

2º Encontro Português de Computação Gráfica

**Porto
Outubro de 1989**

Hardware de Visualização orientado para X Windows

J. Paulo Pereira, INESC.Norte
A. Cardoso Costa, INESC.Norte
F. Nunes Ferreira, DEEC.FEUP/INESC.Norte

ABSTRACT

O "X Window System" é um dos vários sistemas de gestão de janelas disponíveis actualmente, tendo a seu favor o facto de ser suportado por um instituto de renome mundial (MIT) e por alguns fabricantes de primeiro plano (DEC, SUN, etc), de ser um sistema aberto e de permitir a implementação de terminais gráficos compatíveis, com relativa facilidade. Embora não tão "inteligente" quanto o NeWS (enquanto que o X envia comandos gráficos para os terminais, o NeWS envia programas POSTSCRIPT, logo com maior compressão e funcionalidade), tornou-se um standard de facto.

A arquitectura gráfica por nós projectada é muito poderosa e foi orientada para permitir uma implementação fácil e eficiente de um servidor de X. Estes objectivos foram conseguidos através da escolha criteriosa do processador gráfico que melhor servisse os nossos objectivos, o que conduziu a um estudo comparativo de alguns de entre os dispositivos deste tipo comercialmente disponíveis.

1- INTRODUÇÃO

A necessidade de dotar uma estação de trabalho de capacidades de processamento de imagem, nomeadamente captura, emissão, recepção, compressão, descompressão e visualização de imagens dinâmicas de cenas reais e síntese e visualização de imagens estáticas, conduzia ao projecto e posterior implementação de um sistema, designado por processador gráfico/imagem, com a seguinte constituição e características gerais:

- 1- Um processador gráfico, responsável pela síntese de imagens estáticas a cores (256 à escolha entre 16 milhões), de muito alta resolução (1280 x 1024 pixel); a interacção com o utilizador obedecendo ao standard *X Window* criado no *Massachusetts Institute of Technology*;
- 2- Um processador imagem, responsável pela emissão, recepção, compressão e descompressão de imagens dinâmicas a cores de baixa/média resolução;
- 3- Um sistema de captura de imagens a cores (16 milhões), de baixa/média resolução (até 640 x 512 pixel), provenientes de uma câmara ou de outro dispositivo similar;

4- Um sistema de visualização destes dois tipos de imagem, isolada ou simultaneamente, tendo em conta os condicionamentos de ordem temporal impostos pelas imagens dinâmicas, independentemente da sua origem (processador gráfico ou câmara local).

Todo este trabalho está integrado numa tarefa do projecto Estímulo (Estação de Trabalho Integrada Multimedia - imagem, gráficos, voz, texto) apoiado pela Junta Nacional de Investigação Científica e Tecnológica (JNICT).

Esta comunicação começa por fazer uma breve referência ao standard adoptado de interacção com o utilizador: o Sistema *X Window*. A sua estrutura modular e hierárquica vai condicionar a arquitectura do sistema gráfico, a qual deverá resultar de uma solução de compromisso entre a portabilidade e a eficiência requeridas.

A escolha do processador gráfico que melhor servisse os nossos objectivos revelou-se de importância fundamental e conduziu a um estudo pormenorizado de alguns dispositivos comercialmente disponíveis. As conclusões obtidas encontram-se aqui resumidas.

Segue-se a descrição da arquitectura do sistema gráfico, com base num diagrama de blocos simplificado. Desta, dois elementos merecem especial destaque pela sua originalidade: o canal de vídeo e o controlador da memória de visualização de imagem.

Para concluir, serão referidos pormenores de implementação prática do *hardware* já projectado, problemas relacionados com o projecto e a implementação do restante *hardware* e o modo como irá ser desenvolvido o *software* para este sistema.

2- O SISTEMA X WINDOW

O Sistema *X Window* é um standard de interacção entre o sistema gráfico e o seu utilizador [1]. De um modo geral, caracteriza-se pelo uso intensivo de janelas e de um dispositivo apontador (*mouse*, por exemplo). À semelhança do que se verifica, por exemplo, nos modernos sistemas operativos, este standard obedece a uma estrutura largamente modularizada e hierarquizada. Desta forma, cada módulo pode ser implementado da maneira mais conveniente, independentemente dos módulos que o antecedem e lhe sucedem na escala hierárquica, fazendo uso dos recursos postos à sua disposição pelo módulo que o antecede e, por sua vez, prestando serviços de complexidade crescente ao módulo que lhe sucede.

