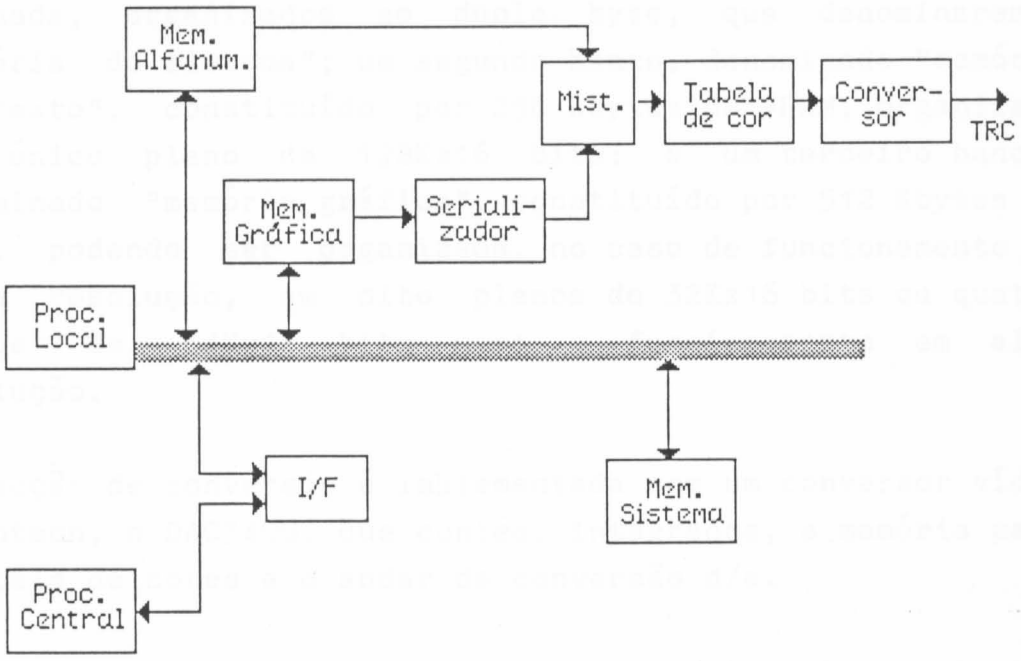


Figura 3.10

Neste caso o processador local é um processador de uso geral, o iAPX186, ao qual cabe a execução de todo o cálculo e processamento gráfico e alfanumérico e ainda algum processamento de iniciação e controlo, nomeadamente a carga dos registos de definição de cores.

O controlo global do sistema é efectuado por um dispositivo específico, o controlador TMS34061 e alguma lógica adicional, que gera os sinais de controlo quer para a secção de memória gráfica e alfanumérica, quer para a secção de serialização, quer para a de conversão d/a.

O sistema de memória é constituído por três bancos distintos, um, com 256 Kbytes de VRAM e até 64 Kbytes de ROM paginada, organizados ao duplo byte, que denominaremos "memória do sistema"; um segundo banco, denominado "memória de texto", constituído por 256 Kbytes de VRAM, organizada num único plano de 128Kx16 bits; e um terceiro banco, denominado "memória gráfica", constituído por 512 Kbytes de VRAM, podendo ser organizada, no caso de funcionamento em média resolução, em oito planos de 32Kx16 bits ou quatro planos de 64Kx16 bits para o funcionamento em alta resolução.

A secção de conversão é implementada com um conversor vídeo da Intech, o DAC3408, que contém, integrados, a memória para a tabela de cores e o andar de conversão d/a.

O processador central comunica com o sistema local através de um dispositivo dedicado de interface, o TMS9650 da Texas Instruments.

3.3.1 - O Interface Multiprocessador TMS9650 da Texas Instruments

Este circuito é, fundamentalmente, um dispositivo de comunicação assíncrona dispondo de dois portos paralelos associados a interfaces para comunicação com processadores, um "buffer" de memória de 256 bytes e um mecanismo de controlo quer da activação dos portos, quer da arbitragem do acesso.
O banco interno de memória, do tipo FIFO circular, só pode ser acedido, em cada momento, por um dos processadores, que lhe deve requisitar o uso reservado, testando e escrevendo num registo específico.
No caso de acesso concorrente, o primeiro processador a pedir acesso ganha o barramento, excluindo o outro processador; no caso de conflito, o mecanismo de arbitragem, que é de natureza bi-estável, transita para um estado instável. Esta transição é entendida, do ponto de vista dos registos de estado, como estando o barramento não acessível para qualquer dos processadores. A transição do estado instável para o estado estável tem natureza quase aleatória e, após ocorrer, um dos processadores ganha o barramento, excluindo o outro.

3.3.2 - O controlador de vídeo TMS34061 da Texas Instruments

O controlador de vídeo TMS34061 (VSC - Video System Controller) é um dispositivo que implementa os interfaces entre o sistema de memória gráfica, o processador gráfico e o sistema de vídeo, gerando todos os sinais de controlo e temporização necessários para tal.

As principais funções do VCS são as seguintes:

- Gerar os sinais de controlo para o sistema de memória gráfica;

- Gerar os sinais de controlo para o monitor de varrimento;

- Gerar os sinais de controlo dos ciclos de memória para o refrescamento do ecrã;

- Gerar os sinais de refrescamento da RAM dinâmica;

- Controlar a comunicação entre o processador e a memória;

- Arbitrar todas as actividades envolvendo o acesso à memória, incluindo a carga dos registos de deslocamento internos da VRAM e refrescamento da DRAM.

Adicionalmente, o VSC implementa outras funções gráficas, nomeadamente, permite implementar o modo de endereçamento X-Y, obtendo-se assim um melhor desempenho na execução das primitivas gráficas, controla a transferência bidireccional entre a memória e o registo de deslocamento das VRAMs, e controla a actividade dos blocos de vídeo e de processamento, os quais podem funcionar assincronamente e a velocidades diferentes.
Como veremos, as características do VSC, são especialmente adequadas para a sua utilização em sistemas gráficos com memória "bit map" implementada com VRAMs, indicando o fabricante que este controlador, nesta arquitectura, confere uma muito elevada largura de banda na comunicação processador-memória, já que o tempo total necessário para

refrescamento do ecrã e das memórias DRAM, mercê da eficiência do dispositivo de arbitragem, não excede 6% do total da largura de banda do barramento do sistema.

3.3.3 - A vídeo-RAM TMS4161 da Texas Instruments

Esta memória vídeo é um dispositivo especialmente concebido para ser utilizado em sistemas gráficos de varrimento do tipo TV. De facto, este tipo de arquitectura requer, tal como já várias vezes referimos, que a saída da memória gráfica ataque a entrada de um andar de serialização, tipicamente constituído por um banco de registos de deslocamento. A memória TMS4161 combina uma RAM dinâmica organizada em 64Kx1 bits com um registo de deslocamento de 256 bits, e inclui o controlo necessário para a transferência de dados entre o banco de memória e o registo de deslocamento.

O circuito é assim uma memória RAM dinâmica de dois portos: um porto da acesso ao banco de memória e outro ao registo de deslocamento. Dado que é possível efectuar acesso quase simultâneo aos dois portos, a vídeo RAM permite que o processador gráfico aceda à memória concorrentemente com a leitura para a serialização, sem necessidade de implementar qualquer tipo de exclusão mútua entre os dois ciclos. A única excepção que se deve ter em conta é a inacessibilidade de qualquer dos dois portos durante a carga do registo de deslocamento interno.
Este registo de deslocamento é configurado como uma ligação série de quatro registos de deslocamento de 64 bits, o que permite organizar a saída em 64x1, 128x1, 192x1 ou 256x1 bits, sendo a sua máxima frequência de 25 MHz.

Uma facilidade adicional oferecida por este dispositivo refere-se à possibilidade de, sob controlo externo, efectuar a carga da memória para o registo de deslocamento e, de novo, do registo de deslocamento para a memória. Como veremos, esta facilidade permitirá implementar, com alguma eficiência, algumas actividades correntes nos sistemas gráficos, nomeadamente, a limpeza de uma área ou de todo o ecrã, o "scroll", etc.

3.3.4 - O conversor vídeo DAC3408, da Intech

Trata-se de um conversor digital-analógico híbrido, constituído por três canais semelhantes e diferenciados, produzindo três saídas independentes, compatíveis TTL ou CMOS, adaptadas para atacar directamente as entradas RGB de um monitor de varrimento, com frequência de pixel até 40 MHz.
Cada canal é constituído por um banco de memória RAM estática de 256x4 bits, um circuito "latch" e um conversor d/a de 4 bits de resolução e 25 ns de tempo de conversão.
O banco de memória destina-se a conter a paleta de cores, a qual pode ser carregada por um processador externo. Um sinal de STROBE activa as "latchs" à frequência de pixel, transferindo dados da tabela de cor para o registo.
A palavra de quatro bits aposta na entrada do conversor d/a é traduzida num nível de tensão analógica, numa gama possível de 16 níveis, correspondentes a 16 gradações da cor do canal.
Assim, o número total de cores simultâneas permitidas é de 256 num total de 4096.

3.3.5 - O princípio de funcionamento

A figura 3.11 representa o esquema detalhado desta arquitectura. A "display-list" é carregada na memória do sistema, quer a partir do processador central, normalmente utilizando o canal DMA, quer localmente, usando o processador local, a partir de um periférico comunicando com o subsistema gráfico através do interface série.
Estando carregada a "display-list", o processador executa-a, utilizando um interpretador que, eventualmente, pode existir no banco de EPROM da memória do sistema. O resultado é escrito na memória gráfica, ou na memória de texto, a qual também está organizada como memória "bit map".
A escrita na memória gráfica pode ser mascarada usando o conteúdo de um registo específico, carregado pelo processador local.
Este registo tem o comprimento de 16 bits, sendo oito usados como máscara de escrita, cada um dos oito bits sendo uma máscara para o respectivo "bit plane". Os restantes bits do registo permitem seleccionar os bancos de cor na tabela de cores e a transparência ou opacidade da janela de texto no ecrã.

O acesso a estas memórias é controlado pelo TMS34061. Este controlador comanda, arbitrando, a escrita na memória gráfica ou na memória de texto, e a respectiva leitura, para carga dos registos de deslocamento internos e consequente serialização.
A passagem, em modo de leitura, da memória gráfica para a memória de texto é da responsabilidade de um "timer" do processador local, que conta as linhas de vídeo: quando o número de linhas coincide com o valor previamente carregado num registo específico, a leitura transita da memória gráfica para a memória de texto. Note-se que esta memória

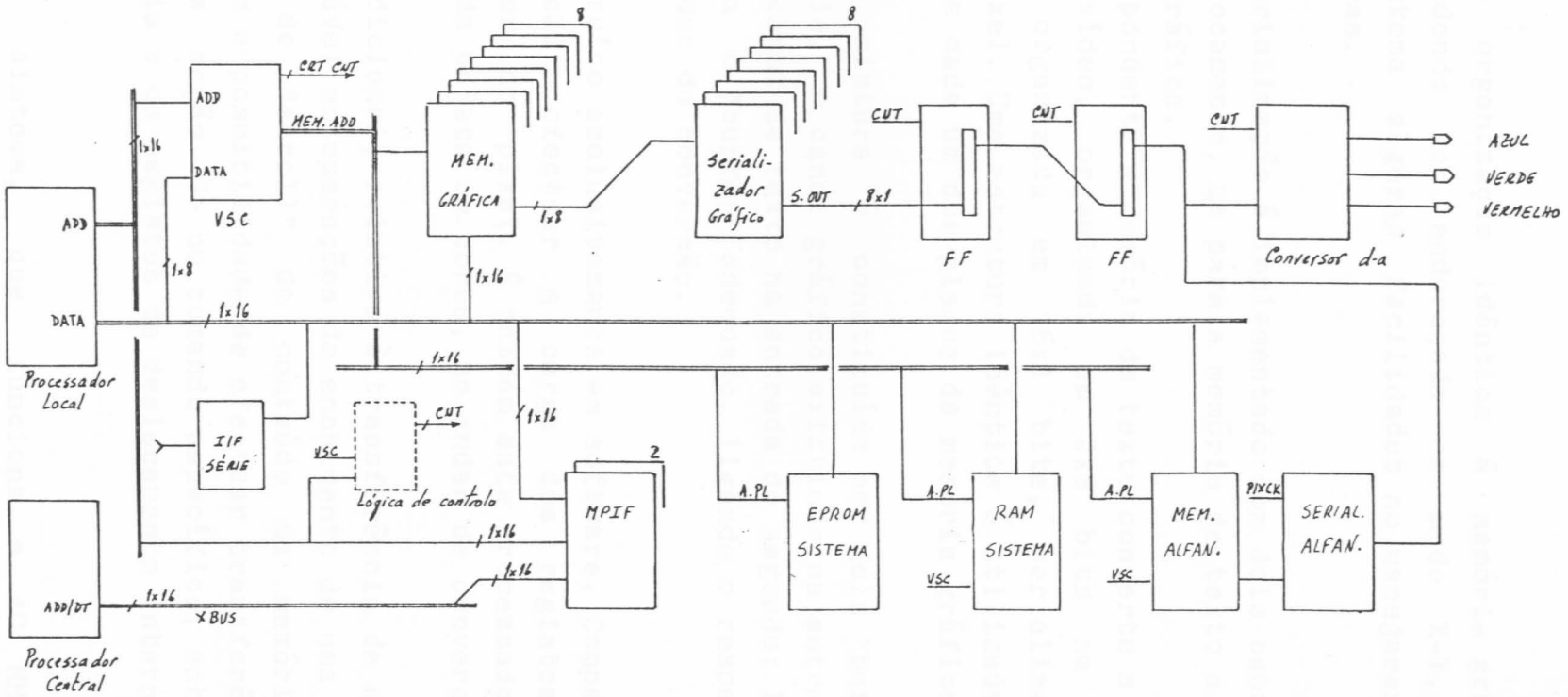

ADD
DATA
Processador Local
1x16
1x8
VSC
ADD
DATA
CRT CNT
MEM. ADD
MEM. GRÁFICA
8
1x8
1x16
Serializador Gráfico
8
S. OUT
8x1
CNT
FF
CNT
FF
CNT
Conversor d-a
AZUL
VERDE
VERMELHO
1x16
IIF SÉRIE
VSC
CNT
Lógica de controlo
1x16
MPIF
2
1x16
1x16
ADD/DT
1x16
XBUS
Processador Central
A.PL
EPROM SISTEMA
A.PL
RAM SISTEMA
VSC
A.PL
MEM. ALFAN.
VSC
PIXCK
SERIAL. ALFAN.

Figura 3.11

tem natureza e organização idêntica à memória gráfica, nomeadamente podendo ser endereçada em modo X-Y, o que confere ao sistema algumas facilidades no manejamento de padrões e texturas.

O andar de serialização é implementado com dois bancos de registos de deslocamento, um para a memória de texto e outra para a memória gráfica.
O banco correspondente à memória de texto converte a saída das memórias vídeo, organizada em 2x8 bits na saída correspondente organizada em 16x1 bits, serializada à frequência do pixel. Uma estrutura idêntica é utilizada para a serialização de cada um dos planos da memória gráfica.

O interface de mistura é constituído por dois "buffers" ligados em série, o canal gráfico existindo na entrada do primeiro e o canal de texto na entrada do segundo; lógica adicional activa o "buffer" adequado, ligando o respectivo canal sobre o andar de conversão.

O cursor é construído exclusivamente em software. Compete ao processador local efectuar a carga dos registos que determinam a cor do pixel. É também este processador que efectua a carga da paleta de cores, no andar de conversão.

Uma facilidade adicional, relativa à transferência de blocos de bits, utilizável em operações de enchimento de uma parte da memória ou de "scroll" do conteúdo da memória, é implementada com a possibilidade de efectuar transferências directas, sob a acção de um comando específico, entre os bancos da memória e os registos de deslocamento internos, e vice-versa.

O relógio do sistema, que funciona a 40 MHz, é sucessivamente dividido e logicamente condicionado de modo a obterem-se os sinais síncronos de comando das secções,

nomeadamente, a carga dos registos de deslocamento internos das VRAMs, o circuito interface de arbitragem e a carga das "latchs" do conversor d/a, gerando também o sinal de relógio a entregar ao controlador vídeo; o processador local trabalha assincronamente relativamente aos canais gráfico e de texto, a frequência inferior (relógio externo de 16 MHz), fornecendo os sinais de relógio a entregar quer ao controlador de vídeo quer ao controlador do barramento.

3.4 - Uma avaliação das arquitecturas testadas

Da primeira implementação descrita, que por comodidade de referência designaremos por Terminal Gráfico 1 (TG1), para a distinguir da segunda implementação, que designaremos por Terminal Gráfico 2 (TG2), apenas foi implementado o canal gráfico.

A primeira característica a referir, quando efectuamos uma avaliação do TG1, é o muito reduzido nível de integração, que tornou extremamente difícil a afinação (debug) do sistema. De facto, só o canal gráfico do TG1 utilizava 171 circuitos integrados distribuídos por três placas com formato Eurocard duplo, tendo sido feita a sua montagem usando "wiring-wrap". Isto dá uma ideia da densidade da implementação, sobretudo se compararmos com os 83 circuitos integrados utilizados no TG2, que foi completamente implementado.

