

IMPLEMENTAÇÃO DE SOFTWARE DE APOIO PARA UM
CONTROLADOR AVANÇADO DE VÍDEO

Celso Hiroshi Yamaguchi *
Maria Ruth P. L. Reganati **
Hélio Azevedo **
Prof. Dr. Clésio Luis Tozzi *

* CNPq Conselho Nacional de Pesquisa e
Desenvolvimento Tecnológico
DCA Departamento de Engenharia de Computação e
Automação Industrial
FEE Faculdade de Engenharia Elétrica
UNICAMP Universidade Estadual de Campinas
Cidade Universitária
13081 Campinas SP

** DCP Departamento de Controle de Processos
IA Instituto de Automação
CTI Centro Tecnológico para Informática
Rodovia SP 340, km. 104,5
13100 Campinas SP

I. OBJETIVOS

O Controlador Gráfico Avançado ACRTC de Advanced CRT Controller reúne em apenas um CHIP as funções de Gerenciamento de Memória de Vídeo (refresh, Sync Signal, Address Control, Frame Window) e a de um excelente manipulador de memória de vídeo, executando nela primitivas gráficas. No caso específico no hardware disponível, temos uma memória total de 1024 por 785 pixels, e 256 cores diferentes escolhidas entre 262144 cores, podendo ser mapeados na tela uma janela de até 1024 por 785 pixel. Para o usuário fica então facilitadas as manipulações gráficas, bastando apenas enviar pela porta de comunicação do Controlador os comandos adequados.

O Sistema gráfico instalado no Sistema HoMuK (Homogeneous Multiprocessor Kernel), uma Estação de Trabalho Gráfica Avançada (Graphics Workstation) é um conjunto de programas escritos em linguagem de máquina (Assembler) 68000, lembrando que o Sistema Operacional instalado no HoMuK é o sistema alemão FlexOS, que possui como linguagens de alto nível ForTran e FLECS ForTran (pré-compilador), além de Assembler 68000. Estes programas realizam todo o acesso necessário para a realização de primitivas gráficas, utilizando-se das facilidades do ACRTC.

Outra implementação utilizando o Controlador Avançado de Vídeo foi a instalação de um Terminal Gráfico Inteligente utilizando o padrão TEK 410x, inclusive com manipulação de

segmentos (encadeamentos de primitivas gráficas), conectado via linha serial, servindo portanto como terminal Gráfico Colorido de Alta Resolução para vários Sistemas Host, como DEC VAX, PCS CADMUS (UNIX), IBM PC-AT (XT), o próprio HoMuK, etc.

Este trabalho pretende, portanto, dar ao leitor uma idéia básica do funcionamento do Controlador de Vídeo Avançado, da Memória de Vídeo e Look-up Table, e de como se implementou Software para o seu funcionamento na Workstation HoMuK, e no Terminal Gráfico Inteligente (tipo TEK 410x).

II. HARDWARE & SOFTWARE DISPONÍVEL

Tivemos disponível para a elaboração de ambos os sistemas gráficos determinados tipos de Hardware (tanto para o desenvolvimento das rotinas do projeto, quanto para a utilização efetiva como o Sistema) e Software (para a elaboração de programas ou utilização como rotinas do Sistema), que descreveremos a seguir.

II.1. Hardware do Sistema HoMuK

O hardware utilizado para o desenvolvimento e utilização dos Drivers do Sistema Gráfico no HoMuK (um dos barramentos VME contido no Sistema HoMuK) é:

1. CPU-Board Eltec El/68K System (CPU 68000), 256K Bytes de memória, duas portas seriais e controlador de Floppy Disk, ligados a um Floppy Disk de 8" (1 Mega Bytes de memória), Terminal do Usuário padrão VT 100, e ou a outro sistema Host, eg. CADMUS (UNIX).