As vantagens duma estrutura deste tipo são óbvias:

- Torna o sistema altamente portátil, uma vez que todos os módulos, à excepção do primeiro, designado por DDX (*Device Dependent X*), são independentes do suporte físico (*hardware*) que controlam;
- Facilita eventuais evoluções do sistema, através da simples alteração de um ou mais módulos que o constituem, sem prejuízo dos módulos que permanecem inalterados.

Em contrapartida, uma estrutura modular e hierarquizada tende a ser, de um modo geral, lenta na execução das tarefas que lhe são atribuídas, o que se reflecte directamente no desempenho do sistema gráfico em consideração.

O código fonte do servidor do Sistema *X Window* que vai ser utilizado é o código da distribuição oficial do MIT. Dado que este código pretende abarcar todas as cartas gráficas *bitmapped* cuja memória gráfica possa ser escrita/lida aleatoriamente (desde as mais simples, sem qualquer dispositivo gráfico, até às mais evoluídas, como é o caso), isso implica que as rotinas mais elementares sejam pouco sofisticadas. As entidades manipuladas por estas rotinas são o *pixel* e o *span* (linha de pixels), que são demasiado elementares para a maioria dos dispositivos gráficos actuais, pois estes conseguem manipular eficientemente entidades mais complexas, tais como linhas, blocos rectangulares de pixels, arcos, etc.

A 1ª fase da implementação do servidor vai consistir na adaptação destas rotinas elementares, de modo a ter uma versão funcional o mais rapidamente possível. Uma outra vantagem consiste na facilidade de teste e na possibilidade de se poderem detectar, sem grandes dificuldades, problemas, pontos de congestão, etc. É natural que o desempenho desta versão não seja muito elevado, em relação aos recursos de *hardware* disponíveis.

A 2ª fase consistirá naturalmente em substituir/complementar a funcionalidade demasiado simples das rotinas elementares por/com outras rotinas, conhecedoras do *hardware* existente, e que tenham um desempenho elevado. As áreas mais susceptíveis de investigação são as primitivas gráficas (linha, circunferência, arco, texto), transferência de blocos de pixels e janelas. A documentação oficial [1, 11, 12, 13] faz sugestões sobre aqueles aspectos que considera mais críticos, e onde o aumento do desempenho pode ser muito importante. É de realçar que a funcionalidade do Sistema *X Window* é demasiado complexa para ser realizada totalmente em *hardware*, pelo terão que coexistir rotinas especializadas e rápidas com rotinas generalizadas e lentas, o compromisso entre ambas dependendo de muitos factores.

Uma outra área importante de investigação relaciona-se com a compatibilização do sistema de gestão de janelas com a janela de visualização de imagens vídeo captadas localmente ou de teleconferência, e que é uma característica nova mesmo nas *workstations* mais avançadas. Dado que a gestão desta janela especializada vai ser quase totalmente realizada por *hardware*, resta

somente o problema de informar o sistema de gestão de janelas, de modo a conseguir o sistema num estado permanentemente equilibrado.

3- ESCOLHA DE UMA ARQUITECTURA

Para usufruir, por um lado, de todas as potencialidades postas à disposição pelo X e, simultaneamente, dispor de um sistema altamente eficiente, torna-se necessário sacrificar, em parte, a portabilidade já referida e proceder à implementação de um sistema gráfico com características específicas, dotado de *hardware* especializado e capaz de prestar directamente os serviços requeridos pelos módulos de nível mais baixo do X, nomeadamente o DDX, o *mi* (*machine independent*) e o *cfb* (*color frame buffer*).