Um segundo aspecto que importa referir diz respeito ao facto de o circuito de controlo utilizar uma temporização extremamente apertada, em algumas zonas junto dos limites de velocidade da tecnologia TTL: a título de exemplo, não foi possível uma tentativa de integração do dispositivo de

leitura-modificação-escrita em memória usando PALs, porque a arquitectura disponível na PAL não coincidia com a utilizada, sendo necessária uma reconfiguração que não foi possível efectuar, dado que o tempo disponível para trânsito neste dispositivo não o permitia.
Embora o aumento do nível de integração fosse possível em algumas zonas do circuito de controlo, tal não modificaria o risco que comporta trabalhar junto dos limites máximos de velocidade de uma lógica. Isto significa que a resolução inicialmente fixada, de 640x480 pixels, se situava nos limites de funcionamento seguro do sistema.

A segunda característica importante da arquitectura utilizada no TG1 diz respeito ao facto de o uPD7220 não utilizar directamente as suas capacidades de controlador gráfico, nomeadamente as de actualização da memória gráfica. De facto o GDC, em funcionamento normal, actualiza a memória gráfica fazendo a leitura da palavra existente no endereço especificado, processando-a, e reescrevendo-a naquele endereço. O processamento inclui não só o cálculo do novo conteúdo, mas também um conjunto de operações lógicas, nomeadamente, as de "REPLACE", "COMPLEMENT", "SET" e "CLEAR", as quais, eventualmente, podem ser executadas apenas sobre alguns dos bits da palavra, utilizando máscaras internas do GDC, externamente acessíveis.
Ora todas estas facilidades, directamente utilizáveis quando o GDC está a escrever sobre um só plano, não puderam ser aproveitadas nesta arquitectura, dado o princípio de funcionamento utilizado.
É claro que por este processo se conferiu à arquitectura um alto grau de paralelismo, e logo um potencialmente elevado desempenho do sistema gráfico; o preço pago foi a complexidade da implementação e o desaproveitamento das possibilidades do GDC: mesmo as operações de "SET" e

"COMPLEMENT" foram implementadas usando lógica externa. Foram, no entanto, completamente aproveitadas as potencialidades de cálculo e execução das primitivas gráficas do GDC e também a generalidade dos sinais de controlo e sincronização da secção de conversão; já todo o controlo da secção de serialização teve de ser implementado externamente.
Se quizermos fazer uma descrição ligeira e sintética, sem grande rigor mas usando uma imagem simples, poderemos dizer que a arquitectura proposta no TG1 implementa um controlador gráfico de alto grau de paralelismo, dispondo de um coprocessador de primitivas gráficas em hardware.

A terceira característica diz respeito ao modo como o canal de texto foi projectado.
O GDC pode funcionar em três modos diferentes: gráfico, texto e misto.
É possível programar o GDC em modo misto, definindo duas janelas, uma gráfica e outra de texto. O projecto do canal de texto, que se mostra na figura 3.12, utilizava estas facilidades.

O GDC carregava um "buffer" com o texto, cada comando constituindo o endereço de um gerador de caracteres em EPROM. O endereço era completado com um contador configurado para uma contagem de 12, estando o carácter definido numa matriz de 8x12 pixels. A saída do gerador de caracteres era seguidamente condicionada pelas máscaras de cor do carácter e fundo, utilizando um modo idêntico ao do canal gráfico, e aposta no barramento da secção de processamento, para ser serializada.

A escolha da largura de oito pixels é praticamente imposta pelo funcionamento do GDC. De facto, em modo gráfico, o controlador escreve uma palavra de 16 bits, usando quatro ciclos de relógio; em modo texto usa apenas dois ciclos de

relógio para escrever um carácter.

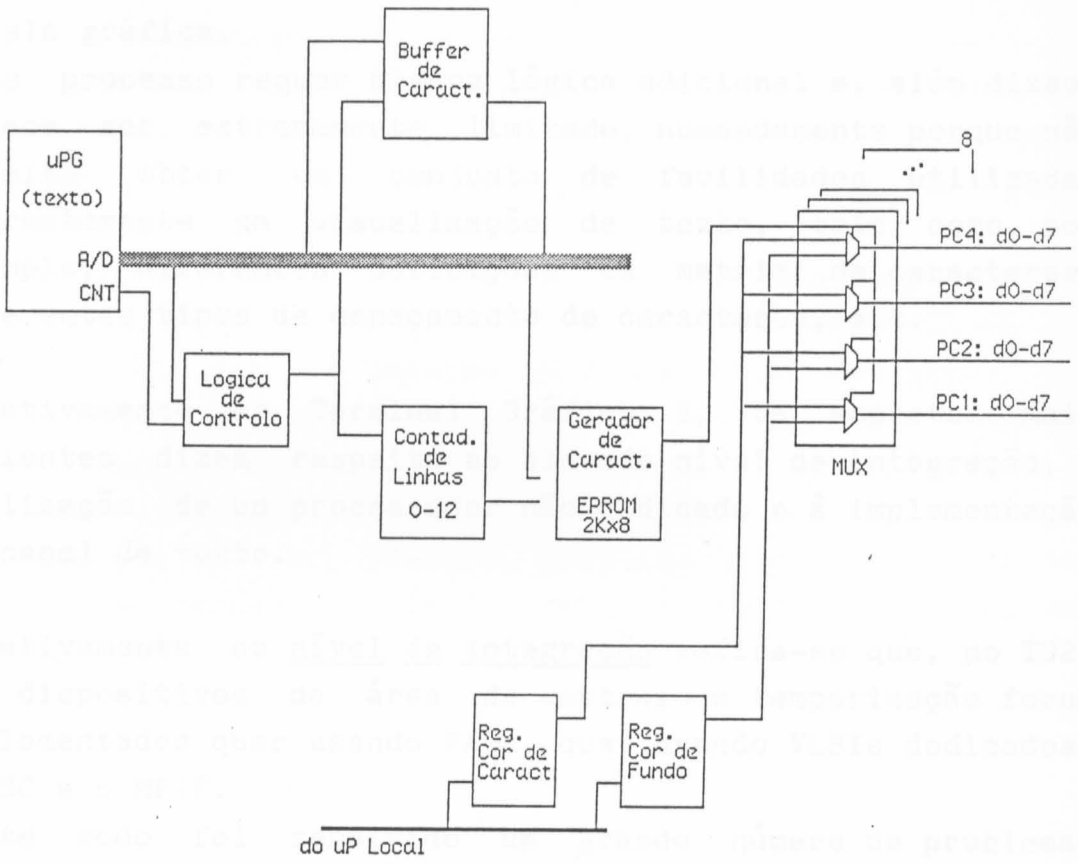


figura 3.12

A compatibilidade do funcionamento, na visualização simultânea de gráficos e texto, só pode ser obtida recorrendo-se a um artifício: escolher, para o carácter, a largura de oito pixels, para que a escrita de dois caracteres seja compatível com a de uma palavra gráfica de 16 bits. É também necessário gerar um sinal de carga dos registos de deslocamento da secção de serialização com a largura de dois ciclos de relógio, o qual será directamente

utilizado quando o controlador está na janela de texto, devendo ser dividido por dois quando o controlador está na janela gráfica.
Este processo requer alguma lógica adicional e, além disso, parece ser extremamente limitado, nomeadamente porque não permite obter um conjunto de facilidades utilizadas correntemente na visualização de texto, tais como, por exemplo, diferentes definições da matriz de caracteres, diferentes tipos de espaçamento de caracteres, etc.

Relativamente ao Terminal Gráfico 2, os aspectos mais salientes dizem respeito ao elevado nível de integração, à utilização de um processador não dedicado e à implementação do canal de texto.

Relativamente ao nível de integração refira-se que, no TG2, os dispositivos da área de controlo e temporização foram implementados quer usando PALs, quer usando VLSIs dedicados, o VSC e o MPIF.
Deste modo foi resolvido um grande número de problemas relacionados com a construção de sistemas digitais complexos usando MSI, que no TG2 se veriam agravados com o facto de se utilizar um processador de uso geral, logo não contendo quaisquer facilidades para o interface quer com o resto do sistema gráfico, quer com o processador central. Como o VSC, além de gerar os sinais de controlo e temporização, também efectua a arbitragem dos acessos à memória, a configuração resultante da sua utilização, juntamente com a implementação da lógica discreta em PAL, traduz-se numa solução simples e elegante.

A utilização de um processador de uso geral, mesmo tratando-se de um processador rápido pode suscitar algumas questões: se bem que confere ao sistema excelentes

potencialidades e flexibilidade no que se refere à implementação algorítmica, terá de ser necessariamente lento, quando confrontado com processadores especializados, na execução das operações gráficas. O seu desempenho deve ser, inclusivamente, comparativamente mais fraco na realização de operações concebidas especialmente tendo em vista melhorar o desempenho tais como, por exemplo, a transferência de blocos de pixels com mascaramento e alinhamento.
Também a obtenção de janelas no ecrã, não as janelas de formatação, que são definidas em hardware pelo "TIMER" do processador, mas as janelas de visualização de imagem, que necessitam de operações de "clipping", exigindo uma carga de cálculo muito elevada, poderão fazer diminuir bastante o desempenho, independentemente do acelerador no acesso à memória que o VSC integra, dado não disporem de qualquer suporte em hardware. O INESC está actualmente a desenvolver software para este sistema pelo que, oportunamente, será possível concluir quanto ao desempenho que esta arquitectura permite.

Quanto à questão da visualização de texto a solução adoptada, embora não seja conceptualmente diferente da utilizada anteriormente, aparece agora naturalmente integrada na arquitectura do sistema. Tal facto resulta sobretudo da elegância da configuração geral, que usa os vários subsistemas de memória como parte de um sistema global com controlo unificado, este controlo sendo efectuado pelo VSC. Claro que, sendo o processador do sistema um processador de uso geral, estão abertas todas as possibilidades quanto à implementação da escrita, embora tarefas importantes nesta área venham a ter, provavelmente, e tal como acontece na escrita gráfica, influência na diminuição do desempenho.
Tal deve verificar-se, pelo menos, na construção das funções de espaçamento de texto (proporcional, sobreposto, etc.), e

de rotação e alinhamento, mais uma vez por não terem suporte em hardware.

Uma característica comum às duas implementações testadas refere-se ao facto de ambas assentarem no modelo clássico de arquitectura referido na figura 3.1, com tratamento separado dos canais de texto e gráfico, embora comportem modificações importantes relativamente àquele modelo.
Assim, o TG1 utiliza um controlador gráfico para construir, quer a imagem quer o texto, em memória. Na construção da imagem o controlador gráfico foi usado de modo especial, para se obter um maior grau de paralelismo, embora funcionando de modo corrente na construção de texto. Outras soluções seriam no entanto possíveis e aparecem referidas na literatura.

Por exemplo, (20) refere a utilização do mesmo controlador gráfico num sistema distribuído com três controladores, um para cada plano de cor, utilizando completamente as suas facilidades.
Cada plano de cor dispunha também do seu próprio banco de memória, de modo a que os três controladores funcionavam independentemente, comandados a partir do processador local ou central. A mesma arquitectura é usada para implementar o canal de texto, necessariamente com menor resolução. Este canal dispõe de um quarto controlador e do respectivo banco de memória. Os canais gráfico e de texto funcionam de modo completamente separado, existindo um interface que permite seleccionar, a partir do processador local, o canal gráfico, o canal de texto ou sobrepor ambos, sendo este último transparente. A escrita de texto efectua-se com as restrições referidas anteriormente.
O autor refere dificuldades existentes na sincronização do funcionamento dos quatro controladores, organizados como um sistema mestre-escravo.

Esta dificuldade foi posteriormente atenuada com a utilização de uma nova versão do controlador (uPD7220A), que inclui mecanismos adicionais para sincronização do conjunto de controladores implementados quer em hardware quer em software.

Uma alteração desta arquitectura é referida em (14), sendo o seu objectivo conferir maiores facilidades à escrita de texto. Neste caso é utilizado um coprocessador de texto, o 82730 da Intel, juntamente com o controlador 82720, que é a versão da Intel do controlador 7220.
O coprocessador de texto funciona agora a alta velocidade, de modo optimizado relativamente ao controlador gráfico, numa organização mestre-escravo. O coprocessador 82730 dispõe do conjunto de facilidades normalmente utilizadas na edição de texto.

O TG2 realiza uma aproximação diferente desta arquitectura: Usa-a praticamente no seu desenho tradicional, mas implementa-a utilizando componentes modernos, com alto nível de integração e velocidade, que optimizam o funcionamento das zonas críticas, nomeadamente a saturação do barramento de memória, o que vai permitir obter, decerto, um desempenho francamente acrescido, relativamente à implementação tradicional.

3.5 - Conclusão

A escolha das características para os processadores (e outros componentes do sistema gráfico) deve ter em vista a definição das características gerais dos sistemas gráficos: desempenho elevado, nomeadamente no que se refere à

capacidade de actualização da imagem no ecrã, grande interactividade, com um diálogo utilizador-sistema rápido e eficiente, e capacidade de formação de imagens realistas, isto é, elevada disponibilidade em cor e resolução e, por último, elevado nível de integração texto-imagem.

3.5.1 - O tratamento unificado de texto e imagem

A tendência actual é a da procura de arquitecturas capazes de suportarem de forma integrada texto e gráficos, com elevada capacidade de cor e resolução e possibilidade de construção de janelas de ecrã. Estas determinantes apontam para a utilização de um novo modelo de arquitectura de sistemas gráficos, figura 3.13.

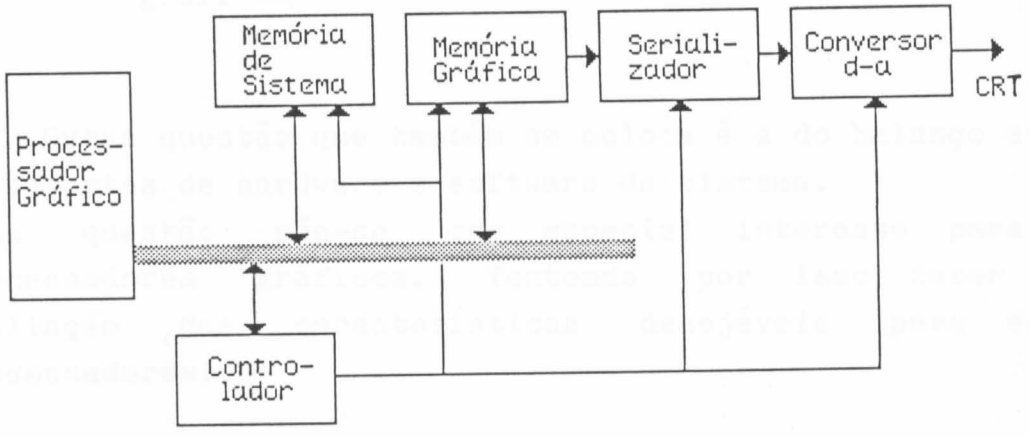


Figura 3.13

A diferença fundamental reside no facto de existir um único canal para texto e gráfico, "bit map", ambas as representações sendo tratadas de modo unificado.
Esta arquitectura unificada aparece referida na literatura, muitas vezes com a apresentação de novos produtos. No entanto, quando a avaliamos do ponto de vista do desempenho parece conter ainda alguns pontos fracos. Por exemplo, mesmo supondo que com as vídeo RAMs e novos controladores o problema da saturação do barramento de acesso à memória estava satisfatoriamente resolvido, não o está o problema da velocidade de escrita na memória, de modo a melhorar significativamente as velocidades de acesso.
Esta questão pode estar ligada quer à implementação de elevados graus de paralelismo, quer ao projecto dos processadores gráficos, quer à organização interna da memória.

3.5.2 - Características desejáveis para os processadores gráficos

Outra questão que também se coloca é a do balanço entre os suportes de hardware e software do sistema.
Esta questão põe-se com especial interesse para os processadores gráficos. Tentemos por isso fazer uma avaliação das características desejáveis para estes processadores.

Podemos considerar três classes de características desejáveis, não completamente independentes: as relacionadas com a definição funcional, com o desempenho e com o tipo de implementação.
Relativamente às características funcionais deve procurar-se, fundamentalmente, uma grande flexibilidade na construção

das primitivas gráficas.
A utilização de um controlador gráfico, se bem que desejável do ponto de vista da velocidade de execução é, em geral, bastante limitada no que respeita às primitivas implementadas. Parece desejável não limitar de uma forma tão drástica as possibilidades gráficas residentes em hardware, o que tem o seu custo em termos da construção de primitivas e comandos em alto nível. Também a solução de não implementar nenhuma primitiva gráfica em hardware, deixando a um processador de uso geral a tarefa de as construir tem inconvenientes, que se traduzem em geral em baixo desempenho global do sistema.
Parece pois conveniente implementar algumas primitivas gráficas directamente em hardware sem impedir a possibilidade de o utilizador definir outras, em software.
Se esta é quase uma conclusão de senso comum, já a repartição do que deve ser implementado em hardware e software é mais discutível: em termos gerais, pensamos que, do ponto de vista do conjunto de instruções, devem existir as operações correntes de movimento de dados, saltos condicionais e incondicionais e manejamento de subrotinas. O processador deverá ter um número elevado de registos internos de uso geral e, eventualmente, unidades funcionais aritméticas coprocessadoras. Do ponto de vista das primitivas gráficas implementadas em hardware, devem existir algoritmos incrementais eficientes para o traçado de segmentos de recta, arcos e poligonais, deverá ser possível efectuar operações lógicas quer sobre um pixel, quer sobre um conjunto de pixels, e também translações, rotações e mudanças de escala. Outros tipos de transformações, tais como "zoom" e "pan" devem também ser implementados em hardware, tal como deve existir a possibilidade de endereçamento X-Y, devendo fazer-se por hardware a sua tradução em endereços de memória.