2. Placa controladora de Winchester (Hard Disk) ligadas a um Winchester de 40 Mega Bytes de memória (interface construída no Laboratório HoMuK)

3. Placas de Vídeo Eltec PIG 1/68K e PIG 2/68K, respectivamente controladora de vídeo e expansão de memória de vídeo com Look-up table, com memória de vídeo de 2 Mega Bytes, e 256 cores entre 262144 (Look-up Table com 24 bits por índice) por pixel (cada pixel ocupa um Byte na memória), Dot clock de 55 MHz entrelaçados, resultando em resolução de 1024 por 785 pixels, ligados à um monitor de vídeo (com entrada RGB e Sync) Colorido Eltec mod. A-120.

4. Placa de BusCoupler, ou acoplamento de barramento para um barramento de comunicação, que realiza comunicação eg., entre o Sistema Gráfico e o Sistema de Digitalização de Imagens (frame grabber), onde podemos transferir e modificar (editar, recolorir, manipular) imagens capturadas pelo Sistema Digitalizador (Utilizando-se de um Editor Gráfico Interativo), e ou retransferir essas imagens manipuladas de volta para a memória do Sistema Digitalizador.

5. Mesa Digitalizadora (tablet) mod. 1030 COMICRO, com

4000 por 4000 pontos de resolução (para a utilização como Cursor da Workstation)

II.II. Software do Sistema HoMuK

O Software utilizado para a implementação do Sistema Gráfico é composto por Compilador ForTran, Pré-compilador Flecs, que adiciona aos comandos do ForTran algumas instruções de Linguagem Estruturada (While, Select, etc.) e Assembler 68000 (MAS) disponíveis para o usuário, dentro do Sistema Operacional FLEXOS, do HoMuK. Os programas do Driver, devido às características de acesso a nível de máquina, são na maioria escritos em Assembler 68000, mas algumas implementações mais acessíveis ao usuário (Editor Gráfico) utilizam-se das demais.

II.III. Hardware para desenvolvimento do Terminal

No Terminal Gráfico Inteligente utilizamos:

1. Placa de CPU Eltec E-1/68K System e terminal alfanumérico tipo VT 100.

2. Placa de Vídeo de Alta resolução Colorido Eltec PIG 1 e PIG 2 68K System e monitor Colorido de Vídeo Eltec Mod. A-120.

Para o desenvolvimento do Terminal Gráfico utilizamos:

1. Sistema de Desenvolvimento "8560 Multi-User Software Development Unit" associado à workstation "8540 Integration Unit" com emulador para MC 68000 desenvolvido pela TEKTRONIX Inc. O Sistema Operacional do 8560 é o TNIX Versão 2.0 compatível com UNIX. O Sistema descrito está instalado no Centro Tecnológico para Informática (CTI).

2. Workstation PCS CADMUS 9000 multiusuário com Sistema Operacional MUNIX V, compatível com UNIX system V., com CPU 68010, instalado no Laboratório de Computação Gráfica do DCA FEE UNICAMP.

III. CONCEITOS BÁSICOS

Antes de descrevermos o funcionamento de cada primitiva interna do Controlador, devemos entender alguns conceitos básicos presentes na estrutura dos seus comandos. São eles:

III.I. Sistema de Coordenadas

O mapeamento da memória de vídeo utiliza um sistema de coordenadas cartesiano, 2D, com origem no canto esquerdo da tela com coordenadas, no eixo-x pertendo a (0,1024) e eixo-y pertencendo a (0,2048). As instruções seguem sempre o formato (x,y). Mapeados ao mesmo tempo na tela temos uma janela de dimensões máximas 1024 * 785 pixels, que podem ser escolhidos em qualquer local da memória de vídeo. É importante observar que o

sistema pode modificar qualquer parte da memória de vídeo, não apenas a mapeada.

III.II. Ponto Corrente (Current point)

O Ponto Corrente é um par de apontadores (X,Y) que guardam o ponto de referência para o desenho de primitivas (geralmente o ponto de início da primitiva). Como ele é utilizado para o processamento de primitivas, temos que modificá-lo a todo instante que quisermos desenhar uma figura num determinado ponto da memória de vídeo.