No estado actual da tecnologia, um sistema gráfico que satisfaça estes requisitos deverá incluir, obrigatoriamente, capacidades de processamento local, bem como funções gráficas específicas, tais como operações de *filling*, *pixblt* (*Pixel Block Transfer*), *clipping* e desenho de texto e de linhas rectas. Para além destas funções, o sistema deverá ser capaz de proceder eficientemente a operações aritméticas em vírgula flutuante, requeridas pelo nível *mi* no desenho de, por exemplo, linhas rectas de espessura arbitrária. A implementação de cursores pelo *hardware* do sistema também é importante pois, apesar do X permitir cursores de dimensões arbitrárias que estão fora do alcance de qualquer circuito, a geração automática de cursores mais simples (e mais frequentemente utilizados), resolve imediatamente um conjunto de problemas que, de outro modo, consumiriam muito tempo de CPU (problemas relacionados com o desenho e a remoção periódicos do cursor, bem como à necessidade de salvar e restaurar a informação situada sob o mesmo).

4- O PROCESSADOR GRAFICO

Uma vez assente que o sistema gráfico deverá incluir capacidades de processamento local, bem como de execução de funções gráficas específicas, resta decidir sobre que dispositivo, de entre os comercialmente disponíveis, irá incidir a nossa escolha.

Este problema conduziu-nos ao estudo comparativo de três dispositivos gráficos [2, 3], mais concretamente:

Intel 82786 Graphics Coprocessor;

Texas Instruments TMS34010 Graphics System Processor;

National DP85XX Advanced Graphics Chip Set.

As conclusões obtidas foram, resumidamente, as seguintes:

- 1- O coprocessador gráfico 82786, apesar de apresentar algumas potencialidades muito interessantes não compartilhadas pelos outros dispositivos, está limitado a resoluções (640 x 480 pixel) que estão muito abaixo da pretendida;
- 2- O processador gráfico TMS34010 é um dispositivo muito completo, que integra num único circuito integrado quase todas as funções dedicadas acima referidas; como contrapartida, a arquitectura da memória de visualização imposta por este dispositivo (*packed-pixel architecture*) limita o seu desempenho em sistemas caracterizados por um elevado número de bits por pixel;
- 3- A família de dispositivos gráficos DP85XX, ao contrário do dispositivo anterior, está orientada para uma arquitectura da memória de visualização do tipo *planar architecture*, o que torna a sua eficiência independente do número de bits por pixel pretendido; como contrapartida, o custo e o espaço físico ocupado por esta família são deveras significativos.

Pouco tempo após a conclusão deste estudo, surgiu nos meios de comunicação especializados a notícia da comercialização de um trio de dispositivos gráficos com características muito interessantes. Tratava-se de um processador gráfico de 32 bits (TMS34020) [4], um coprocessador de vírgula flutuante dotado de primitivas sofisticadas de manipulação de gráficos 2-D e 3-D (TMS34082) [4] e uma memória VRAM dotada de mecanismos especiais de *filling* e de desenho de texto (TMS4C251) [5]. O recurso a um *bus* externo de 32 bits, bem como a ciclos de acesso à memória do tipo *enhanced page-mode memory cycle*, implicava que o TMS34020 fosse cerca de seis vezes mais rápido que o seu predecessor, o TMS34010. Além disso, o desempenho na execução de operações de vírgula flutuante era assegurado pelo TMS34082, capaz de efectuar 40 milhões de operações por segundo. Por fim, a memória VRAM TMS44C251 dava mais um passo no sentido das memórias inteligentes, ao incluir um *color latch* que acelerava apreciavelmente as operações de *filling* e de desenho de texto.

Além destas potencialidades, era ainda de salientar o facto de o conjunto de instruções deste processador incorporar o do TMS34010, havendo já da nossa parte uma experiência bastante significativa, adquirida em projectos anteriores, na elaboração de *software* em linguagem *assembly* para este último [6, 7, 8].

Todos estes argumentos contribuíram, assim, para que a escolha incidisse sobre este novo trio de dispositivos gráficos.

5- DESCRIÇÃO DA ARQUITECTURA

A figura 5.1 ilustra um diagrama de blocos simplificado da arquitectura do sistema gráfico. Os blocos que não têm nome representam simplesmente *hardware* de interligação dos restantes elementos do sistema.