Finalmente, satisfazendo as tendências actuais, devem permitir a utilização de janelas, pelo que deverão dispôr, implementada em hardware, de capacidade de "clipping".

Quanto à característica de desempenho a tendência actual parece ser a utilização de paralelismo, pelo que se deverá procurar aumentar a velocidade dos dispositivos e aumentar o nível de paralelismo. Assim, será de prever o aparecimento de processadores gráficos trabalhando com 32 ou mais bits. Outra questão, já referida anteriormente, diz respeito à procura de novas organizações da escrita em memória devendo, nessa eventualidade, o processador estar optimizado para essas organizações.

Finalmente, é obviamente desejável procurar altos níveis de integração quer da lógica de controlo quer da temporização do sistema.

CAPÍTULO 4 - O estado-da-arte dos dispositivos para subsistemas gráficos

4.1 - Introdução

Escolhemos para análise um conjunto de dispositivos tornados disponíveis, pelos fabricantes, durante o ano de 1986 e no início de 1987.
Consideramos ser este ano, 1986, um ano de fronteira na concepção de dispositivos para sistemas gráficos, concepção que utiliza a experiência adquirida nos anos anteriores com os controladores gráficos da primeira geração, tais como o uPG7220.
O critério para a escolha dos dispositivos foi o enunciado no fim do capítulo anterior.
Assim, na área dos processadores gráficos, referiremos os controladores gráficos 82786 da Intel, o qual representa uma evolução directa dos controladores da primeira geração; o HD63484 da Hitachi, por ser um dos mais potentes controladores actualmente oferecidos; o Am95C60 da AMD, por ser o primeiro controlador a realizar uma aproximação à arquitectura unificada, anteriormente referida.
Finalmente, referimos o TMS34010 da Texas, por ser o mais versátil processador gráfico disponível.

Incluimos ainda uma referência específica às memórias vídeo, pela nossa convicção de que virão a ser amplamente utilizadas no futuro imediato, e aos conversores d/a mais avançados, mencionando em especial um deles, o Bt451 da Brooktree, que se distingue dos seus congéneres pelas suas capacidades especiais.

4.2 - O controlador gráfico 82786, da Intel

Uma análise das características enunciadas pelo fabricante deste controlador gráfico leva a concluir que se trata de uma arquitectura que deriva directamente da concepção dos controladores gráficos da geração do uPG7220 (do qual o fabricante produz uma segunda fonte), tendo-lhe sido acrescidas características de flexibilidade, velocidade e nível de integração funcional. A arquitectura básica dos subsistemas gráficos não é sensivelmente modificada e, provavelmente, este facto leva a denominar o controlador gráfico 82786 como coprocessador gráfico (18).

A característica de flexibilidade pode ser encontrada na existência de um conjunto mais poderoso de comandos gráficos, na existência de um alto grau de controlo e independência das operações de serialização e da capacidade de arbitragem do acesso à memória, quer do processador local quer do controlador gráfico, que utiliza um processo mais aberto, passível, nomeadamente, de intervenção exterior.

A característica de velocidade é-lhe conferida essencialmente pela sua arquitectura, que permite utilizar uma concepção de várias memórias gráficas separadas, podendo-se efectuar a carga de uma delas enquanto se envia o conteúdo de outra para o ecrã.

A característica de integração deriva do facto de residirem no controlador muitas das funções habitualmente implementadas usando lógica externa, nomeadamente, o controlo de memórias dinâmicas, os dispositivos de interface e arbitragem do acesso aos barramentos do sistema gráfico e a maior parte do interface para a secção de conversão, com suporte das operações comuns de "scroll", "pan" e janela.

4.2.1 - A arquitectura interna do 82786

A arquitectura do 82786 caracteriza-se, essencialmente, pela existência integrada de quatro unidades funcionais: o processador gráfico, o controlador de vídeo, a unidade de interface com os barramentos e o controlador de RAM dinâmica.
Tratando-se de um coprocessador, a arquitectura do 82786 não pode ser apreciada isoladamente mas sim no contexto da sua ligação com o processador principal. Aliás, a unidade de interface com os barramentos é a responsável pelo mapeamento, quer do processador principal, quer do coprocessador, sobre as unidades de memória do sistema.

A figura 4.1 mostra um esquema de princípio da arquitectura global, incluindo o processador principal e o coprocessador, a memória de sistema e a memória gráfica.

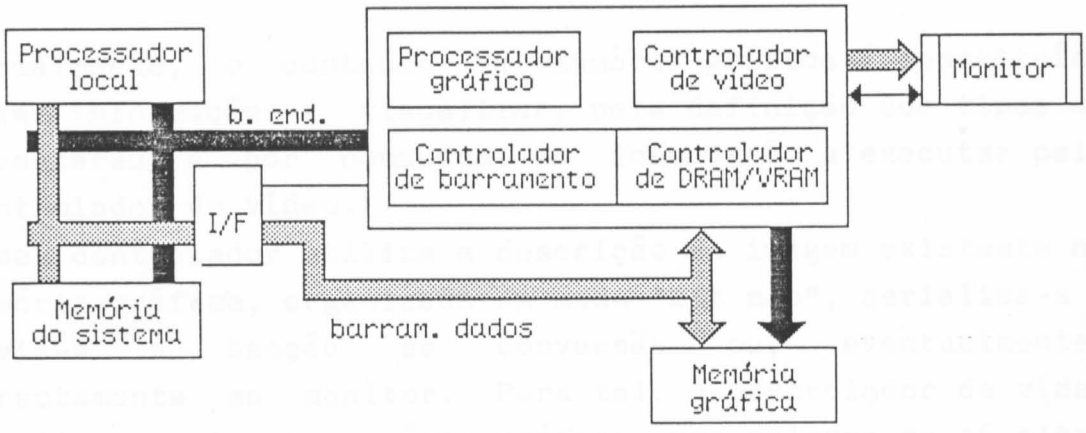


Figura 4.1

O exame do interface do coprocessador com o processador principal mostra que este está concebido fundamentalmente para a utilização do processador 80286 ou 80386 do mesmo fabricante ou, eventualmente, com os processadores desta linha, nomeadamente o 8086 ou 8088, funcionando neste caso em modo síncrono, o que permite optimizar o desempenho do sistema.
Poderá ainda funcionar como coprocessador de famílias de microprocessadores de outros fabricantes, mas em modo assíncrono, sendo de prever uma quebra de desempenho por razões devidas essencialmente ao manejamento dos interfaces dos barramentos.
O processador central comunica com o coprocessador através de um bloco de 128 registos com a organização que se indica na figura 4.2.

Este bloco é dividido em três subblocos de seis registos, o primeiro dizendo respeito ao interface entre o processador central e o coprocessador, o segundo bloco promovendo o interface com o processador gráfico e o terceiro com o controlador de vídeo.

Normalmente, o conteúdo da memória gráfica é constituído pela informação a visualizar, pela definição dos tipos de caracteres e por comandos de formatação a executar pelo controlador de vídeo.
Este controlador utiliza a descrição da imagem existente na memória gráfica, organizada em modo "bit map", serializa-a e envia-a à secção de conversão ou, eventualmente, directamente ao monitor. Para tal, o controlador de vídeo lê, linearmente, na memória gráfica, uma palavra de 16 bits e envia-a, formatada como pixels, para o monitor.

Figura 4.2

O dispositivo de formatação permite definir, de um modo variável, a profundidade utilizada para cada pixel, a qual pode ir de 1 a 8 bits. Obviamente, a formatação tem influência na velocidade de escrita no ecrã, mas este inconveniente é amplamente compensado pela flexibilidade acrescida, nomeadamente em termos de uma aproximação ao

tratamento unificado de gráficos e texto: usando várias janelas sobre um mesmo ecrã, podemos trabalhar com uma dada profundidde de pixel numa das janelas, por exemplo, para escrever texto, usando outra profundidade, noutra janela, para a escrita de imagem.

A construção de janelas é uma característica fundamental deste coprocessador, o qual utiliza uma concepção particular da organização da memória "bit map" para a definição, por hardware, de janelas sobre o ecrã.
Nesta concepção podemos ver a memória gráfica como um conjunto de faixas horizontais, subdivididas em vários blocos, figura 4.3, cada uma com o comprimento fixo de uma linha de ecrã e uma altura variável entre 1 e o número total de linhas de ecrã, ambos os parâmetros sendo definidos aquando da iniciação do sistema.

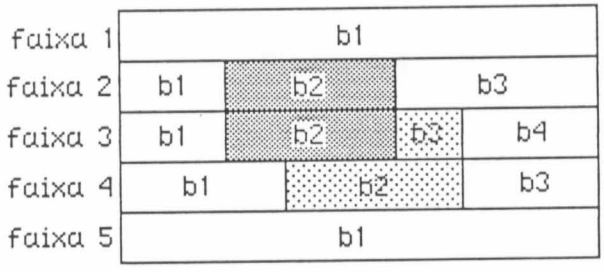


figura 4.3

Para organizar a informação contida na memória gráfica o processador principal constrói uma lista ligada de descritores, com a organização que se mostra na figura 4.4.
O refrescamento do ecrã é obtido por varrimento desta lista de descritores, o primeiro descritor sendo acedido sempre

após o retorno vertical.

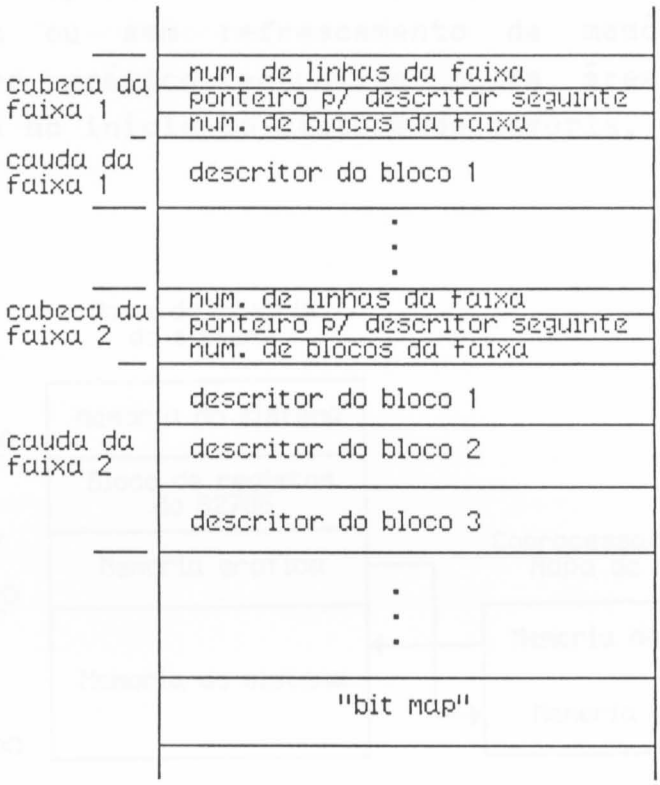


Figura 4.4

Para modificação do aspecto do ecrã, o processador principal apenas necessita de reconfigurar os ponteiros da lista de descritores e, eventualmente, os comandos de formatação. Consegue-se, por este processo, executar com facilidade movimentos sobre a imagem, nomeadamente "scroll" e "pan", sem ter de efectuar actualizações do conteúdo da memória "bit map".
Adicionalmente, é possível a construção simultânea de várias descrições "bit map". De facto, a definição dos endereços actuais dos limites da memória gráfica está contida na

respectiva secção do bloco de registos de interface entre o processador principal e o controlador de vídeo. Assim, este controlador pode executar operações de refrescamento de ecrã, com ou sem refrescamento da memória, enquanto o processador gráfico actualiza outra área de "bit map" localizada no início do sistema de memória, figura 4.5.

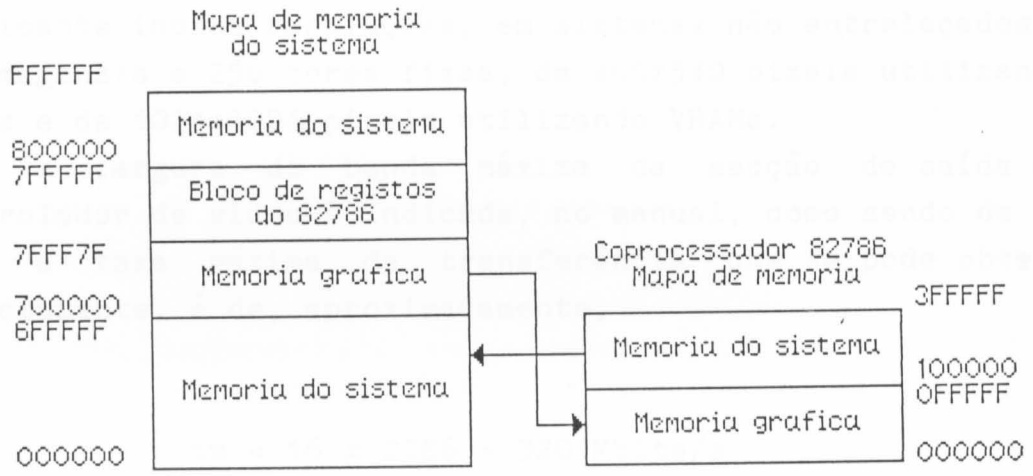


Figura 4.5

Um novo mapeamento da memória gráfica no sistema de memória é obtido por modificação, no bloco de registos, dos respectivos endereços limite.

A unidade de interface do controlador gráfico é constituída pelo controlador de memória dinâmica e pelo interface com os barramentos.
É a este interface que compete arbitrar o acesso, por parte dos vários processadores, ao sistema de memória. Este interface define dois modos de funcionamento: mestre e escravo.

Quando o interface está em modo mestre, o acesso à memória gráfica só é possível por parte do coprocessador, o qual arbitra o acesso quer do processador gráfico quer do controlador de vídeo; quando o interface está em modo escravo, é o processador principal que lhe tem acesso, ficando activo apenas o controlador de DRAM.
Este controlador pode ser usado quer com as memórias dinâmicas usuais, DRAM, quer com as memórias vídeo VRAM. Obviamente que o desempenho obtido é bastante diferente: o fabricante indica resoluções, em sistemas não entrelaçados a 60 imagens/s e 256 cores fixas, de 460x340 pixels utilizando DRAMs e de 1024x1024 pixels utilizando VRAMs.
Como a largura de banda máxima da secção de saída do controlador de vídeo é indicada, no manual, como sendo de 20 MHz, a taxa máxima de transferência que se pode obter, directamente, é de, aproximadamente,

$$tm = 16 \times 20E6 = 320 \text{ Mbits/s}$$

É de esperar que este valor não possa ser atingido por motivos relacionados com o estrangulamento no processo de arbitragem do acesso ao barramento da memória gráfica.

4.2.2 - O firmware do 82786

O conjunto de comandos gráficos disponível neste coprocessador inclui, como habitualmente, a escrita de segmentos de recta, arcos, circunferências e polígonos. Adicionalmente, existem dois comandos de transferência de blocos de dados, quer dentro de um determinado "bit map",

quer de um "bit map" para outro.

Um desses comandos, BitBlT, é dedicado à transferência de blocos de bits, à velocidade de 29 Mbits/s; o outro comando, ChaBlT, transfere caracteres definidos na memória "bit map" à velocidade de 25 Kcaracteres/s.
O comando BitBlT é utilizado, normalmente, para a construção de primitivas gráficas tais como enchimento de figuras e desenho de caracteres gráficos.

O mesmo comando é também usado para a transferência de caracteres de texto. No entanto, tal traduz-se numa sobrecarga do barramento do sistema dada a pequena dimensão dos blocos transferidos em cada comando. Para aumentar o desempenho nesta situação, o coprocessador utiliza um acelerador implementado em hardware, o qual é activado sempre que uma transferência de um bloco de 16x16 bits ou menor é especificada.

Os comandos de escrita de caracteres utilizam tipos pré-definidos carregados em memória pelo processador principal. Os atributos destes comandos podem especificar que a escrita se efectua com espaçamento fixo ou proporcional e, também, que toda a linha deve ser justificada, podendo ainda ser especificada a direcção da escrita ou a rotação a efectuar a cada carácter.

O enchimento de áreas baseia-se num comando específico: o processador principal cria uma tabela com os pontos limites de definição da área a ser cheia e o processador gráfico escreve uma série de linhas horizontais entre aqueles pontos.