III.III. Padrão (Pattern)

Utilizamos no Sistema Gráfico a memória de Padrões (Pattern RAM) interna do Controlador, que é uma matriz de bits, que mascara o desenho de primitivas. O controlador possui cinco registros de cores, dois dos quais representando as cores correntes: Cor 0 e Cor 1. Quando um padrão está sendo colocado na memória, cada bit da matriz de padrões corresponde a uma das cores correntes (Cor 0 ou Cor 1), de acordo com seu valor (0 ou 1), produzindo figuras na memória iguais ao padrão. Desta maneira podemos preencher áreas da memória com padrões bicoloridos, eg. para inserir textos, ícones, texturas, etc.

III.IV. Registros de Cores (Color Registers)

São cinco os Registros de Cores, cujas funções são comentadas a seguir: Os Registros de cores Cor 0 e Cor 1 guardam respectivamente os valores das cores que serão colocadas na memória caso os bits 0 e 1 forem lidos da matriz de padrões. Guardam, portanto os valores das cores correntes. O registro de cor de Comparação guarda o valor da cor que será comparada, por exemplo, para preenchimento condicional de pixel. O Registro de Cor de Borda define a cor de borda para preenchimento (GPAINT), e o registro de cor de Máscara define uma máscara de bits para preenchimento de pixels.

III.V. Área de Trabalho (Area Definition)

Podemos definir uma área de trabalho para o desenho de primitivas gráficas, e também vários modos de utilização dessas áreas, como: desenho habilitado apenas dentro da área definida, desenho habilitado apenas fora da área, desenho habilitado tanto dentro quanto fora da área. O formato da área é sempre retangular, sendo definidos as coordenadas dos cantos deste retângulo (Xmínimo, Ymínimo, Xmáximo, Ymáximo).

IV.I. Modo de Cores

O modo de Cores determina a maneira que será utilizada para o "pixel setting", ou seja, para cada bit lido da matriz de padrões, que tipo de cor (Lida dos registros de cores) será alocado na memória de vídeo. Os valores de COL devem ser 0, 1, 2, 3. A tabela V.II. mostra o modo de operação para cada valor de COL.

Valor COL	Modo de Cor
00	Se Pattern RAM Bit = 0, Cor 0 usada Se Pattern RAM Bit = 1, Cor 1 usada
01	Se Pattern RAM Bit = 0, suprime Se Pattern RAM Bit = 1, Cor 1 usada
02	Se Pattern RAM Bit = 0, Cor 0 usada Se Pattern RAM Bit = 1, Cor 1 usada
03	Conteúdo da Pattern RAM usado diretamente como Cor.

Figura V.II. Modos COL.

Obs.: Os registros de Cor 0 e 1 tem 8 Bits, a Pattern RAM possui ate 16 * 16 Bits.

IV.III. AREA Area Mode

O modo de área indica de que maneira será utilizada a área de trabalho definida pelo Definidor de Áreas. O parâmetro AREA deve ser 0, 1, 2, 3, 4, 5, 6, 7. A tabela V.III descreve o procedimento correspondente à cada valor de AREA. É importante frisar que o usuário NUNCA deve utilizar os MODOS PROIBIDOS, pois o processamento, nesses casos, pode ser abortado, abortando todo o sistema.

Valor AREA	Modo de Área de Trabalho
00	Desenho executado sem consulta de Área de Trabalho.
01	Modo Proibido
02	Desenho suprimido FORA da Área de Trabalho
03	Modo Proibido
04	Mesmo que AREA = 0
05	Modo Proibido
06	Desenho Suprimido DENTRO da Área de Trabalho
07	Modo Proibido

Figura V.III. Modos de AREA

V. COMANDOS DO CONTROLADOR GRAFICO AVANÇADO

Daremos agora uma introdução ao funcionamento dos comandos já implementadas no Controlador Avançado de Vídeo (ACRTC). Todos os comandos Gráficos do Controlador utilizam-se dos conceitos e modos de operação explicados anteriormente.

Podemos dividir os comandos do Controlador em tres tipos distintos: Comandos de Acesso a registros, Comandos de Transferencia de Dados e Comandos Gráficos, a seguir detalhes dos mesmos.