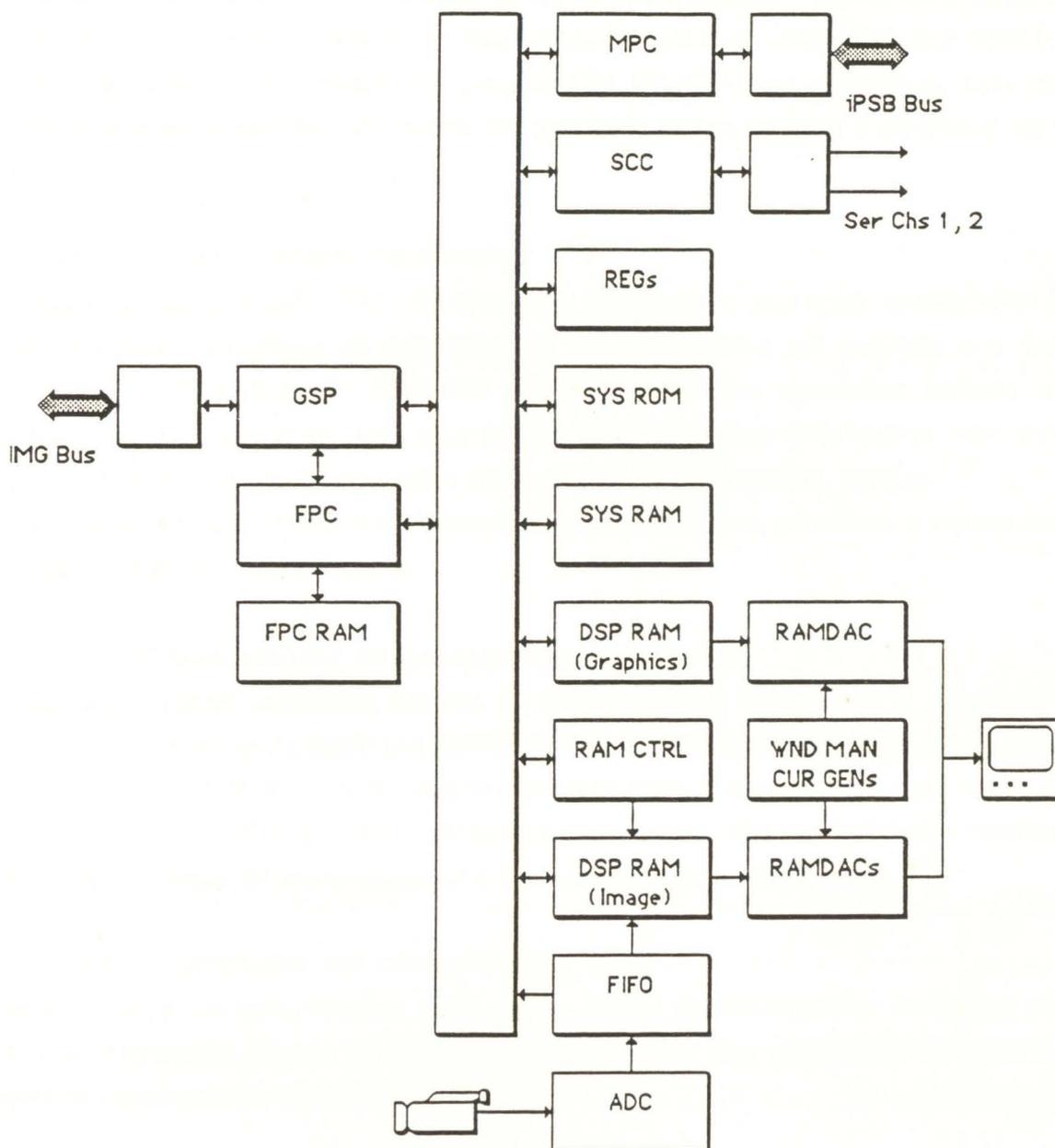


Figura 5.1 - Arquitectura do sistema gráfico

5.1- Processador gráfico

O processador gráfico (GSP - *Graphics System Processor*) é responsável pelo controlo de todo o sistema e, também, pela comunicação com o processador imagem, a implementar futuramente. Essa comunicação será feita através do denominado *Image Bus* (IMG Bus), controlado pelo processador imagem, o qual deverá ser capaz de transferir informação a velocidades elevadas. De momento, e até que o *hardware* correspondente ao processador imagem tenha sido projectado e as características do *Image Bus* especificadas, existe um *interface* que permite ligar o sistema ao *bus* de um computador pessoal IBM PC-XT/AT ou compatível. Esta ligação é temporária e serve apenas para facilitar as operações iniciais de teste e de *debug* do sistema [4].