Todas as operações gráficas podem especificar um conjunto de atributos, isoladamente ou combinados. Estes atributos podem ser agrupados em seis classes. A primeira executa operações lógicas entre os operandos fonte e destino, nomeadamente o AND, OR, XOR e as suas complementares, e ainda as operações SET e REPLACE; a segunda classe, "bit mask", especifica quais os bits de cada pixel que sofrem a modificação especificada pelo comando; a terceira, "Pattern", especifica o tipo a usar na representação dos segmentos de recta; a quarta, permite definir as cores de escrita e de fundo de caracteres e de segmentos de recta texturados; a quinta, permite definir se o fundo é sólido ou transparente; a sexta, especifica uma janela fora dos limites da qual a escrita não é aparente (clipping).
Adicionalmente, o controlador gráfico pode gerar um pedido de interrupção se se verificar escrita fora dos limites da janela.

O controlador de vídeo também dispõe de um conjunto específico de comandos, que permite carregar um bloco de 40 registos, individualmente ou em conjunto. Estes registos especificam o modo de visualização, nomeadamente, o desenho, a cor e posição do cursor, a cor de fundo e das margens e o factor de "zoom". Adicionalmente, o controlador de vídeo pode especificar ao processador principal a execução de um comando específico, que define a informação a ser enviada para a sua saída durante o tempo de retorno. Esta possibilidade é utilizada para efectuar a carga das paletas de cor sem afectar o funcionamento geral do sistema.

4.3 - O controlador gráfico HD63484, da Hitachi

O controlador HD63484 representa uma evolução dos controladores gráficos da geração anterior, à qual pretende adicionar maior funcionalidade e menores custos de implementação. O fabricante reclama, para este produto, um conjunto de características que lhe conferem essa funcionalidade acrescida:

- Uma linguagem de comandos de alto nível, que inclui 15 comandos de iniciação e parametrização do controlador e 23 comandos gráficos, capacidade de endereçamento X-Y e "clipping" por hardware;

- Controlo eficiente da memória, permitindo resoluções até 4096x4096 para a memória gráfica "bit map" e até 32x256 caracteres para a memória de texto; bancos de memória separados, podendo ir até 2 Mbytes para a memória gráfica e 128 Kbytes para a memória de texto; até três janelas de formatação do ecrã; facilidades de "zoom", "pan" e refrescamento das DRAMs;

- Interface com um processador, optimizado para o diálogo com os microprocessadores da família 68000, utilizando FIFOS separadas para leitura e escrita;

- Interface com o monitor, gerando todos os sinais necessários para sincronização, dispondo ainda de dois cursores em hardware e detecção de caneta luminosa;

- Flexibilidade na definição da arquitectura dos sistemas, permitida pela sincronização mestre-escravo de vários controladores.

Os custos de implementação são diminuídos com a introdução de dois controladores dedicados, o HD63485 (graphics memory interface controller) e o HD63486 (graphics video attribute controller).
A utilização de um 63485 e vários 63486, de acordo com a resolução pretendida, permite obter um elevado nível de integração com a consequente diminuição dos custos de implementação.
Estes dois controladores, juntamente com o controlador gráfico, implementam todas as tarefas necessárias para a gestão do acesso à memória gráfica.

O HD63485 é um dispositivo realizado em tecnologia HI-BiCMOS, que permite uma alta velocidade com baixa dissipação e incorpora todos os circuitos de controlo de barramento e interface com a DRAM; é pois um controlador do barramento do sistema gráfico. Adicionalmente, gera todos os sinais de controlo necessários ao HD63486, o qual é um dispositivo LSI que implementa a secção de serialização: inclui um registo de deslocamento para a conversão paralelo-série dos dados em memória, programada para diferentes comprimentos, o que se traduz em alguma flexibilidade na fixação da arquitectura do sistema. Gera também todos os sinais de controlo e sincronização do monitor, implementando algumas funções correntes de tratamento do ecrã, nomeadamente "zoom" e "pan".

4.3.1 - A arquitectura interna do HD63484

A figura 4.6 mostra um esquema simplificado, de blocos, do controlador.

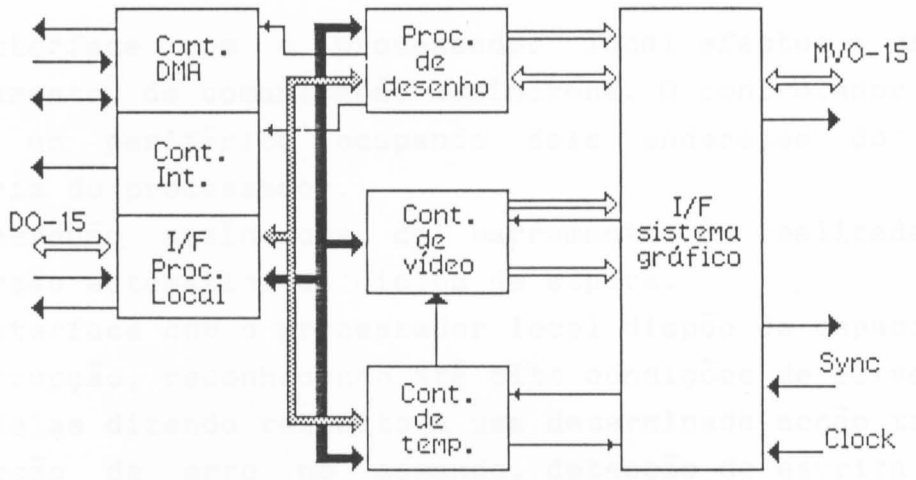


Figura 4.6

A característica mais saliente é a existência de dois barramentos separados, um para o trânsito processador local-controlador e outro para o trânsito controlador-memória gráfica. A existência deste duplo barramento, em contraposição à utilização de um barramento único para todo o sistema, atenua o problema da existência de saturação devida à sua utilização pelo processador local, embora a saturação do barramento gráfico continue dependente da eficiência da arbitragem dos seus acessos.
Permite também configurar o sistema como um grande banco de memória gráfica (até 2 Mbytes para cada controlador gráfico), independentemente do processador local dispor ou não de capacidade de endereçamento desta ordem de grandeza.
A programação do controlador, quer na iniciação, quer no

funcionamento, é feita pela carga de um banco de registos internos, com capacidade superior a 200 bytes, aos quais o processador local (ou central) pode aceder.

O interface com o processador local efectua a gestão do barramento de comunicação assíncrona. O controlador é visto como um periférico ocupando dois endereços do mapa de memória do processador.
A operação assíncrona do barramento é realizada com a inserção automática de ciclos de espera.
O interface com o processador local dispõe de capacidade de interrupção, reconhecendo até oito condições de estado, cada uma delas dizendo respeito a uma determinada acção tal como, detecção de erro no comando, detecção de escrita fora da janela, etc. Cada condição tem associado um bit de estado, mascarável.
É possível ainda implementar transferência DMA, quer usando um controlador externo, quer por software, sob controlo do processador local.

O processador de desenho tem como função interpretar os comandos que lhe são enviados pelo processador local, usando como interface de comunicação dois bancos FIFO de 16 bytes, um para leitura e outro para escrita.
Em resposta a estes comandos, e executando os algoritmos de desenho residentes, determina o conteúdo final da memória utilizando para tal um ciclo de leitura-modificação-escrita. Embora cada palavra processada tenha 16 bits de comprimento, é possível definir, por programação, um formato específico para o funcionamento do controlador, o qual define a profundidade do pixel pelo número de bits sequenciais a considerar. Os valores possíveis são 1, 2, 4, 8 ou 16, o que significa que o sistema gráfico pode manejar directamente até 16 "bit planes".

O controlador de vídeo efectua a gestão da memória gráfica nas actividades de refrescamento de ecrã e de memória.
A memória é dividida numa partição lógica de quatro elementos: três desses elementos correspondem a janelas residentes cujas posições no ecrã e dimensão são programáveis; o quarto elemento, também definido por programação, constitui outra janela que tem prioridade, selectivamente, às anteriores.

Assim, quando esta janela se sobrepõe a alguma das outras é visualizada no ecrã em modo opaco.
É possível, no entanto, programar o controlador para que esta janela seja transparente, aparecendo então os dois conteúdos sobrepostos.
Este controlador é também responsável quer pela gestão dos cursores, implementados em hardware, um para a memória gráfica e outro para a memória de texto, quer pela implementação de "scroll", "pan" e "zoom".

O interface com o monitor efectua a gestão do barramento da memória gráfica, arbitando os ciclos de refrescamento e de escrita na memória, sendo possível estabelecer externamente, por programação, o grau de prioridade de cada um. Gera também os sinais de controlo e temporização do sistema gráfico.

Finalmente, o controlador de temporizações gera todos os sinais de sincronização internos do controlador.

4.3.2 - O firmware do HD63484

Como já referimos anteriormente, o controlador dispõe de um conjunto de 38 comandos. Estes comandos podem ser divididos em três classes, uma incluindo cinco comandos para acesso ao banco de registos internos, outra dez comandos para efectuar transferência de dados e a última 23 comandos para operações gráficas. O formato de cada comando inclui uma cabeça com um "op-code" de 16 bits, seguido de uma cauda contendo uma ou mais palavras que são parâmetros do comando. Estes comandos são enviados pelo processador local para a FIFO ou, directamente, usando o canal DMA, a partir do processador central, se existir.

Os comandos de transferência de dados incluem a possibilidade de leitura, escrita ou modificação do conteúdo das memórias, directamente ou por transferência DMA, sendo a acção de modificação mascarável bit a bit. Permitem também efectuar operações lógicas, nomeadamente, REPLACE, OR, AND e XOR entre os bits de origem e destino.
Outros comandos permitem limpar uma área definível na memória gráfica, ou copiar uma área da memória gráfica para outra área da mesma memória, eventualmente utilizando uma máscara.

Os comandos gráficos têm por função modificar o conteúdo da memória gráfica e são implementados pela execução de algorítmos microcodificados inclusos no processador de desenho. Estes comandos podem especificar endereçamento X-Y, a translação para endereços físicos (lineares) sendo efectuada por hardware.

As primitivas gráficas disponíveis incluem a escrita de segmentos de recta, rectângulos, polígonos, circunferências,

elipses, arcos de circunferência e de elipse, enchimento de uma área rectangular ou de contorno especificável, tracejado, padrões com eventual rotação de 45° e cópia de uma área da memória gráfica para outra diferente, eventualmente com uma rotação de 90°.

4.4 - O controlador gráfico Am95C60 (Quad Pixel Data Flow Manager), da AMD

A característica fundamental deste controlador reside no facto de misturar texto e imagem na mesma memória (gráfica), organizada em "bit map", aproximando-se do modelo unificado de arquitectura gráfica, usufruindo assim das vantagens naturais desta arquitectura, nomeadamente a possibilidade de realizar sobre texto operações idênticas às habitualmente adstritas à escrita gráfica, nomeadamente, rotação, translação, sobreposição, etc.

4.4.1 - A arquitectura interna do Am95C60

A figura 4.7 mostra-nos uma representação simplificada da arquitectura deste controlador.
O elemento principal é um microcontrolador de 16 bits, funcionando a 20 MHz, capaz de dialogar com um processador local ou central através de duas FIFOS, uma funcionando como uma "cache" de instruções e outra como "cache" de dados.
Este microcontrolador é responsável pela execução dos

comandos gráficos que lhe são enviados; para obtenção de um desempenho elevado é dividido em duas secções separadas, endereçamento e dados, as quais comunicam entre si através de um banco de registos específicos.

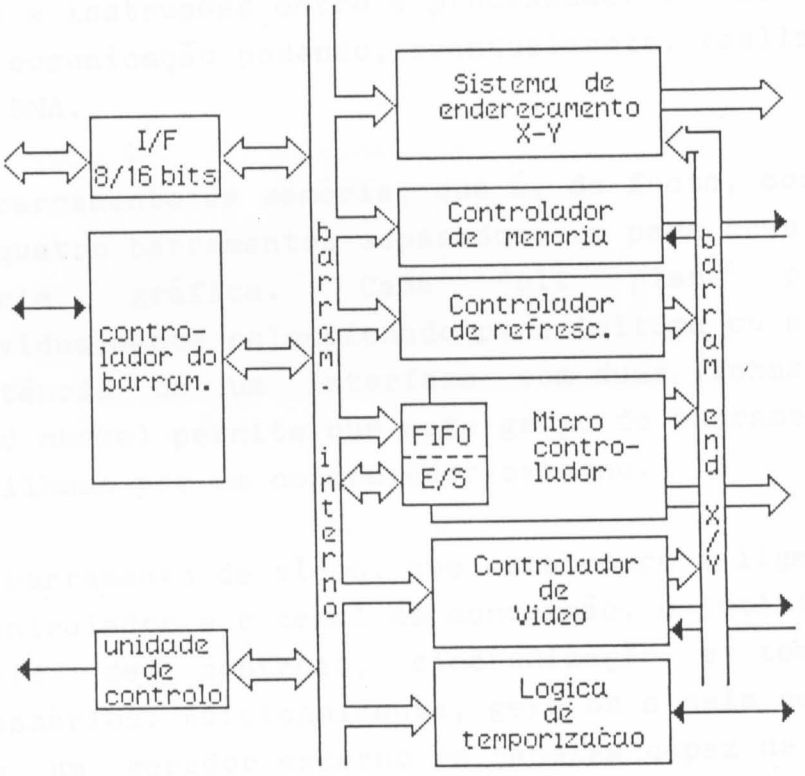


Figura 4.7

Este processador, além de conter os dispositivos correntes dos processadores de uso geral, tais como um acumulador de 16 bits, a Unidade Aritmética e Lógica e um banco de registos de dois portos, dispõe de hardware dedicado à

construção de gráficos.
Outra característica invulgar é a sua ligação com o exterior, a qual se faz por meio de três barramentos diferenciados:

- o barramento do sistema, que liga o controlador a um processador de uso geral e é a sede da transferência de dados e instruções entre o processador e o controlador, esta comunicação podendo, eventualmente, realizar-se em modo DMA.

- O barramento de memória, que é, de facto, constituído por quatro barramentos separados, um para cada plano de memória gráfica. Cada "bit plane" pode ser individualmente seleccionado para leitura ou escrita. A existência de um interface com duas linhas de teste (hand shake) permite que este grupo de barramentos seja partilhado por um controlador externo.

- O barramento de vídeo, que estabelece a ligação entre o controlador e o canal de conversão, e inclui todos os sinais de controlo, sincronização e temporização necessários. Adicionalmente, gera os sinais necessários para um gerador externo de janelas capaz de definir e manejar janelas sobre o ecrã, implementado exclusivamente em hardware.

4.4.2 - O firmware do Am95C60

O conjunto de instruções do controlador assenta sobre três operações primitivas, implementadas usado algoritmos eficientes: geração de linha, cópia de blocos de pixels e enchimento de polígonos.

Sobre estas primitivas é construído um conjunto de comandos, que as invocam de modo transparente para o utilizador, o qual realiza as operações habituais tais como desenho de segmentos de recta, arcos, circunferências e rectângulos, translação de blocos de pixels de uma área para outra da memória gráfica, enchimento de polígonos com uma cor ou padrão, etc, sempre com possibilidade de mascarar o plano escrito ou com realização de operações lógicas.
Além destas actividades triviais, o controlador dispõe de algoritmos especiais para seleccionar uma imagem (pick) e efectuar a sua translação para outro local -uma actividade que coloca sempre dificuldades quando a memória gráfica é do tipo "bit map"-, corte em janela (clipping), "scroll", "pan" e "antialiasing".

O modo de escrita de texto, quase unificado, utiliza um método que deriva do processo tradicional, com algumas modificações.
Para visualizar um carácter o controlador acede a duas zonas diferentes da memória "bit map", existentes numa área não visualizável da memória: uma contém o tipo dos caracteres e a segunda uma tabela de 256 entradas, uma para cada carácter possível. Esta tabela contém também os atributos e parâmetros do carácter, tais como a dimensão, posição na tabela de tipos, cor e formato (dimensão da matriz de caracteres). O controlador, com base nestas indicações, gera os endereços de acesso aos caracteres tal como se de uma operação de transferência de um bloco de pixels se tratasse.

4.5 - O processador gráfico TMS34010, da Texas Instruments

O TMS34010 (Graphics System Processor - GSP) foi o primeiro processador comercializado projectado como um

processador gráfico, isto é, com características gerais, potentes, de programabilidade, tentando obviar alguns dos principais defeitos dos controladores gráficos.
De facto, pela sua constituição, estes controladores dispõem sempre de características de programabilidade reduzidas ou nulas: o TMS34010 é, de certo modo, um controlador gráfico concebido como um microprocessador de 32 bits, com um conjunto de instruções misto, contendo instruções de uso geral e gráficas, capaz de manejar não só palavras segundo o conceito "packed pixels" mas também matrizes de bits ou pixels. Possui também dispositivos eficientes para o traçado incremental de linhas e para a implementação e manejamento de janelas de ecrã.

4.5.1 - O GSP como processador de uso geral

O GSP é um processador de uso geral, no sentido em que dispõe de um conjunto de instruções específico, capaz de executar as operações existentes num processador de uso não dedicado, utilizando o ciclo clássico com extracção de uma instrução, descodificação e controlo da execução.
Tal significa que, ao contrário dos controladores gráficos, o GSP não dispõe de um conjunto pré-fixado de primitivas, implementado em hardware. As primitivas habituais são implementadas em software, e o utilizador é livre de escolher os respectivos algoritmos de acordo com as necessidades da sua aplicação específica. Para obter um desempenho adequado às necessidades do processamento gráfico o GSP utiliza memória "cache" e um conjunto de instruções essencialmente (mas não totalmente) do tipo RISC; o fabricante atribui um desempenho de 6 MIPS, embora não refira quais as operações ou algoritmos usados para este resultado.
Pode também ser programado em linguagem de alto nível, normalmente C, para o que dispõe do respectivo compilador.