V.I. Comandos de Acesso a Registros

São comandos que manipulam os registradores internos do Controlador de Vídeo, que são:

PR 00	Cor 0
PR 01	Cor 1
PR 02	Cor de Comparação
PR 03	Cor de Borda (filling)
PR 04	Máscara de Cor
PR 05	
...	Registros de Controle da Memória de Padrões
PR 07	
PR 08	Coordenada de definição de AREA Xmin
PR 09	Coordenada de definição de AREA Ymin
PR 0A	Coordenada de definição de AREA Xmax
PR 0B	Coordenada de definição de AREA Ymax
PR 0C	Apontador de Leitura e Escrita (Upper 16 Bits)
PR 0D	Apontador de Leitura e Escrita (Lower 16 Bits)
PR 10	Apontador Gráfico (Upper 16 Bits)
PR 11	Apontador Gráfico (Lower 16 Bits)
PR 12	Ponto Corrente Coordenada X
PR 13	Ponto Corrente Coordenada Y

Fig. V.I. Registradores de Parametros do ACRTC 63484

ORG Inicializa a relação entre a origem do sistema de coordenadas e os endereços físicos.

WPR n Escreve um word no registro n (n é um parâmetro contido na palavra de comando).

RPR n Lê um word do registro n do controlador.

WPTN Escreve dados na RAM (memória de) Padrões

RPTN Lê dados da Pattern RAM.

V.II. Comandos de Transferencia de Dados

São comandos que executam transferência direta de dados da memória de vídeo para a memória da CPU (no nosso caso um 68000) ou interna na memória de vídeo. Lembramos que no hardware Eltec PIG 68K System não existe a transferência de dados por DMA, mas listaremos também os comandos DMA que o Controlador possui.

DRD Leitura DMA de memória de vídeo

DWT Lê dados da memória de vídeo por DMA

DMOD	Modifica dados da memória de vídeo por DMA
RD	Lê um word da memória de vídeo
WT	Escreve um word na memória de vídeo
MOD	Modifica um word da memória de vídeo
CLR	Apaga a memória de vídeo
SCLR	Apagamento Seletivo da memória de vídeo
CPY	Copia um bloco da Vídeo RAM num outro bloco
SCPY	Cópia Seletiva de bloco da memória de Vídeo

V.III. Comandos Graficos

Cada primitiva gráfica executada pelo controlador inicia-se com uma palavra de comando de 16 bits, que indica qual primitiva a ser executada, se o deslocamento é absoluto ou relativo em relação às coordenadas atuais do Ponto Corrente, e o modo de operação (AREA, COLOR, OPM). Todos os comandos levam em conta a posição do Ponto Corrente que geralmente aponta para o local (coordenada) da última manipulação gráfica realizada na memória. Os comandos gráficos são:

AMOVE	Move Ponto Corrente a Coordenada Absoluta
RMOVE	Move PC a um deslocamento relativo ao PC atual
ALINE	Traça linha do PC atual a Coordenada Absoluta
RLINE	Traça linha do PC a deslocamento relativo
ARCT	Traça retângulo entre PC e Coordenada Absoluta
RRCT	Traça retângulo entre PC e Coordenada Relativa
APLL	Traça poli linhas (linhas encadeadas)
RPLL	Traça poli linhas (Coordenadas relativas)
APLG	Traça polígono (Coordenadas absolutas)
RPLG	Traça polígono (Coordenadas relativas)
CRCL	Traça uma circunferência (raio dado)
ELPS	Traça uma elipse (Horizontal)
AARC	Traça Arco Circular (Coordenadas Absolutas)
RARC	Traça Arco Circular (Coordenadas Relativas)
AEARC	Traça Arco Elíptico (Coordenadas Absolutas)
REARC	Traça Arco Elíptico (Coordenadas Relativas)
AFRCT	Traça retângulo preenchido (Coords. Absolutas)
RFRCT	Traça retângulo preenchido (Coords. Relativas)
PAINT	Preenche uma Área fechada em torno do PC
DOT	Coloca um ponto na Memória
PTN	Desenha um padrão na Memória
AGCPY	Cópia gráfica de dados (diferentes direções)
RGCPY	Cópia gráfica (Coordenadas Relativas)