5.2- Coprocessador aritmético

O coprocessador aritmético (FPC - *Floating Point Coprocessor*) está ligado directamente ao GSP. Pode efectuar 32 milhões de operações de vírgula flutuante por segundo e é totalmente compatível com o standard IEEE-754 de 1985. Além das operações comuns a outros coprocessadores aritméticos, este dispositivo efectua operações matemáticas mais complexas, tais como convolução 3 x 3, operações sobre matrizes 4 x 4 e *splines* cúbicas.

O FPC pode ainda ser ligado a RAM estática de alta velocidade, permitindo a implementação de outras funções em microcódigo [4].

5.3- Coprocessador de passagem de mensagens

O hardware gráfico projectado assenta no bus de sistema Multibus II. O coprocessador de passagem de mensagens deste bus (MPC - *Message Passing Coprocessor*) é responsável pelo processamento completo das mensagens trocadas entre o processador local (GSP) e outros sistemas ligados ao *IPSB Bus*. Além da troca de mensagens, estão implementados mecanismos de acesso aos espaços de memória, de I/O e de interligação destes sistemas [9, 10].

5.4- Controlador de comunicações série

Um controlador de comunicações série (SCC - *Serial Communications Controller*) permite a troca de informação, através de dois canais RS232C, com uma vasta gama de dispositivos tais como, por exemplo, um teclado e um *mouse*.

5.5- Memória

No que respeita aos recursos de memória, podemos distinguir os seguintes blocos:

- ROM do sistema (SYS ROM), que serve de suporte inicial ao armazenamento do código necessário ao funcionamento do mesmo;

- RAM do sistema (SYS RAM), para onde é transferido o código residente na ROM, de modo a aumentar o desempenho da sua execução; para além do código, é nesta memória que é armazenada a informação criada pelo Sistema *X Window*;
- VRAM de visualização de gráficos (DSP RAM - Graphics), na qual é possível armazenar a informação correspondente a uma imagem estática de 1280 x 1024 pixel, 8 bit por pixel;
- VRAM de visualização de imagem (DSP RAM - Image), na qual é possível armazenar a informação correspondente a duas imagens dinâmicas de 640 x 512 pixel, 24 bit por pixel, ou uma imagem estática de 1024 x 1024 pixel, 24 bit por pixel.

5.6- Tabelas de cor e conversores digitais-analógicos

As tabelas de cor e os conversores digitais-analógicos (RAMDACs) recebem a informação proveniente das memórias de visualização (gráficos e imagem) e procedem à descodificação da cor e à conversão do sinal digital num sinal analógico de características compatíveis com a generalidade dos monitores comercialmente disponíveis.

5.7- Gestor de janelas

O gestor de janelas (WND MAN) é responsável pela geração da janela correspondente à imagem dinâmica, permitindo torná-la ou não visível e alterar as suas dimensões e localização.

5.8- Geradores de cursores

Os geradores de cursores (CUR GENs) são responsáveis pela geração de dois cursores, sendo um para a(s) janela(s) de gráficos e o outro para a janela de imagem. Cada cursor pode ter uma de duas configurações à escolha: um bloco de 64 x 64 pixel definido pelo utilizador ou uma cruz de dimensões arbitrárias.