4.5.2 - O GSP como processador gráfico

O GSP pode ser categorizado como processador gráfico pela sua capacidade de manipular memória organizada em "bit map", em termos de endereçamento X-Y, de manipular informação a nível de bit ou de blocos de bits, e a de gerar sinais de comando para sincronização do controlador de vídeo.
Relativamente à capacidade de endereçamento X-Y (coordenadas de ecrã), o GSP efectua a sua tradução, por hardware, para endereços reais, (lineares); no que se refere à capacidade de manuseamento de bits, o GSP dispõe das possibilidades habituais de processamento funcional do par bit-origem, bit-destino; além destas possibilidades o GSP dispõe de um conjunto de instruções capaz de efectuar o processamento funcional sobre um bloco rectangular de bits, o que lhe confere a possibilidade de manipular, com facilidade, texto. Possui também capacidade intrínseca de manipulação de janelas.

4.5.3 - A arquitectura interna do GSP

A figura 4.8 mostra o esquema simplificado, de blocos, do GSP. Na sua versão actual usa um sinal de relógio de 50 MHz, o qual é dividido internamente por oito, pelo que o relógio do sistema é de 6,25 MHz.

Neste esquema de blocos podem distinguir-se três zonas fundamentais:

- o bloco de processamento, contendo a unidade aritmética e lógica, o banco de registos de uso geral e o registo de estado;

- o bloco de execução, com uma unidade "cache" para instruções e a máquina de estados de controlo geral do GSP;

- o interface com o sistema de memória;

- o bloco de registos de entrada/saída, responsável pela comunicação com o processador central, se existir, e pela geração dos sinais de sincronismo para o controlador de vídeo.

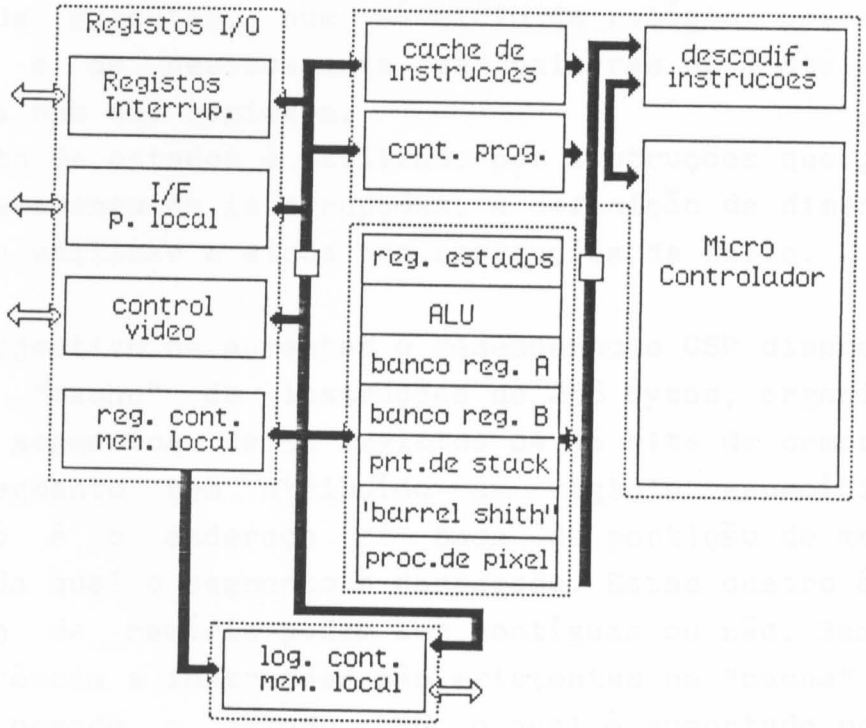


Figura 4.8

O banco de registos é composto por 31 registos de 32 bits de largura, organizados em dois blocos, denominados ficheiros A e B; qualquer destes registos é de uso geral, podendo ser usado para endereçamento ou dados, com excepção do ponteiro de "stack", que é comum para os dois ficheiros e tem o uso reservado habitual.
A Unidade Aritmética e Lógica, é responsável pela execução operativa, utilizando, em geral, conteúdos dos registos dos ficheiros A ou B: operações do tipo registo-memória ou memória-memória são possíveis mas passam, intermediamente, por um registo. As instruções que executam operações aritméticas sobre inteiros incluem a adição e subtracção com ou sem transporte, e a multiplicação e divisão de números com ou sem sinal. Dispõe ainda de oito operações booleanas.
O GSP possui ainda uma unidade funcional adicional, constituída por um registo de deslocamento (barrel shift), capaz de executar, num só ciclo de relógio, operações de rotação e de deslocamento de palavras de 1 até 32 bits, contidas num dos registos.
O registo de estados é utilizado por instruções que permitem o mascaramento de interrupções, a definição da dimensão dos campos a utilizar e ainda por instruções de salto.

Com o objectivo de aumentar o desempenho o GSP dispõe de uma unidade "cache" de instruções de 256 bytes, organizada em quatro segmentos de 32 registos de 16 bits de comprimento. Cada segmento tem atribuído um registo específico cujo conteúdo é o endereço de base da partição de memória a partir da qual o segmento é carregado. Estas quatro áreas da partição de memória podem ser contíguas ou não. Sempre que há referência a instruções não existentes na "cache" a troca faz-se usando o método LRU, o qual é suportado por outro registo específico cujo conteúdo é um ponteiro para o segmento menos recentemente usado.

O controlador de memória "local" tem três funções diferenciadas: controlar os pedidos de acesso do computador central, efectuar o serviço do GSP e gerar os sinais para refrescamento do ecrã; adicionalmente pode gerar sinais para refrescamento das memórias DRAM.
Para tal, além dos sinais habituais de controlo de memória dinâmica, dispõe de sinais para controlo dos "buffers" de isolamento do barramento e de sinais para controlo do controlador de vídeo.
O barramento geral da memória é de 16 bits.

A secção de controlo de vídeo gera os sinais de sincronismo e de apagamento (blanking) do monitor e também o sinal de sincronização do registo de deslocamento das vídeo-RAMs, se usadas. Três registos específicos permitem ao utilizador configurar os tempos de varrimento de acordo com a organização da memória local.
Dispõe ainda de uma entrada de relógio separada do relógio do CPU, de modo a poder-se optimizar separadamente a temporização do controlador de vídeo. Esta temporização pode ser determinada interna ou externamente.

A comunicação com o computador central, se existir, faz-se através de um barramento bidireccional de 16 bits, permitindo a ligação com um elevado número de processadores existentes. Este interface inclui um registo de 32 bits para endereços e outro de 16 bits para dados, multiplexados. Dispõe ainda de um registo separado que é um ponteiro para a memória e de um banco de 28 registos de E/S. Existem ainda dois registos de 8 bits usados como registos de controlo, permitindo ao processador central parar a execução do GSP (sem paragem do controlador da memória local, nomeadamente na geração de sinais de refrescamento da memória), gerar um pedido não mascarável de interrupção e ganhar o controlo do GSP, ou gerar um pedido mascarável de interrupção para enviar mensagens ao GSP.

O GSP dispõe, internamente, de seis barramentos separados, de acesso dedicado, sendo três desses barramentos seccionáveis em dois tramos separados. Esta arquitectura permite um funcionamento em "pipeline" de três secções: enquanto uma instrução contida na "cache" é descodificada e determinada a sua execução pode, separadamente, haver movimento de dados entre registos ou entre registos e unidades funcionais e, simultaneamente, o interface da memória local pode estar a executar ciclos de leitura-modificação-escrita.

4.5.4 - O software do GSP

O GSP possui um conjunto de instruções essencialmente do tipo RISC, no sentido em que a sua arquitectura comporta um banco de registos de uso geral de dimensão apreciável, descodificação directa de instruções, execução das instruções em "pipeline" de pequeno comprimento e operações rápidas do tipo registo a registo. No entanto, tem também algumas instruções do tipo CISC tais como operações aritméticas de 32 bits, transferências complexas de blocos de bits, instruções de movimento de memória a memória, e múltiplos tipos de endereçamento.
O seu conjunto de instruções agrupa 130 instruções em quatro classes, a saber:

- um grupo de instruções de controlo de programa, com salto condicional e incondicional, controlo de "stack", de subrotina e de interrupção;

- um grupo de instruções de movimento de dados entre registos, entre registos e memória ou memória a memória, com endereçamento absoluto ou indirecto;

- um grupo de instruções de uso geral, que inclui operações aritméticas e booleanas, de rotação, deslocamento e troca, de limpeza e de teste de bits;

- um grupo de instruções gráficas que inclui transferência de pixels, isoladamente ou em blocos, escrita de linhas, enchimento de áreas e operações aritméticas sobre endereços X-Y de pixels.

Uma característica importante do GSP reside no facto de endereçar a memória não ao nível do byte ou palavra mas segundo o conceito de campo de comprimento variável, o que torna o endereçamento independente do modo como a memória está organizada. Por esse motivo, todas as operações envolvendo acessos à memória podem ser conceptualizadas como uma extracção ou uma inserção de um campo.
O movimento de dados entre registos é especificado indicando uma ou duas dimensões de campo, (uma para a origem e outra para o destino), podendo, eventualmente, ser extendidas. Esta especificação é contida no registo de estados e é programável separadamente. Quando dados são movidos memória a memória o campo é justificado á direita e extendido em sinal ou a zeros.

O GSP manuseia dois tipos de campos de dados: bits e pixels. O campo de bits pode ter qualquer comprimento entre 1 e 32 e ter início em qualquer ponto da memória; um campo de pixels só pode ter o comprimento de 1, 2, 4, 8, ou 16 bits e deve estar contido dentro dos limites de endereçamento de uma palavra de memória.

Para localizar um campo o processador gera um endereço físico de 26 bits, o qual é extendido a um endereço lógico de 32 bits. No caso de se tratar de um campo de bits, os quatro bits menos significativos apontam o início do campo dentro da palavra no endereço físico correspondente. O comprimento do campo, origem e destino, são palavras de 5 bits definidas, como referimos, no registo de estado.
O interface que efectua este manuseamento de endereços é implementado em microcódigo na unidade de controlo, pelo que o CPU apenas tem de gerar o endereço físico e especificar o número de bits a ser lido ou escrito, após o que fica liberto para executar outra instrução da "cache".

4.5.5 - As instruções gráficas do GSP

Já referimos que o bloco de instruções gráficas suporta tanto endereçamento linear como endereçamento X-Y.
No caso de endereçamento X-Y, um valor de 32 bits é tratado como dois valores separados de 16 bits.
Estes dois valores são convertidos, quando se efectua um acesso a memória, num endereço linear, o qual é função dos valores de X e de Y, e da dimensão da matriz de ecrã (programável pelo utilizador), podendo o resultado ser concatenado com um deslocamento (offset) de 32 bits, o que permite que um endereço X-Y seja relativo.

A instrução de transferência de um bloco de pixels, PixBlT, permite a manipulação de uma matriz de pixels. Esta instrução combina os pixels da matriz fonte com os pixels da matriz destino, segundo uma de várias operações. O alinhamento das duas matrizes, relativamente à organização da memória, pode ser diferente: o GSP efectua o deslocamento

dos dados fonte até obter o alinhamento especificado para o destino.
Também os espaçamentos (pitch) da matriz fonte e destino podem ser diferentes: por exemplo, uma matriz de 11x9, de definição de um carácter, pode ser guardada em memória local (não organizada em "bit map"), como um vector de palavras de 16 bits e ser transferida para a memória "bit map" de acordo com o espaçamento (pitch) definido para o ecrã.
Esta instrução é largamente utilizada na manipulação de texto. Para escrever um carácter o GSP dispõe da sua descrição em memória adicional, não "bit map", e transfere-o para a memória "bit map". Esta transferência pode ser efectuada com um conjunto adicional de facilidades, tais como espaçamento proporcional, justificação ou sobreposição.

As operações possíveis para a instrução PixBlT podem ser agrupadas em quatro classes: operações booleanas e aritméticas, definição de transparência, mascaramento e expansão de um bit a duas cores.
As operações destas quatro classes podem ser combinadas entre si. Por exemplo, ainda na transferência de um carácter, a sua definição existente em memória adicional, não visualizável, pode ser expandida de um bit para uma de duas cores (cor 0 ou cor 1), a qual pode ter 1, 2, 4, 8 ou 16 bits de comprimento. Estas duas cores podem estar definidas em registos específicos, um 0 na fonte correspondendo ao conteúdo de um dos registos e um 1 a outro. O resultado da expansão pode seguidamente ser operacionalmente combinado com o destino, por uma operação booleana ou aritmética, seguidamente mascarada e só os pixels não mascarados escritos na memória "bit map". A figura 4.9 ilustra parte das operações descritas.

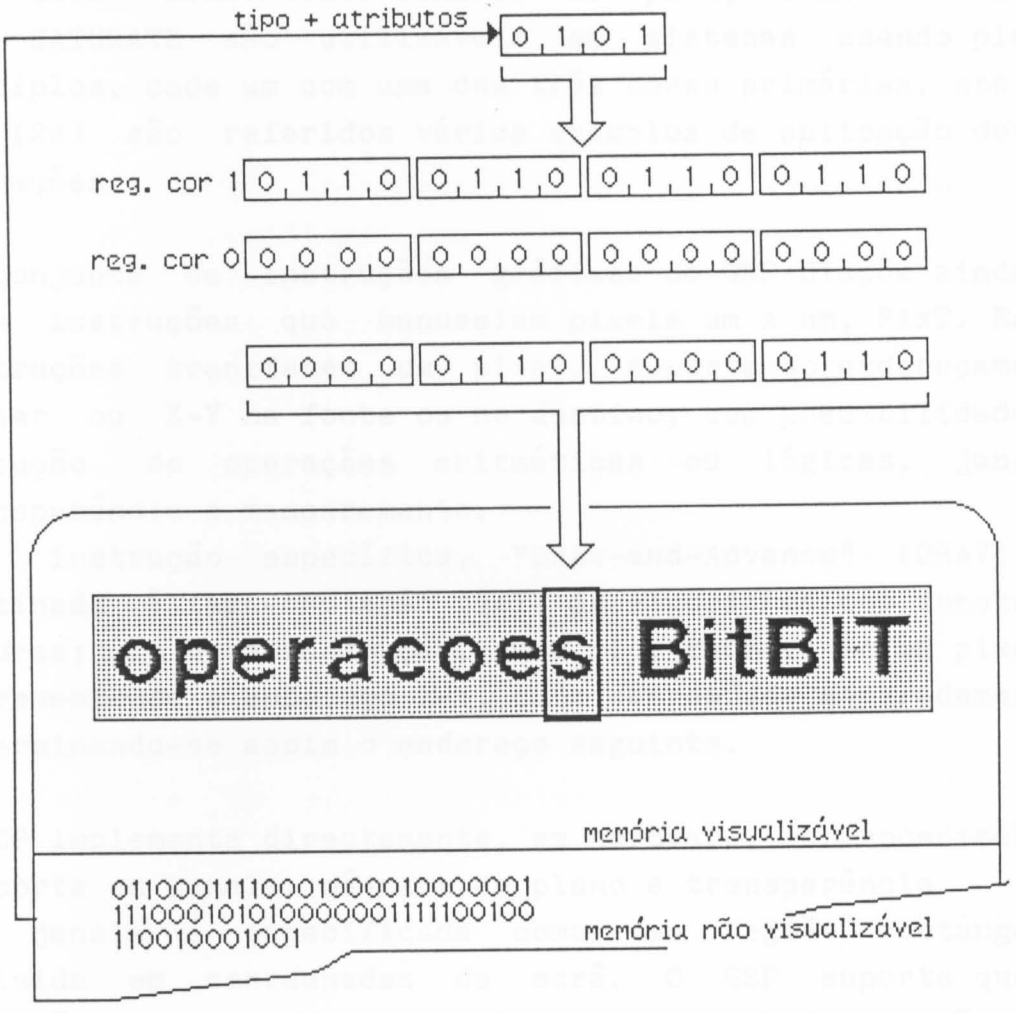


Figura 4.9

As operações booleanas permitidas são 16, incluindo as funções lógicas AND, OR, NOT, XOR e REPLACE entre fonte e destino. Além destas estão definidas mais seis operações aritméticas, a saber, ADD e SUBTRACT com ou sem SATURAÇÃO, MAXIMUM e MINIMUM entre fonte e destino.
Estas operações são importantes para o tratamento da cor, dado que a utilização de funções booleanas sobre pixels, isto é, sobre bits múltiplos em profundidade, pode não ter sentido.

A função MAXIMUM compara pixels da fonte e destino e escolhe o que tiver valor binário maior; a função MINIMUM escolhe o que tiver menor valor binário. As operações ADD e SUBTRACT com SATURATE são utilizáveis em sistemas usando planos múltiplos, cada um com uma das três cores primárias, etc.
Em (24) são referidos vários exemplos de aplicação destas operações.