VI. IMPLEMENTAÇÕES DO CONTROLADOR GRAFICO

A implementação do controlador gráfico inicia-se com a rotina de inicialização do sistema, que prepara o controlador para a execução de primitivas gráficas. Este programa é comum tanto

estudantes. Basicamente esta fase compreende a: implementação de um driver para Mesa Digitalizadora (tablet), e um Driver mais eficiente para o Terminal Alfanumérico do Usuário, um driver para utilização do sistema de Arquivo de Dados e dos Drivers implementados num programa Interativo. Este programa Interativo compreende um editor gráfico com a utilização do Tablet (Cursor no Terminal Gráfico) e do Terminal Alfanumérico (Comandos, Menus, Dados do Usuário, Cores, texturas, padrões, textos, etc.), que execute comandos gráficos (inicialmente bidimensionais) que realizam desenhos na tela gráfica, armazenando-os no Hard-Disk (Winchester), podendo o usuário acessá-los à medida de suas necessidades (animação, Computer Art, Projetos Auxiliados por Computador, etc.).

VI.II. Terminal Grafico Inteligente

O outro projeto desenvolvido sobre as Placas Gráficas Eltec PIG 68000 System foi um Terminal Gráfico Inteligente, com a utilização de um Barramento VME. O Firmware residente na placa de CPU, denominado NUGRA (Núcleo Gráfico) foi desenvolvido no IA CTI em linguagem C. O NUGRA tem por objetivo atender as necessidades de usuários por recursos gráficos 2D, contendo: primitivas gráficas, segmentação e conceito de view.

Uma das principais características do NUGRA é a sua transportabilidade. Atualmente ele se encontra instalado nos seguintes ambientes:

1. IBM-PC com interface realizada através de rotinas via biblioteca.
2. PLACA GRÁFICA PV-02, desenvolvida no CTI, com interface realizada através de uma memória Dual-port.
3. PLACA PIG 01/02 Eltec com interface realizada através de linha serial com protocolo TEK 410x.

O NUGRA necessita, quando instalado num novo tipo de hardware de um driver que o compatibilize com este ambiente: módulo Máquina Alvo. Este módulo é responsável pelas manipulações de nível de máquina do Sistema gráfico, eg. Entradas e Saídas (Comunicação Serial e Comandos para a placa gráfica), manipulação de interrupções, etc., que são específicas de cada Hardware. No nosso caso, ele é composto por programas escritos em Linguagem C e Assembler 68000.

Os programas da Máquina Alvo foram inicialmente projetados para serem compilados no Sistema de Desenvolvimento TEKTRONIX, por isso apresentam especificidades de programação do Compilador C e Assembler do Sistema Operacional TNIX no desenvolvimento de Software para projetos 68000, além de facilidades comuns em Sistemas de Desenvolvimento. Este foi um ponto crítico para a geração do núcleo no Sistema CADMUS, pois não sendo Sistema de Desenvolvimento, apresenta algumas rotinas da

biblioteca padrão C que acessam rotinas do Sistema Operacional, que evidentemente não pode ser ligado com o Núcleo. Tivemos que desenvolver então algumas rotinas que as substituam, além de resolver problemas de diferenças de sintaxe entre os Assemblers dos dois sistemas.

A manipulação das primitivas gráficas na tela (alocação, edição, cortes, etc.) segue o sistema de Segmentos, muito facilitado pela programação em Linguagem C. Os segmentos são uma lista encadeada de primitivas gráficas. Abrimos um segmento e inserimos nele algumas primitivas gráficas, depois fechamos. Um próximo segmento pode conter um já fechado. Podemos executar sobre o segmento, ie., sobre todas as primitivas que ele contem, transformações gráficas, como translação, rotação, escalamento, etc., inserindo-o em outro, assim por diante. Este conceito fica transferido para o terminal, e o usuário (ou o Host e seus programas) podem utilizá-lo, apenas enviando os comandos apropriados.