5.9- Controlador da memória de visualização de imagem

O controlador da memória de visualização da imagem dinâmica ainda não está implementado e terá as seguintes funções:

- Geração dos ciclos de armazenamento da informação proveniente do conversor analógico-digital (ADC) via memória FIFO (*First-In First-Out Memory*); estes ciclos de acesso à memória terão de ser do tipo *enhanced page-mode memory cycle*, de modo a satisfazer as exigências de ordem temporal impostas pelo carácter dinâmico da imagem a visualizar;
- Geração dos ciclos de refrescamento da memória dinâmica;
- Geração dos ciclos de transferência da informação para os *shift-registers* internos da referida memória;
- Geração de endereços e controlo da frequência de visualização, de modo a permitir a ampliação dinâmica da imagem quer na vertical, quer na horizontal.

O controlo das saídas analógicas e, portanto, a decisão em cada instante da fonte de informação (gráficos ou imagem) que está a ser visualizada, ficará a cargo do gestor de janelas.

Ainda no âmbito do canal de vídeo, é de referir o *hardware* de controlo que permite que o GSP esteja configurado, à escolha do utilizador, num dos seguintes modos de funcionamento:

- O modo normal, no qual o GSP gera um ciclo de transferência de informação para os *shift-registers* das VRAMs durante o período correspondente ao retorno horizontal do feixe electrónico; este modo, semelhante ao existente no TMS34010, obriga a reservar, para cada linha de vídeo, uma quantidade de memória de dimensão em pixels igual a uma potência de base dois;
- Um modo especial, que faz uso de uma característica exclusiva das novas VRAMs (TMS44C251) e que consiste na partição dos *shift-registers* das mesmas em duas metades independentes: enquanto uma metade está a fornecer informação ao canal de vídeo, a outra metade está disponível para, em qualquer momento, ser carregada com nova informação proveniente da memória; este tipo especial de ciclo de transferência é denominado *split shift-register transfer cycle* e tem a vantagem de permitir que a dimensão da memória necessária para armazenar uma linha de vídeo seja arbitrária.

7- O CONTROLADOR DA MEMÓRIA DE VISUALIZAÇÃO DE IMAGEM

A memória de visualização da imagem dinâmica pode ser controlada por duas entidades distintas, GSP ou controlador dedicado de memória.

O primeiro caso aplica-se a situações em que o processador imagem pretende ter acesso à imagem armazenada nesta memória. O acesso é feito através do *Image Bus* e do GSP. Também permite que o utilizador disponha de uma janela especial de gráficos, com a dimensão máxima de 1024 x 1024 pixel, 24 bit por pixel.

O segundo caso aplica-se a situações de captura de imagens dinâmicas provenientes de uma câmara ou outro dispositivo similar. A entrada de informação faz-se pelo canal de acesso aleatório das VRAMs, enquanto que a saída se processa pelo canal série da referida memória. Torna-se, portanto, necessário gerar os seguintes ciclos de funcionamento:

- *Enhanced page-mode write cycle*;
- *Memory to shift-register transfer cycle*;
- *CAS-before-RAS refresh cycle*.

Cabe ao controlador da memória gerar os endereços e os sinais de controlo correspondentes a estes três tipos de ciclos.

O endereço gerado quando da execução de um ciclo de escrita deverá contemplar as seguintes possibilidades:

- Possibilidade de o dispositivo de entrada fornecer uma imagem entrelaçada ou não entrelaçada;
- Possibilidade de armazenar a imagem num de dois locais distintos da memória, de modo a tornar viável a visualização de uma imagem e, simultaneamente, a captura da imagem seguinte.

O endereço gerado quando da execução de um ciclo de transferência deverá poder ser repetido um número arbitrário de vezes, de modo a permitir a ampliação dinâmica da imagem na direcção vertical.

A frequência de vídeo deverá poder ser dividida por um factor arbitrário, de modo a permitir a ampliação dinâmica da imagem na direcção horizontal.

A alteração do endereço inicial dos ciclos de transferência permitirá deslocar a imagem em todas as direcções, dentro da respectiva janela de visualização.

Não é necessário gerar qualquer endereço quando da execução de ciclos de refrescamento, pois este é gerado e actualizado automaticamente pela própria memória.

8- CONCLUSÃO

No que se refere ao *hardware* já projectado, estamos em condições de proceder à sua implementação. Esta irá ser feita usando a tecnologia de *wire wrapping*, em placas apropriadas e com as dimensões exigidas pelas especificações do bus *Multibus II*. Para conter o elevado número de circuitos integrados, duas soluções se apresentam:

- Usar uma placa de WW à qual vão ligar uma ou mais placas de circuito impresso contendo a memória do sistema gráfico;
- Usar duas ou mais placas de WW interligadas.