O conjunto de instruções gráficas do GSP dispõe ainda de seis instruções que manuseiam pixels um a um, PixT. Estas instruções transferem um pixel, suportando endereçamento linear ou X-Y na fonte ou no destino, com possibilidade de execução de operações aritméticas ou lógicas, janela, transparência e mascaramento.
Uma instrução específica, "Draw-and-Advance" (DRAV), é destinada à implementação fácil de algoritmos que constróem figuras; com esta instrução, em ciclo, escreve-se um pixel e incrementa-se o endereço X-Y, cada coordenada separadamente, determinando-se assim o endereço seguinte.

O GSP implementa directamente, em hardware, os procedimentos de corte em janela, máscara de plano e transparência.
Uma janela é especificada como uma região rectangular definida em coordenadas de ecrã. O GSP suporta quatro operações de corte em janela (window clipping): a opção "não corte" ignora os limites da janela, não protegendo a escrita de pixels; a opção "clip-to-fit" impede que pixels sejam escritos fora dos limites da janela; a opção "interrupt on window violation" aborta o funcionamento, gerando uma interrupção se for executada uma operação que utilize pixels fora dos limites da janela; a opção "pick" inibe a escrita de pixels e activa a interrupção por violação quando um pixel fosse escrito nos limites da janela.

Uma instrução específica, "Compare Point to Window", implementada por hardware, permite construir, com facilidade, o algoritmo de Cohen-Southerland de corte de linha (line-clipping).

A facilidade de mascaramento de um plano permite ao utilizador especificar que bits, pertencentes a um pixel, devem ser inibidos para escrita; por exemplo, se os bits do pixel pertencem a um conjunto de planos, a escrita em alguns desses planos pode ser inibida.

A facilidade de selecção de transparência permite tratar os bits de um pixel que tenham o valor zero como transparentes; um bit transparente não é escrito, deixando o bit de destino sem modificação. Esta facilidade permite, de um modo simples, escrever texto sobre gráficos.
As facilidades que acabamos de descrever, janela, máscara e transparência, são utilizáveis com todas as instruções de transferência de pixels, isoladamente (PixT) ou em bloco (PixBlT).

4.6 - Avaliação comparativa dos processadores/controladores gráficos

É sempre difícil comparar as características de um conjunto de processadores na medida em que, regra geral, os fabricantes não publicitam os dados mais importantes dos seus produtos, a menos que lhes sejam extremamente favoráveis, face aos seus concorrentes.
Para estabelecer comparações é pois necessário recorrer a testes efectuados por terceiros, muitas vezes ligados a fabricantes, tendo sempre de haver algum cuidado na

aceitação dos valores expressos. É tanto assim quanto a ausência de normas para avaliação de processadores permite, regra geral, que determinados aspectos, favoráveis para uma determinada marca, sejam medidos num contexto capaz de os fazer sobressair relativamente aos seus congéneres, e que outros, igualmente relevantes, sejam atenuados.

O quadro comparativo que se apresenta na figura 4.10, foi elaborado tendo presente os factos referidos. Como critério para a escolha dos itens a avaliar, utilizámos uma classificação das características em três tipos: o primeiro, dizendo respeito a características gerais desejáveis nestes dispositivos, na linha das que enunciámos no final do capítulo 3; o segundo, tentando dar uma medida do desempenho, contém os parâmetros de velocidade fundamentais; o terceiro, dizendo respeito às características funcionais dos dispositivos, aqui entendidas como o conjunto de funções neles directamente implementadas.

Uma avaliação dos valores expressos parece comprovar algumas afirmações anteriormente feitas.
De facto, o controlador 82786 é o que apresenta um conjunto de características gerais mais modestas, nomeadamente no que se refere à largura do barramento da memória gráfica e à existência de hardware dedicado.
Também no que se refere às características funcionais, importantíssimas para as mais recentes utilizações de sistemas gráficos, nomeadamente gráficos de gestão, desenho e CAD, o conjunto de facilidades disponível é bastante reduzido, nomeadamente no que diz respeito às possibilidades de texto.
Dos outros controladores, o Am95C60 parece ser, em geral, superior ao oferecido pela Hitachi, ao menos à luz dos parâmetros que escolhemos para elaborar este quadro. É um processador ainda não muito referido na literatura especializada, pelo que devemos esperar pela comercialização

	82786	63484	95C60	34010
Tipo de dispositivo	cont.	cont.	cont.	proc.
Implem. primitivas gráf.	hard.	hard.	hard.	hrd+sft
Inst. de uso geral	não	não	não	sim
Inst. gráficas	sim	sim	sim	sim
Grande banco de registos	sim	sim	sim	sim
Largura barram. gráfico	16	19	4x16	32
Hardware dedicado				
"zoom"	não	sim	sim	não
"pan" e "scroll"	não	sim	sim	não
endereçamento X-Y	não	sim	sim	sim
janelas	sim	sim	sim	sim
"clipping"	sim	sim	sim	sim
"pick"	sim	não	sim	sim
"antialiasing"	não	não	não	sim
Texto e gráf. unificados	não	não	sim	não
I/F DRAM incluso	sim	sim	sim	sim
Suporte directo de VRAM	sim	não	sim	sim
Profundidade de pixel	1-8	1-16	1-16	1-24
Velocidade de escrita				
segm. recta Mpixel/s	2,5	1,2-2	2-3,5	1,25
caracteres Kcar/s	25	3,8	60	35
BitBlT Mbit/s	24	2,4-4	80	25
Gráficos				
segm. recta				
1 pixel largura	sim	sim	sim	sim
qualquer largura	não	não	sim	sim
Circunferência	sim	sim	sim	sim
Elipse	não	sim	não	sim
Enchimento de rectângulo				
1 direcção	sim	sim	sim	sim
2 direcções	não	sim		sim
qualquer direcção	não	não		sim
Enchimento de polígono	não	sim	sim	sim
PixBlT				
n. operações booleanas	16	4	5	16
n. operações aritméticas	0	0	0	0
mascaramento	sim	sim	sim	sim
transparência	sim	sim	sim	sim
alinhamento	não	sim	sim	sim
Texto				
"display-list"	hard.	hard.	hard.	soft.
dim. max. do carácter	16x16	256x32	32x32	qualquer
atributos	sim	sim	sim	sim

figura 4.10

dos primeiros sistemas que o utilizem para aceitarmos, com segurança, as características apresentadas.
Quanto ao 34010, um produto da classe dos processadores gráficos, as suas características são francamente melhores do que as que se poderiam esperar tendo em vista os aspectos menos favoráveis imputáveis aos processadores gráficos, já antes referidos. O seu desempenho parece ser superior ao dos controladores 82786 e 63484, sendo apenas suplantado, aparentemente, pelo 95C60.
Parece não ser temerário afirmar-se que, no futuro imediato, os produtos competitivos a ter em conta são o 95C60 e o 34010, os quais irão ser confrontados com um conjunto de novos produtos, cuja saída se encontra anunciada para o corrente ano, nomeadamente o controlador gráfico uPD72120 da NEC, que anuncia velocidade de escrita de 2 Mpixel/s e de BitBlT de 32 Mbit/s, ou o processador gráfico AGCS DP85XX da National, que anuncia velocidade de escrita de 10 Mpixel/s e 160 Mpixel/s em operações de BitBlT e frequências de relógio de pixel até 225 MHz.

4.7 - A memória gráfica

Já vimos anteriormente que a principal questão que se põe relativamente às actividades que envolvem a memória gráfica diz respeito à saturação do seu barramento, partilhado entre o processador ou controlador gráfico, que tem de actualizar o conteúdo da memória, e o controlador de vídeo, que tem igualmente de aceder à memória, lendo, para actualizar o ecrã.
Classicamente, a arbitragem desta contenção no uso do barramento atribuía, em geral, prioridade ao ciclo de refrescamento do ecrã, atrasando o ciclo de actualização da

FACULDADE DE ENGENHARIA
Centro de Informática

memória, o que implica uma quebra acentuada do desempenho global do sistema gráfico.
Um modo simples e corrente de implementar esta arbitragem é efectuar a actualização da memória apenas durante o tempo de retorno do feixe do monitor: com este processo o desempenho do processador não é superior a 30% quando comparado com um funcionamento em que não exista contenção.

Já referimos também alguns processos, hoje clássicos, de aumentar este desempenho. Por exemplo, a utilização de um banco duplo de memória permite elevar o desempenho para cerca de 50%. Posteriormente, outras técnicas foram exploradas: memória de duplo porto, que é uma boa mas cara solução, ou um "buffer" FIFO entre a memória gráfica e o registo de deslocamento da secção de serialização, o que permite independizar a temporização entre as secções de memória e de serialização.
Este sistema evoluiu para as actuais vídeo-RAMs que integram, além do banco de memória normal, o registo de deslocamento e um mecanismo de isolamento entre os dois dispositivos, o qual actua sobre um barramento de interligação com uma largura francamente grande. A organização mais corrente, oferecida pela TMS4161, pela Am90C644 ou pela uPD41264C, e por vários outros fabricantes, inclui um banco de memória dinâmica CMOS de 64Kx4 bits, e um registo de deslocamento de 256x4 bits interligados por um barramento com largura de 1k bits. Tipicamente, a frequência de relógio do registo de deslocamento é de 25 MHz.
Este barramento, tal como os de acesso exterior são, em geral, bidireccionais, o que significa que o fluxo de dados é reversível. Os dados podem ser escritos na memória e transferidos para o registo de deslocamento e vice-versa.
Estes dispositivos são acessíveis por dois portos independentes: durante o funcionamento normal o barramento interno é isolado, podendo os dois dispositivos funcionar independentemente; ciclicamente, é efectuada a carga rápida,

em paralelo, do registo de deslocamento, a partir da memória, e este é o único momento em que o barramento interno não está isolado, não podendo efectuar-se acesso simultâneo a ambos os portos. Com este modo de funcionamento, essencialmente assíncrono, o desempenho atinge cerca de 99%.

As aplicação gráficas requerem, em geral, a modificação, não de todos os pixels acedidos em paralelo pelo controlador, mas apenas de alguns deles, seleccionáveis. Normalmente tal é conseguido usando um ciclo de leitura-modificação-escrita, geralmente lento, dado que necessita, pelo menos, de dois ciclos de acesso para ser realizado. As memórias vídeo actuais resolvem este problema incluindo uma máscara de escrita que impede a modificação de alguns dos bits. A selecção faz-se carregando a máscara num registo específico, usando as próprias linhas do barramento de dados da memória, no início do ciclo de escrita. Esta característica é amplamente utilizada, por exemplo, no caso usual em que cada palavra inclui bits pertencentes a mais do que um pixel.

Uma avaliação dos produtos oferecidos permite concluir que existe ainda uma grande diferença de custos entre os sistemas de memória a DRAM dinâmica, incluindo os respectivos controladores, e as VRAMs. O volume de produção das memórias dinâmicas e o permanente aumento de densidade estão, provavelmente, na base deste facto. É no entanto de esperar que, dado o notável aumento de desempenho que as VRAMs permitem, aliadas à simplificação do projecto que o seu uso permite e à cada vez mais rápida expansão do uso de sistemas gráficos, e ainda devido ao facto de não serem

conhecidos produtos alternativos com estas características, que as memórias vídeo venham a constituir, a muito breve prazo, o componente mais utilizado para a implementação da memória gráfica.

4.8 - Os conversores vídeo d/a

O andar de conversão tem sempre representado um ponto crítico no projecto do sistema gráfico. De facto, se a necessidade de aumento de desempenho tem podido ser satisfeita com a utilização de paralelismo nas secção de memória e serialização, o andar final da secção de conversão obriga, inevitavelmente, a abandonar o paralelismo, tornando-se necessário o processamento de pixels a alta velocidade.
Este problema foi-se agravando à medida que os sistemas gráficos foram evoluindo desde a resolução de 512x512 para as, hoje correntemente oferecidas, resoluções na gama dos 1280x1024 pontos.
Notemos que, como já vimos anteriormente, uma resolução de 1280 pontos equivale a um tempo de pixel da ordem dos 8,5 ns, isto é, o conversor deve poder funcionar na banda dos 120 MHz.
Esta velocidade não é possível com tecnologia bipolar TTL e a solução adoptada foi implementar a entrada da secção em tecnologia ECL. Esta solução apresenta, no entanto, diversos inconvenientes, nomeadamente a utilização de conversores de nível e fontes de alimentação não usuais.

O aperfeiçoamento da tecnologia CMOS, que tem conduzido ultimamente a velocidades bem acima das obtidas com TTL permitiu, no entanto, o aparecimento de solução com

características compatíveis com as necessidades actuais e, inclusivamente, com um elevado nível de integração, permitindo a implementação de toda a secção, compreendendo as paletas de cor, o conversor e lógica dedicada, num único circuito integrado.
Assim, existe actualmente uma razoável gama de produtos que resolvem os problemas postos normalmente por esta secção. O quadro da figura 4.11 dá uma ideia do que é actualmente oferecido.

velocid. (MHz)	resolução (bits)	componentes conv. d/a	componentes paleta	tecnologia	fabricante
300	8	1	não	bipolar	Brooktree
200	4	3	não	bipolar	Honeywell
150	4	3	não	bipolar	Signetics
150	8	1	sim	bipolar	AMD
125	4	3	sim	CMOS	Brooktree
80	4	3	sim	bipolar	Signetics
75	8	1	sim	bipolar	Brooktree
75	4	3	sim	CMOS	Brooktree
50	8	3	não	CMOS	Brooktree
40	4	3	sim	NMOS	TI
40	4	3	sim	CMOS	Intech

Figura 4.11

Os factores para a escolha de um conversor vídeo são, essencialmente, três: largura de banda analógica (ou taxa de refrescamento do ecrã), resolução do conversor (ou número de cores simultâneas), e o grau de integração.
Uma subtil diferença aparece no que se refere ao diálogo entre o conversor e o processador local: geralmente a carga da paleta de cores só se pode efectuar num dos tempos de retorno do feixe; no entanto, em alguns produtos, o banco de memória da paleta de cores é de dois portos, permitindo que a sua carga seja efectuada em qualquer momento.

4.8.1 - O conversor vídeo Bt451, da Brooktree

Trata-se de um circuito em tecnologia CMOS, operando a 125 MHz, composto por três conversores d/a de 4 bits de resolução e paleta de cores de 256x12 bits, duplo porto. A figura 4.12 representa o esquema de blocos deste conversor.

Este conversor dispõe de duas entradas de dados, cada uma delas multiplexadas, podendo uma das entradas, a principal, ser programada para aceitar quatro ou cinco pixels com profundidade de 8 bits, e a outra, entrada de sobreposição, aceitando quatro ou cinco pixels de 2 bits de profundidade.
O conversor dispõe de uma paleta de cores principal de 256x12 bits e outra, de sobreposição, de 4x12 bits. Este canal de sobreposição pode ser usado para apor, no ecrã, mensagens ou cursores.
Adicionalmente, o conversor dispõe de duas máscaras, uma de leitura e outra de cintilação, permitindo o controlo da escrita plano a plano.

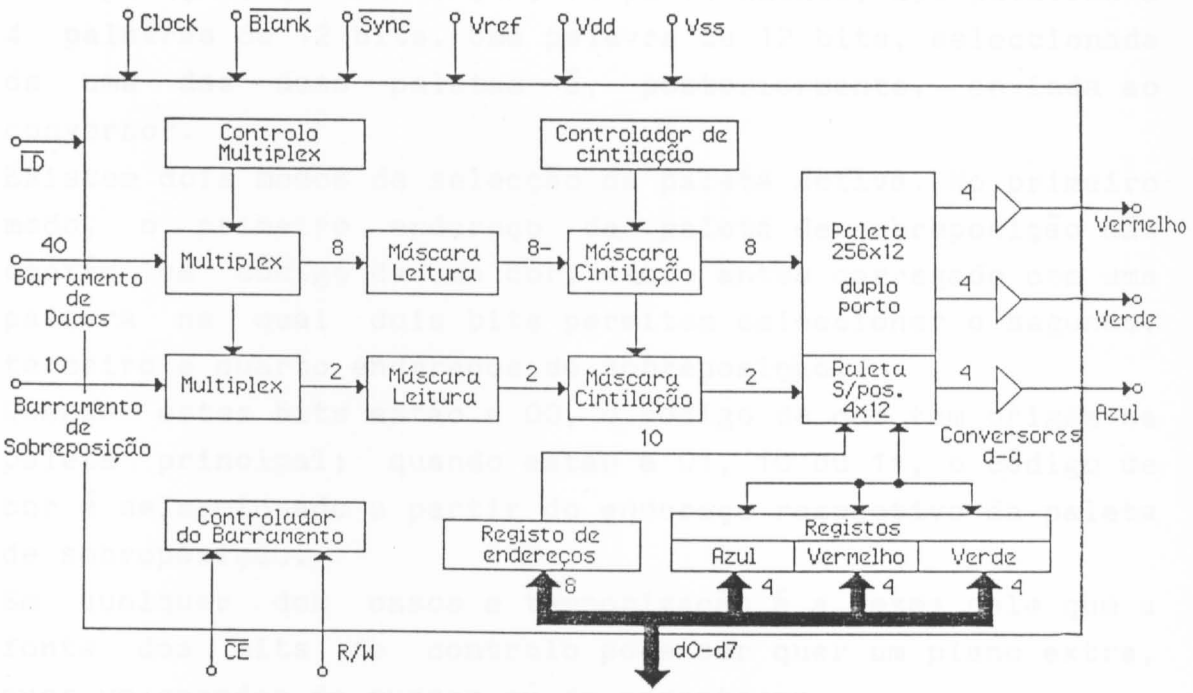


Figura 4.12

A máscara de leitura é constituída pelo conteúdo de um registo carregável externamente, o qual é combinado com a palavra em trânsito (usando a operação booleana AND), determinando assim qual é o plano que endereça a paleta de cor. Isto permite uma acção selectiva dos planos activos. Uma aplicação típica desta facilidade é o desenho em camadas, tal como, por exemplo, no projecto de circuitos impressos, no qual se pode associar cada camada a um "bit plane".
A máscara de cintilação permite obter o efeito de pisca-pisca sobre planos seleccionáveis, a uma frequência que é definida por um contador interno.

