UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA DE CAMPINAS

DEPARTAMENTO DE ENGENHARIA ELÉTRICA/ELETRÔNICA

SISTEMA PARA REPRESENTAÇÃO DA INFORMAÇÃO EM TERMINAIS GRÁFICOS DE PEQUENO PORTE

BEATRIZ MASCIA DALTRINI

Orientador: Prof. MANUEL DE JESUS MENDES

Tese de Mestrado apresentada à Faculdade de Engenharia da Universidade Estadual de Campinas

DEZEMBRO - 1976

ÍNDICE

INTRODUÇÃO

Desde a idade da pedra, quando o homem começou a desenhar nas paredes de sua caverna, ele sentiu a necessidade de obter uma forma de se comunicar com o resto do mundo.

E o mundo evoluiu.

Cientistas de todas as épocas usaram desenhos, gráficos e símbolos matemáticos para explicar suas teorias.

Para melhor desenvolver estas teorias o homem inventou máquinas que suprissem suas deficiências. O computador surgiu para suprir uma deficiência humana: o homem é inteligente e criativo, porém lento e sujeito a erros. O computador é muito rápido e praticamente livre de erros, sendo portanto uma ferramenta poderosa a serviço da humanidade. Porém, sem uma interface para comunicação com o homem, o computador é surdo e mudo. Assim surgiu o problema de comunicação entre o homem e a máquina. Desde o aparecimento da primeira máquina de calcular o homem vem tentando aprimorar a integração homem-máquina. Como resultado destes esforços apareceram as leitoras, impressoras e periféricos gráficos.

Os periféricos gráficos desenvolveram-se muito desde o aparecimento do primeiro "plotter" até o moderno "display" interativo.

O "display" interativo trouxe um diálogo maior entre o homem e a máquina. A partir da figura obtida no "display", o homem, com sua inteligência e criatividade, pode tomar uma decisão e modificar o processamento, agindo diretamente na tela. Por ser um dispositivo de entrada e saída tão eficiente, seu uso foi generalizado em diferentes aplicações. Devido a essa diversidade, o "software" de apoio, fornecido pelos fabricantes, é também bastante genérico e esta qualidade trará problemas para aplicações muito específicas.

Este trabalho tem a finalidade de resolver um destes problemas, criando um algorítmo que será usado somente em aplicações bem determinadas, nas quais o "software" genérico é ineficiente.

O sistema gráfico instalado na FEC utiliza o "software" fornecido pelo DEC. O problema apareceu ao se tentar a realização de gráficos de muitos pontos. Devido à pequena capacidade do terminal gráfico e à complexidade do "software" genérico, a memória era insuficiente para conter tantos dados. A solução apresentada neste trabalho será um algorítmo que tratará somente de gráficos de muitos pontos.

CAPÍTULO I - INFORMAÇÃO GRÁFICA

1.1 Introdução

Apesar da enorme variedade de periféricos gráficos existentes no mercado, podemos classificá-los em dois grupos:

- periféricos passivos;
- periféricos ativos.

Os periféricos passivos produzem imagens processadas no computador conforme o programa do usuário. São periféricos gráficos passivos o "plotter" e o "storage tube device", entre outros.

Os periféricos ativos, além de produzirem imagens, podem receber estímulos do usuário, estabelecendo portanto o diálogo gráfico entre o homem e a máquina. O periférico gráfico ativo utilizado é do tipo "refreshing tube device", cuja característica é uma tela de baixa persistência. Ao contrário do "storage tube device" a imagem não se mantem na tela; para sua visualização é necessário que esta imagem seja repetida cerca de trinta vezes por segundo, ou seja, o máximo intervalo de tempo entre duas repetições consecutivas não deverá ultrapassar 33 ms.

A imagem é gerada pelo processamento de um programa gráfico ("display file"), residente na memória, composto de uma sequência de instruções gráficas e dados.

A qualidade da imagem vai depender do tempo necessário para a execução do programa gráfico. Para a obtenção de uma imagem estável na tela, o tempo de execução do programa gráfico não deve ultrapassar um limite dado pelo máximo intervalo de tempo entre duas repetições consecutivas da imagem. No caso do programa gráfico ser muito extenso, a imagem não será repetida a cada 33 ms. e a persistência da tela será insuficiente para mantê-la.