O *hardware* de ligação ao bus do IBM PC será implementado numa placa de WW com o formato apropriado para este computador e comunica com a placa principal através de um *flat cable* de quarenta condutores.

No que se refere ao *hardware* que ainda está a ser projectado, são de salientar alguns problemas que se fazem desde já sentir. São eles:

- A necessidade de transferir a velocidades elevadas a informação correspondente às imagens dinâmicas, obriga ao tratamento simultâneo de 2 pixel num intervalo de tempo relativamente reduzido. Como consequência, todo o *hardware* que lida com esta informação terá de ser dimensionado de modo a suportar 48 bit (2 pixel x 24 bit/pixel), o que irá conduzir a um número relativamente elevado de circuitos integrados;

- Outra consequência do que atrás foi dito é a necessidade de se dispor de um controlador de memória capaz de gerar ciclos de acesso do tipo *enhanced page-mode write cycle*. Juntamente com outras condições impostas pelo *hardware* de ampliação vertical, tal como seja a necessidade de repetir um número arbitrário de vezes o endereço gerado quando da execução dos ciclos de transferência, prevê-se que o controlador em causa terá de ser implementado usando elementos discretos. Esta solução irá ser, como é óbvio, bastante complexa em termos de projecto além de bastante custosa em termos de implementação prática;

- O *hardware* de ampliação horizontal deverá gerar vários sinais de *clock*, dos quais três são do tipo ECL, resultando na utilização de um número apreciável (catorze) de circuitos integrados desta família lógica.

No que se refere ao *software*, este irá ser desenvolvido por etapas. Numa primeira fase, aproveitaremos ao máximo o código em linguagem C já existente. Posteriormente, procederemos ao desenvolvimento de um número sempre crescente de procedimentos alternativos, escritos em linguagem *assembly*, de modo a aproveitar os recursos do sistema.

BIBLIOGRAFIA

- 1 - Susan Angebrannt, Raymond Drewry, Philip Kariton, Todd Newman
Strategies for Porting the X v11 Sample Server
Massachusetts Institute of Technology, 1988
- 2 - J. Paulo Pereira
Estudo comparativo de dispositivos gráficos
Rel. Int. INESC-Norte, 1988
- 3 - J. Paulo Pereira, F. Nunes Ferreira
Os Controladores/Processadores Gráficos actualmente disponíveis
Primeiro Encontro Português de Computação Gráfica, 1988

- 4 - TMS34020/TMS34082 Preliminary Specification
Texas Instruments, 1988
- 5 - TMS44C251 Preliminary Specification
Texas Instruments, 1988
- 6 - F. Nunes Ferreira, J. Paulo Pereira, A. Cardoso da Costa, A. Teixeira Puga
Visualização Monocromática em Muito Alta Resolução
3.º Simpósio das Telecomunicações, 1988
- 7 - F. Nunes Ferreira, J. Paulo Pereira, A. Cardoso da Costa, A. Teixeira Puga
Visualização Gráfica para Microcomputadores
Primeiro Encontro Português de Computação Gráfica, 1988
- 8 - F. Nunes Ferreira, J. Paulo Pereira, A. Cardoso da Costa, A. Teixeira Puga
Graphics Hardware for Microcomputers
First Luso-German Meeting on Computer Graphics, 1988
- 9 - J. Paulo Pereira
Multibus II
Rel. Int. INESC-Norte, 1988
- 10 - MPC User's Manual
Intel Corporation, 1986
- 11 - David Rosenthal, Adam de Boor, Bob Scheifler
Godzilla's Guide to Porting the X v11 Sample Server
Massachusetts Institute of Technology, 1988
- 12 - Susan Angebrannt, Raymod Drewry, Philip Karlton, Todd Newman
Definition of the Porting Layer for the X v11 Sample Server
Massachusetts Institute of Technology, 1988
- 13 - X11 Server Extension
Digital Equipment Corporation, 1987-88