A paleta de cores dispõe também de algumas facilidades não usuais.
Como vimos, o conversor permite oito bits de cor e dois de

sobreposição, que endereçam, respectivamente, 256 palavras e 4 palavras de 12 bits. Uma palavra de 12 bits, seleccionada de uma das duas paletas é, posteriormente, enviada ao conversor.
Existem dois modos de selecção da paleta activa. No primeiro modo, o primeiro endereço da paleta de sobreposição não contém um código de uma cor, sendo antes carregado com uma palavra na qual dois bits permitem seleccionar o segundo, terceiro e quarto endereços de sobreposição.
Quando estes bits estão a 00, o código de cor tem origem na paleta principal; quando estão a 01, 10 ou 11, o código de cor é seleccionado a partir do endereço respectivo da paleta de sobreposição.
Em qualquer dos casos a temporização é a mesma pelo que a fonte dos bits de controlo pode ser quer um plano extra, quer um gerador de cursor ou de caracteres.

O segundo modo de funcionamento é activado pela carga de uma "flag" a partir do processador local e suspende o acesso à paleta principal, usando apenas a paleta de sobreposição. Isto permite remover a imagem existente, funcionando o primeiro endereço da paleta de sobreposição como cor de fundo. Este modo pode ser usado em situações nas quais uma mensagem requer atenção especial e imediata.

4.8.2 - Os conversores vídeo DAC3401 da Honeywell e Bt108 da da Brooktree

Estes conversores funcionam, respectivamente, a 200 MHz e 300 Mhz, permitindo resoluções na banda dos 2.000 pontos por linha.

FACULDADE DE ENGENHARIA * Centro de Informática *

Utilizam uma arquitectura clássica, sendo de notar apenas a existência, integrada, de conversores de nível TTL/ECL para CML. Toda a parte analógica é construída em CML (current mode logic), o que permite maximizar a velocidade, aumentar a imunidade ao ruído e diminuir a potência.
Ambos os conversores dispõem de registos de entrada para minimizar a geração de ruído na saída analógica (glitch), descodificadores segmentados, para assegurar monotonicidade e fontes de corrente e de referência altamente estáveis.

CAPÍTULO 5: O desenvolvimento a curto prazo dos sistemas gráficos

5.1 - Introdução

Referimos anteriormente que um dos nossos objectivos era encontrar uma arquitectura "fácil" para os subsistemas gráficos. Estamos agora em condições de precisar o significado deste termo: trata-se de uma arquitectura que satisfaça as características gerais enunciadas no fim do capítulo 3, a saber, uma boa definição funcional, elevado desempenho e alto nível de integração.
Também já referimos o facto de termos assistido durante o ano de 1986 e início de 1987 à produção de dispositivos cujo avanço tecnológico se situa exactamente nesta perspectiva: podemos mesmo afirmar que começa a ser possível construir, com estes dispositivos, o segmento final de um sistema gráfico respondendo satisfatoriamente às necessidades postas pelas aplicações actuais.

Propomo-nos pois efectuar uma primeira avaliação da arquitectura de um tal sistema, usando um conjunto de dispositivos produzidos muito recentemente ou com produção anunciada para breve, sendo já conhecidas as suas características preliminares.

O esboço que fazemos utiliza um traço muito ligeiro, não chegando a ser um projecto preliminar; é antes o apontar da exequibilidade de uma direcção de investigação que poderá ser aprofundada em acções futuras.

5.2 - Uma "nova" classificação dos sistemas gráficos

Já referimos anteriormente que uma avaliação dos produtos correntemente oferecidos permite concluir que os sistemas do topo da gama especificam resoluções da ordem dos 1280x1024 pontos, e até 256 cores, numa paleta de 16,7 milhões (cor total).
Provavelmente, a necessidade do aumento de resolução não se colocará no futuro próximo, essencialmente por duas razões: por um lado, esta resolução é satisfatória e responde às necessidades actuais da generalidade das aplicações gráficas; por outro lado, existem algumas restrições tecnológicas ainda fortemente impeditivas do desenvolvimento de maiores resoluções, restrições ligadas essencialmente à capacidade de construção das máscaras do TRC, à execução do alinhamento entre a máscara e as camadas de deposição de fósforo, e à manutenção da convergência dos feixes electrónicos sobre a máscara.
O estado actual do desenvolvimento tecnológico permite produzir, com custos viáveis, tubos com resolução até 2.000x2.000 pontos; a produção experimental de um tubo para a resolução de 2.448x2.048 foi já efectuada pela Hitachi (16), mas os custos de produção são tão elevados que não é razoável pensar-se que poderão, mesmo a médio prazo, ser comercializados.

Se as questões inerentes à resolução têm uma resposta mais ou menos consensual, já o mesmo não se poderá dizer do que se refere às características funcionais e de desempenho oferecidas pelos vários produtos. Nesta área, o desenvolvimento tecnológico está a efectuar-se aceleradamente, tanto assim que pudemos assistir, de certo modo, ao aparecimento de três gerações de processadores

gráficos no espaço dos últimos quatro ou cinco anos. Adicionalmente, o desenvolvimento da tecnologia na área da implementação VLSI tem feito deslocar as fronteiras entre as várias classes de sistemas, a tal ponto que parece lícito poder afirmar-se que produtos da classe mais elevada, de organização complexa e cara, disponíveis nos anos de 83/84, são hoje implementáveis com dispositivos VLSI capazes de oferecer desempenho e funcionalidade semelhantes.

A par deste desenvolvimento tecnológico assistiu-se ao incremento das áreas de utilização de sistemas gráficos. Não é relevante discutir-se se a expansão da sua utilização deriva do impacto dos avanços tecnológicos ou vice-versa: importa antes reter que nos anos imediatos iremos assistir ao contínuo crescimento e desenvolvimento de aplicações utilizando sistemas gráficos, e que os produtos oferecidos deverão corresponder às necessidades impostas por essas aplicações.

Parece pois útil avaliar que aplicações usando sistemas gráficos estão em utilização ou emergindo, e que necessidades prescrevem.
O quadro da figura 5.1 apresenta o resultado de uma avaliação que ensaiamos.
Inevitavelmente, uma avaliação deste género contém alguma subjectividade, que decorre pelo menos da tentativa de projectar o enunciado dessas necessidades no futuro imediato. É no entanto útil, ao menos como primeira aproximação, para o fim que temos em vista.

Com base nesta avaliação podemos tentar uma (nova) classificação dos sistemas gráficos, do ponto de vista da sua utilização em aplicações.
Nesta classificação incluiríamos na classe mais baixa os dispositivos aptos para a execução de actividades de processamento de texto, com inclusão de figuras simples, de

gráficos para gestão usando um número limitado de cores simultâneas e essencialmente construídos à base de polígonos e circunferências, e de desenho, nomeadamente desenho simples com utilização de segmentos de recta, arcos e enchimento de figuras, usando pelo menos 16 cores simultâneas; na classe média os dispositivos adequados à execução de actividades de composição de texto, combinando o uso simultâneo de texto e gráficos, usando facilidades de modificação da imagem implementadas por "zoom", "pan", "scroll", "clipping", "pick" e "antialiasing"; finalmente, os produtos do topo da gama, capazes da execução de actividades necessitando de grandes "display-lists" e potência de cálculo, tal como a modelação 3D, simulação e síntese de imagem.

activ. / caract.	Proc. Texto	Comp. Texto	Graf. Gest.	Des.	CAD	sim.	mod. 3D	sint. imagem
Inst. Uso Geral	S	S	N	N	S	S	S	S
Inst. Gráficas	N	N	S	S	S	S	S	S
Potência Cálculo	N	N	N	N	S	S	S	S
Facilid. Hard.								
"Zoom"	N	S	N	S	S	S	S	S
"Pan","Scroll"	S	S	S	S	S	S	S	S
Janela	N	S	S	N	S	S	S	S
"Clipping"	N	S	N	N	S	S	S	S
"PicK"	N	S	S	S	S	S	S	S
"Antialiasing"	N	S	N	N	S	S	S	S
Texto e Gráf. Sim.	S	S	N	N	S	S	N	S
Transf.Geom. Texto	N	S	N	N	S	S	S	S
Segm. Recta								
1 Pixel larg.	S	S	S	S	S	S	S	S
qualquer larg.	N	S	S	S	S	S	S	S
Arcos	N	N	S	N	S	S	S	S
Enchim. Polig.	N	S	S	S	S	S	S	S
PixBlT	S	S	S	S	S	S	S	S

Figura 5.1

5.3 - Esboço para o projecto de um sistema gráfico usando uma arquitectura "fácil"

Actualmente, a implementação de produtos da classe inferior é banal, usando os dispositivos VLSI da segunda geração, por exemplo o 82786 da Intel ou o 63484 da Hitachi. O mesmo não acontece com a implementação de produtos da classe intermédia, a qual no entanto é possível com dispositivos da segunda geração, já comercializados ou anunciados.
Quanto aos produtos do alto da gama, as implementações actuais lançam mão de arquitecturas distribuídas utilizando coprocessadores dedicados ou matrizes SIMD, envolvendo organizações adequadas do sistema de memória. Referências interessantes sobre estas arquitecturas poderão ser encontradas em (1), (3), (10), (12) e (22), as quais permitem perspectivar as direcções de desenvolvimento actual relativa a estes produtos.
Parece no entanto legítimo esperar-se que os novos processadores e controladores gráficos, pelos desempenhos elevados que já oferecem, possam vir a ser utilizados em arquitecturas distribuídas que permitam atingir e melhorar os desempenhos actualmente conseguidos, impondo, de certo modo, uma modificação na linha de desenvolvimento que vem sendo seguida.

Tentemos concretizar esta afirmação efectuando uma avaliação ligeira da possibilidade de implementação de um sistema gráfico típico, com resolução de 1280x1024 e 256 cores simultâneas, para texto e gráficos unificados, centrando a nossa atenção na zona do canal gráfico.

A primeira questão que se põe é a escolha do processador gráfico: do estudo que fizemos anteriormente parece evidente

que a nossa atenção se deveria centrar no 34010 ou no 95C60. No entanto, tendo em vista as características desejáveis para esta gama de produtos, indiciada na figura 5.1, nomeadamente a vantagem da utilização de um conjunto de instruções de uso geral, pensaremos na utilização do processador 85XX da National, cuja comercialização total está anunciada ainda para este ano.

5.3.1 - Os dispositivos utilizados

O "Advanced Graphics Chip Set" da National Semicondutor's é de facto uma família de circuitos integrados que, no seu conjunto, constróem um sistema gráfico.
O primeiro elemento da família é o RGP, "raster graphics processor", que tem a função de processador principal do sistema.
É um processador de uso geral, trabalhando a 20 MHz e dispondo de um conjunto de instruções de uso geral do tipo RISC, especialmente projectado para aplicações gráficas. O fabricante anuncia uma velocidade de escrita de 10 Mpixels/s, e operações de BitBlT de 160 Mpixels/s, ambos os valores qualquer que seja o número de planos usado. O RPG dispõe, residentes, de todas as funções especificadas no quadro da figura 4.10, implementadas em hardware.
Adicionalmente, dispõe da capacidade de execução simultânea, em paralelo, de operações de escrita e de "clipping".

A segunda unidade é o DP8510 BPU, "BitBlT Processing Unit", que é um coprocessador funcionando em modo escravo relativamente ao RGP.
Este processador, correndo também a 20 Mz, é responsável

pelas operações de transferência de blocos, efectuando as operações lógicas, de máscara e de alinhamento necessárias. Cada BPU trabalha sobre um dos planos do sistema e cada RGP pode controlar qualquer número de BPUs, embora o máximo possível, sem uso de lógica adicional, seja oito.

O terceiro elemento da família é o DP8512 VCG, "Video Clock Generator", o qual é responsável pela geração de todos os sinais de controlo para o sistema gráfico. É o gerador dos relógios para o RGP e BPU, até 20 MHz, gerando também o sinal de relógio de pixel, cuja frequência máxima poderá ir até 225 MHz.

Finalmente, o DP8515/16 VSR, "Video Shift Register", é capaz de gerar um feixe série de pixels à taxa de relógio máxima.

A escolha deste dispositivo para base deste estudo justifica-se pois, essencialmente, pelo facto de ser um processador gráfico de alta velocidade, dispondo de um coprocessador para execução de operações BitBlT que lhe confere um desempenho excepcional.

5.3.2 - O esquema geral do subsistema gráfico

A figura 5.2 mostra um esquema geral deste sistema gráfico.

Este esquema põe em relevo o alto grau de integração do sistema. Basicamente, é composto por 10 dispositivos da família AGCS, a saber, o VSG, o RGP e 8 BPUs, 64 memórias do tipo TMS4461, construindo a memória gráfica, o controlador de memória dinâmica e o conversor d/a, no total de 76 circuitos integrados.

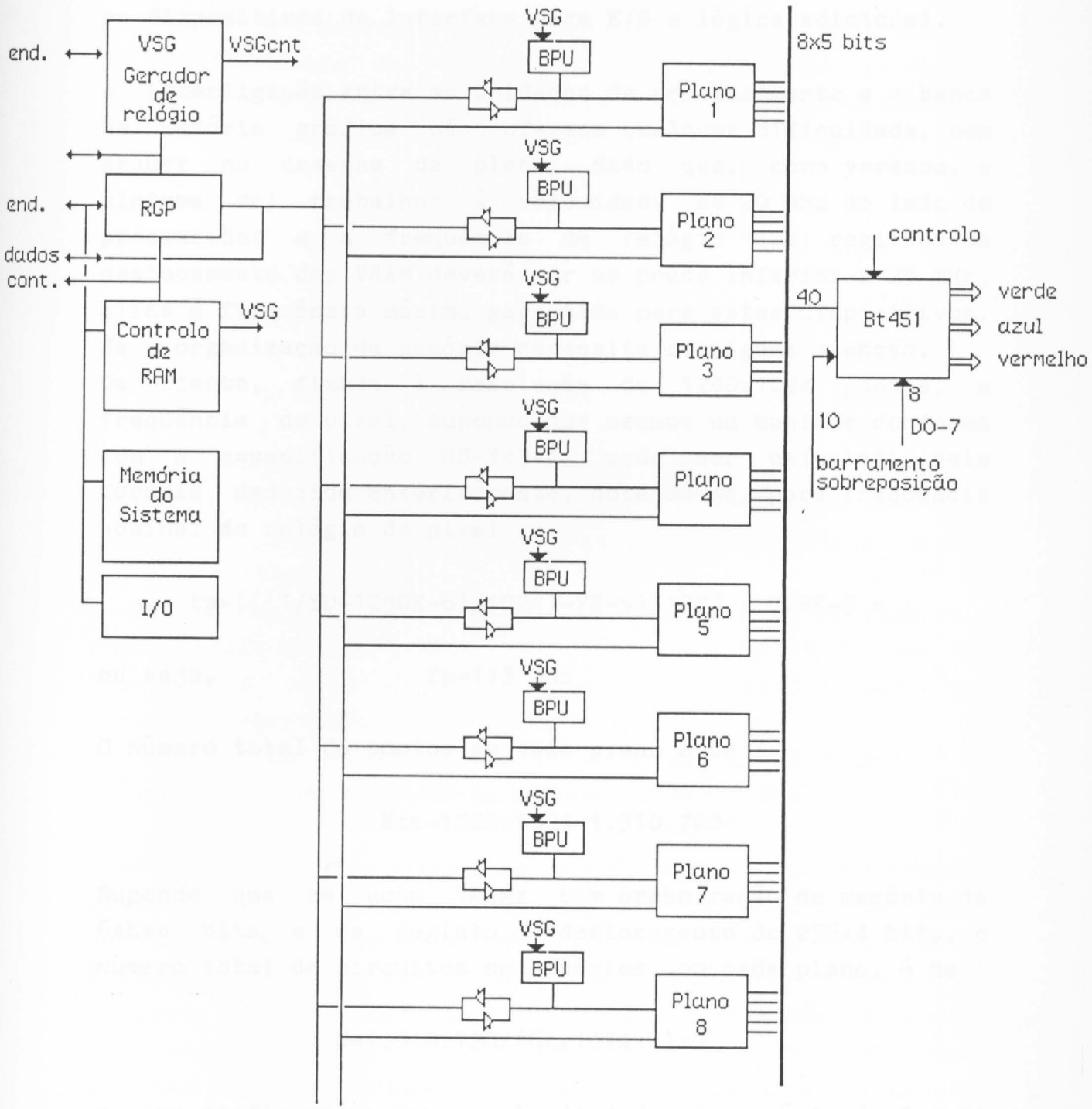

end.
VSG
Gerador de relógio
VSGcnt
end.
RGP
dados
cont.
Controlo de RAM
VSG
Memória do Sistema
I/O
VSG
BPU
Plano 1
VSG
BPU
Plano 2
VSG
BPU
Plano 3
VSG
BPU
Plano 4
VSG
BPU
Plano 5
VSG
BPU
Plano 6
VSG
BPU
Plano 7
VSG
BPU
Plano 8
8x5 bits
controlo
40
Bt451
verde
azul
vermelho
8
10
D0-7
barramento sobreposição

Figura 5.2

Naturalmente, deverá associar-se alguma memória de sistema, os dispositivos de interface para E/S e lógica adicional.