O Módulo de comunicação é o parte do NUGRA que trata da comunicação do Sistema com o Terminal do Usuário (terminal tipo VT 100) e o Host. A comunicação com o Terminal do Usuário é realizada através de polling, e com o Host através de interrupções.

O Módulo de execução de primitivas utiliza-se de comandos do Controlador de vídeo, e basicamente executa pontos, linha, retas, retângulos (boxes), polígonos, preenchimento e splines. Todos os comandos do terminal gráfico são sequências de ESCapes seguindo o padrão TEK 410x, inclusive com os conceitos de transformação de janela de trabalho, e janela lógica e física da memória constantes no padrão TEK. Além das primitivas gráficas simples, o sistema possui um módulo de texto, sendo possível ao usuário a confecção de textos na Tela Gráfica. Os textos podem possuir 4 direções, e podemos programar sua altura, largura e espaçamento.

A utilização atual do Terminal Gráfico, vem sendo como apoio (Saída Gráfica) de figuras de Realismo, projeto desenvolvido no Laboratório de Computação Gráfica, com utilização de técnicas de realismo como Ray-Tracing. Para esta função foram implementadas dois comandos: um alocador de frames (linhas de imagem na tela) e um programador de Look-up table. Como a confecção de figuras de realismo geralmente geram uma matriz de pontos, seria necessário apenas alocar esta matriz na memória de vídeo. Cada acesso ao terminal gráfico pode ativar o máximo de 1024 pontos na memória de vídeo (correspondente a uma linha do frames). O programa de manipulação da Look-up Table é necessário por que nesta aplicação é mais conveniente ao usuário um acesso direto à Tabela de Índices de Cores. Atualmente cerca de 2 minutos são necessários para a transferência de uma imagem completa (1024 por 785 pixel).

- (I). PCS, Periphere Computer Systeme GmbH, W. Germany,
"MUNIX V.2/03 Manual" (1984)
- (Ia). PCS, Periphere Computer Systeme GmbH, W. Germany,
"MUNIX V.2/03 Manual Chap. 5, Special Files & Functions" (1984)
- (Ib). PCS, Periphere Computer Systeme GmbH, W. Germany,
"MUNIX V.2/03 Manual Chap. 4, As 68000 User's Guide V 1.1" (1982)
- (II). Eltec Elektronik GmbH, W. Germany,
"68K System Documentation PIG-1 & PIG-2/68K System, Rev A" (1986)
- (III). Yamaguchi, Celso H., UNICAMP-FEE,
"Relatório de Estágio de Iniciação Científica (CNPq)" (1988)
- (IV). HITACHI Co., Japan,
"HD 63484 ACRTC Specification, Rev 2.0" (1985)
- (V). Newmann, W. M., Sproull, R. F., "Principles
of Interactive Computer Graphics, 2nd Ed.", Mc Graw Hill, (1979)
- (VI). Kernigham, B. W., Ritchie, D. M.,
"The C Programming Language", Prentice Hall, (1978)
- (VII). Eltec Elektronik GmbH, W. Germany,
"68K System Description 68K Bug Version x.y (z) Rev. A" (1983)
- (VIII). MINATO Electronic Inc., Japan,
"MINATO Model 1866A EP-ROM Programmer Instruction Manual" (1986)
- (IX). TEKTRONIX Inc., USA,
"TEK Reference Guide 4107/4109 Computer Display Terminals" (1983)
- (X). TEKTRONIX Inc., USA,
"TEKTRONIX 4107/4109 Operators Manual" (1983)
- (XI). TEKTRONIX Inc., USA,
"TEKTRONIX 4107/4109 Programmers Reference Manual" (1983)
- (XII). Reganati, Maria Ruth P. L., UAT IA SEI/CTI,
"Editor Gráfico Interativo Para Terminais TEK 410x" (1986)
- (XIII). Foley J. D., Van Dam, A., "Fundamentals
of Interactive Computer Graphics", Addison Wesley, (1981)