A interligação entre as unidades de processamento e o banco de memória gráfica não oferece qualquer dificuldade, nem sequer no desenho da placa, dado que, como veremos, o sistema vai trabalhar à velocidade de 20 MHz do lado do processador e a frequência de relógio dos registos de deslocamento das VRAM deverá ser um pouco inferior a 25 MHz, aliás a frequência máxima garantida para estes dispositivos.
Já a organização da memória necessita de alguma atenção.
De facto, fixada a resolução de 1280x1024 pontos, a frequência de pixel, supondo que usamos um monitor conforme com a especificação RS-343-A, pode ser calculada pela fórmula deduzida anteriormente, obtendo-se, para frequência nominal do relógio de pixel

$$tp=(((1/50-1250E-6)/1024)-7E-6)/1280 = 8.8E-9 \text{ s}$$

ou seja, $fp=113$ MHz

O número total de pontos em cada plano é de

$$Ntp=1280x1024=1.310.720$$

Supondo que se usam VRAMs com organização de memória de 64Kx4 bits e de registo de deslocamento de 256x4 bits, o número total de circuitos necessários, em cada plano, é de

$$N=1.310.720/(64x1024x4)=5$$

o que configura um barramento de dados da memória de 5x4=20 bits.
Dado que a largura do barramento de dados do processador é de 16 bits, esta incompatibilidade tem de ser resolvida.

A figura 5.3 mostra a organização escolhida. O RGP escreve um pixel usando duas palavras sucessivas, na primeira carregando simultaneamente os bits D0-D3 de profundidade do pixel e a segunda os bits D4-D7. O relógio da VRAM deverá ser ajustado para

Vclock=fp/5=113E6/5=22,6 MHz

Tal significa que em cada 1/22,6E6=44,2 ns é serializado um bit sobre o barramento de pixel. Este barramento tem a largura de 40 bits e, em cada momento, é sede dos bits (1+K) a (5+K), com K tomando valores em (0-64Kx4).
O barramento de pixel, assim organizado, pode ser directamente aposto na entrada do conversor d/a Bt451, o qual deve ser programado para executar a sua multiplexagem segundo a relação 5:1.
A frequência de relógio deste conversor deverá, obviamente, ser ajustada para 113 MHz. Este valor está dentro da gama de frequências do gerador de relógio da família AGCS, pelo que a sua implementação não põe problemas.
A ligação do sistema gráfico a um processador central é trivial, podendo eventualmente usar um canal DMA.

Como é evidente, o sistema gráfico foi especificado tendo em vista o aproveitamento global das possibilidades dos componentes escolhidos. No entanto a configuração resultante é bastante corrente em sistemas desta classe, pelo que a proposta de implementação que fazemos neste exemplo parece ser adequada, quer devido ao aproveitamento total das possibilidades do processador, quer pelo nível de integração do sistema, o qual pode ser implementado sem recorrer a ECL. De notar apenas a restrição imposta pela pequena largura do barramento de dados do processador, obrigando à utilização de duas palavras para a escrita de um pixel, sublinhando a

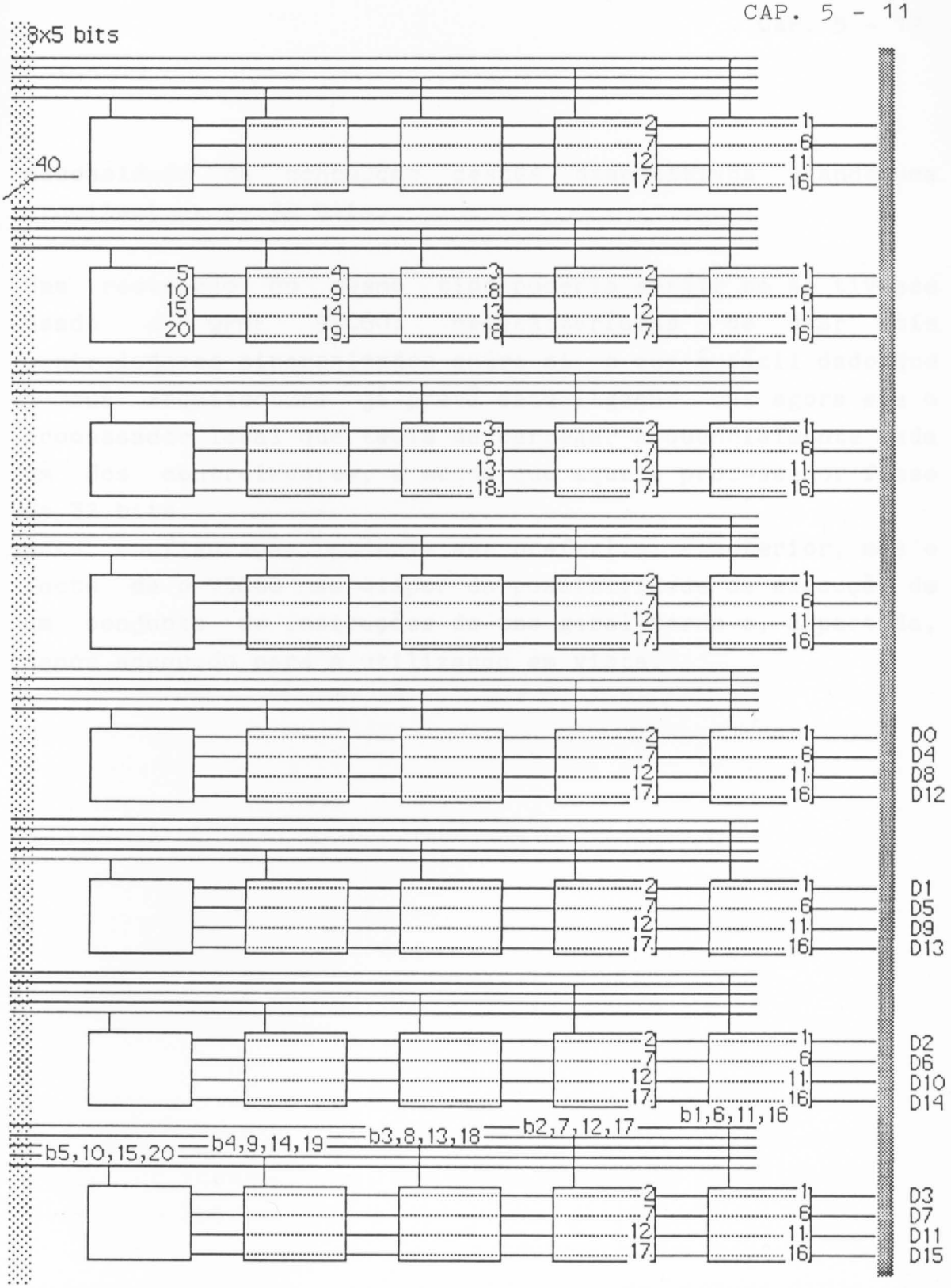

8x5 bits
40
5
10
15
20
4
9
14
19
3
8
13
18
2
7
12
17
1
6
11
16
D0
D4
D8
D12
D1
D5
D9
D13
D2
D6
D10
D14
D3
D7
D11
D15
b1,6,11,16
b2,7,12,17
b3,8,13,18
b4,9,14,19
b5,10,15,20

Figura 5.3

necessidade da concepção destes dispositivos usando uma arquitectura de 32 bits.

Uma restrição do mesmo tipo poderia surgir se se tivesse usado o QPMF 95C60: necessitaríamos de usar dois controladores sincronizados entre si, o que é fácil dado que a sua arquitectura já prevê esta ligação, mas agora era o processador local que teria de carregar sequencialmente cada um dos controladores, a menos que aquele processador fosse de 32 bits.
Esta configuração poderia ser preferível à anterior, mas o facto de o 95C60 não dispor da possibilidade de execução de um conjunto de instruções de uso geral torna-o, à partida, menos adequado para a utilização em vista.

CAPÍTULO 6: Conclusão

Não obstante o início da utilização de meios computacionais gráficos, nomeadamente dos dispositivos de saída, ter as primeiras referências na década de 50, só nos últimos anos se verificou uma acentuada expansão da sua utilização, mercê do interesse manifestado por várias áreas científicas e profissionais.

No decurso da década actual foram-se pois solidificando quer as áreas de utilização dos sistemas gráficos, quer as tecnologias utilizadas. É hoje possível estabelecer três classes de actividades envolvendo estes meios: uma primeira classe, que compreende os sistemas dispondo de um conjunto modesto de facilidades, adequados para a execução de actividades na área do processamento de texto e gráficos simples; a classe intermédia, dispondo dos recursos necessários para o traçado gráfico envolvendo altos níveis de interacção e desempenho, podendo ser utilizados, por exemplo, em actividades de CAD/CAM; e o topo da gama, sendo constituído por sistemas dispondo de potência de cálculo elevada, que é requerida, normalmente, pelas actividades de simulação, modelação 3D e síntese de imagens realistas.

O objecto deste trabalho é, essencialmente, a análise da arquitectura dos sistemas gráficos, no que se refere ao andar de visualização, com especial ênfase nos dispositivos da classe intermédia.
A partir da análise dos produtos oferecidos que devem corresponder, em certa medida, às necessidades dos utilizadores, foi-se estabelecendo a tipologia da arquitectura actualmente mais usada.
Esta arquitectura envolve a utilização de processadores gráficos com um razoável nível de integração, grandes bancos

de memória contendo uma descrição da imagem no ecrã organizada em "bit map", e monitores de varrimento de entrada analógica.

Com base neste suporte foi detalhado um modelo geral para a arquitectura destes sistemas, tendo-se procurado também estabelecer a relação entre o modelo do suporte físico e o modelo de suporte da programação, isto é, processos e estruturas de dados.
Este modelo caracteriza-se pela existencia de três secções funcionando em série (pipeline): a secção de processamento gráfico, a de serialização e a de conversão. Como é óbvio, os dispositivos físicos utilizáveis na implementação do modelo podem reunir funções pertencentes a mais do que uma secção.

As características fundamentais de um sistema gráfico são o grau de interactividade e o grau de realismo das imagens obtidas.
Nos sistemas sobre os quais incide a nossa atenção tal significa, essencialmente, velocidade de processamento e alta resolução de imagem.
A velocidade de processamento obtem-se usando processadores de desempenho (performance) elevado, com um alto grau de paralelismo; a resolução da imagem parece estar, no futuro imediato, satisfatoriamente estabilizada em valores um pouco inferiores a 2.000x2.000 pontos.

Estes factores foram avaliados em duas implementações testadas no âmbito do trabalho desenvolvido nesta área no INESC.
A primeira usava um controlador da primeira geração, inserido numa arquitectura envolvendo elevado paralelismo: o facto de ter sido implementada em SSI e MSI/LSI, isto é, e um pouco impropriamente, implementada em modo discreto, tornou o sistema extremamente pesado, não operacional e

rapidamente obsoleto; a segunda implementação, que recolheu a experiência da anterior e de outras pesquisas, foi implementada com base num processador de uso geral, usando uma arquitectura corrente, estando actualmente a ser construído software que permitirá, a breve prazo, concluir quanto ao seu desempenho.

No ano de 1986 verificou-se um conjunto de desenvolvimentos tecnológicos nos dispositivos gráficos, com um elevado impacto na relação custo/desempenho e acessibilidade dos sistemas gráficos.
Embora estes avanços se tenham verificado quer a nível do hardware quer do software, o motor desse desenvolvimento foram os avanços no hardware, dado que um número elevado de fabricantes anunciou e produziu um conjunto de processadores e controladores gráficos com características tais que configuram uma nova geração de dispositivos gráficos.

Fizemos uma avaliação desses dispositivos. Embora não dispondo ainda de dados decorrentes de uma medida exaustiva das suas características, a impressão geral com que se fica é que, a par do inegável aumento de desempenho que tornam possível, não existe ainda um consenso quanto ao balanço óptimo entre as suas componentes de hardware, firmware e software.
De facto, e como vimos, são oferecidos produtos, com desempenho equivalente, em que o maior peso da implementação se centra no binómio hardware-software, com uma fraca incidência no firmware, e outros em que esse peso é deslocado para o binário hardware-firmware, fazendo diminuir a incidência do software.

É neste balanço entre as várias componentes do sistema que se foca a investigação actual.
Em nossa opinião, a direcção prevalecente deverá apontar para o uso de processadores de "uso geral" , projectados

como processadores gráficos, dispondo de alta velocidade e com arquitecturas do tipo RISC de, pelo menos, 32 bits.
Este processador deverá ser apoiado por coprocessadores realizando, integralmente em hardware, algumas actividades críticas para a obtenção de desempenho elevado.

No penúltimo capítulo deste trabalho ensaiamos o esboço do projecto de um sistema gráfico usando um processador gráfico que, em nossa opinião, ainda que timidamente, dá alguns passos na direcção que preconizamos.
Esperamos os resultados dos ensaios deste dispositivo, que deverá ser colocado à disposição do público ainda este ano, para avaliarmos se o vector de desenvolvimento do hardware dos sistemas gráficos que referimos constituirá ou não, como é nossa presunção, a direcção dominante da investigação a desenvolver no futuro imediato.

Referências Bibliográficas

(1) ARABNIA H.R.,OLIVER M.A.; Fast operations on raster images with SIMD machines architectures;Computer Graphics Forum; 1986.

(2) BORREL J.; Graphics Users gain from vendor's rivalry Mini-MicroSystems; Abril 1985.

(3) CHAROT F.,ROUSEE F.; CSI: A Processor for Image Syntesis; Eurographics'86.

(4) CUNHA A.M.; Modelos de endereçamento no sistema SMD; INESC; Março 1985.

(5) DOORNINK D.J., DALRYMPLE J.C.; The Architectural Evolution of a High-Performance Graphics Terminal; Computer Graphics,IEEEE; Abril 1984.

(6) DRUMM M.J.,HARRIS J.B.,EBERTIN M.; Dual-Port Static RAMs can remedy contention problems; Computer Design; Agosto 1984.

(7) FOLEY J.D., DAM A.; Fundamentals of Interactive Computer Graphics; Addison-Wesley Publishing Co.; 1982.

(8) GULLEY D.N.; Match DRAM organization to memory requirements; Computer Design; Dezembro 1983.

(9) GULLEY D.M.; Joining text and graphics enhances video performance; Computer Design; Agosto 1984.

(10) HINDIN H.J.; High-end Workstations incorporate new technologies; Computer Design; Outubro 1985.

(11) KANKAANPAA A.; An Advanced display with natural interactivity; Eurographics'86.

(12) KAUFMAN A.; Memory Organization for a Cubic Frame Buffer; Eurographics'86.

(13) MACHOVER C., MYERS W.; Interactive Computer Graphics; Computer, IEEE; Outubro 84.

(14) MAY B.A.; VLSI chips combine text and graphics; Computer Design; Março 1984.

(15) MCDONOUGH K.; Graphics ICs provide high-resolution, color CRTs Displays; Computer Design; Janeiro 1986.

(16) MICHAELS I.; Focus on high-resolution color CRTs; Electronics Design; Janeiro 1986.

(17) PRICE S.M.; CMOS 256-kbit video RAM, with two way bus picks up speed, drops power; Electronic Design; Setembro 1985.

(18) RANDALL M., JOARY A.; Graphics coprocessor chip gives small systems a look at the big picture; Electronic Design; Maio 1986.

(19) REED J.S.; Computer Graphics; Mini-MicroSystems; Dezembro 1982.

(20) SLADE S.; Multiple Controllers create high speed color graphics; Computer Design; Julho 1984.

(21) STAUDHAMMER J.; Reaping the benefits of the hardware revolution; Computer Graphics,IEEE; Janeiro 1986.

(22) WELLS C.,KEEN V.; An Application of a Large-Image Memory Graphics Workstation; Computer Graphics,IEEE; Fevereiro 1985.

(23) WHITTON M.C.; Memory Design for Raster Graphics Displays; Computer Graphics,IEEE; Março 1984.

(24) WIENTJES B.,GUTTAG K., ROSKELL D.; First Graphics Processor takes complex orders to run bit-maped displays; Electronics Design; Janeiro 1986.

(25) WILLIAMS T.; Graphics developments geared toward speed and capacity; Computer Design; Dezembro 1986.

BIBLIOTECA F.E. PORTO

FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
BIBLIOTECA
0000022663