Durante a execução do programa gráfico, o processador envia dados para a formação da imagem na tela através de um canal de dados. (Fig. 1.1)

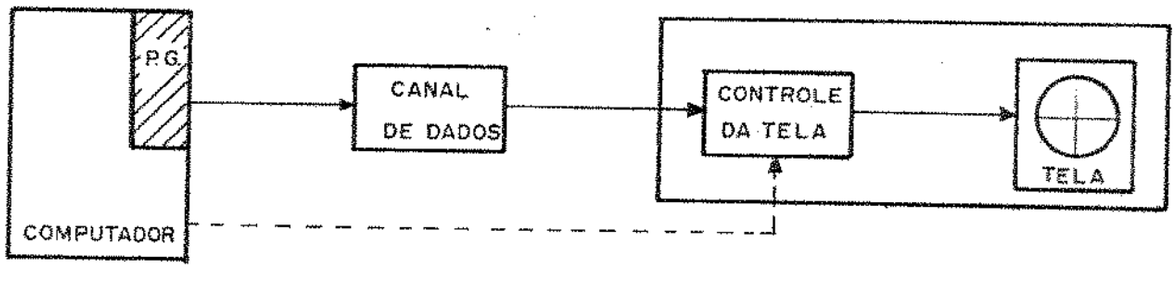


FIG. 1.1

Geralmente o periférico gráfico é um terminal ligado a um computador central trabalhando em multi-programação. A transferência de dados pela linha é muito lenta; as interfaces convencionais são de 110 a 9600 bauds (bits por segundo), ou seja, transferem no máximo 500 palavras por segundo. Se compararmos o tempo mínimo para transferência de uma palavra (2 ms) com o máximo intervalo de tempo entre duas repetições consecutivas da imagem (33 ms), veremos que, nestas condições, é impossível obtermos uma imagem mais elaborada estável na tela. Assim, o terminal gráfico, ligado a um computador central, deve possuir uma memória auxiliar ("buffer") onde estará uma cópia do programa gráfico. (Fig. 1.2)

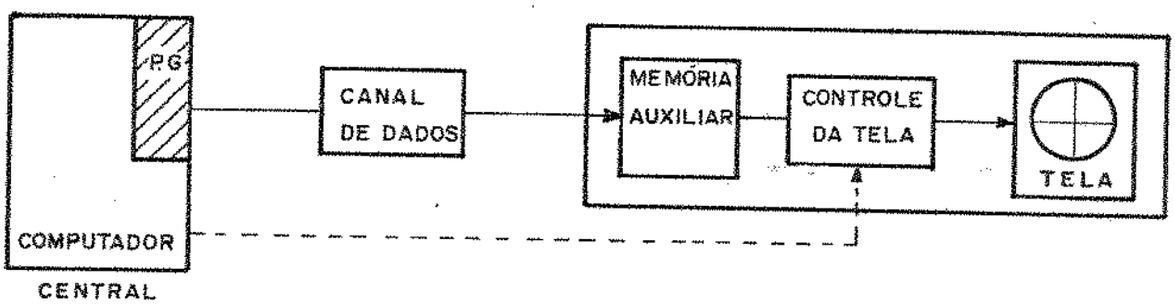


FIG. 1.2

Existem terminais gráficos que possuem um mini-computador com uma unidade de processamento gráfico. Neste caso, a execução do programa gráfico será local (Fig. 1.3). Em sistemas mais sofisticados o mini-computador possui também uma unidade de processamento geral, existindo a possibilidade de uma interação local, pela modificação do programa gráfico a partir de um comando do usuário, sem a interferência do computador central.

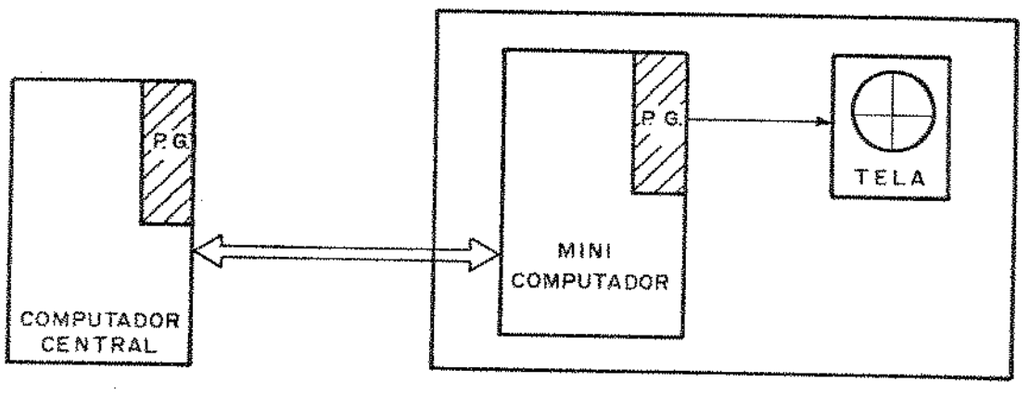


FIG 1.3

1.2 Funções Básicas

No processamento de um gráfico podemos distinguir várias etapas; o cálculo das abcissas e ordenadas, sua transformação para formar o banco de dados, a geração do programa gráfico e sua execução, a saída da imagem para a tela e a interação do usuário com o computador.

Todo esse processo foi dividido em cinco funções básicas, representados no Diagrama da Fig. 1.4:

1) Função de aplicação (FA);
2) Função de dados (FD);
3) Função gráfica (FG);
4) Função de saída (FS);
5) Função de entrada (FE).

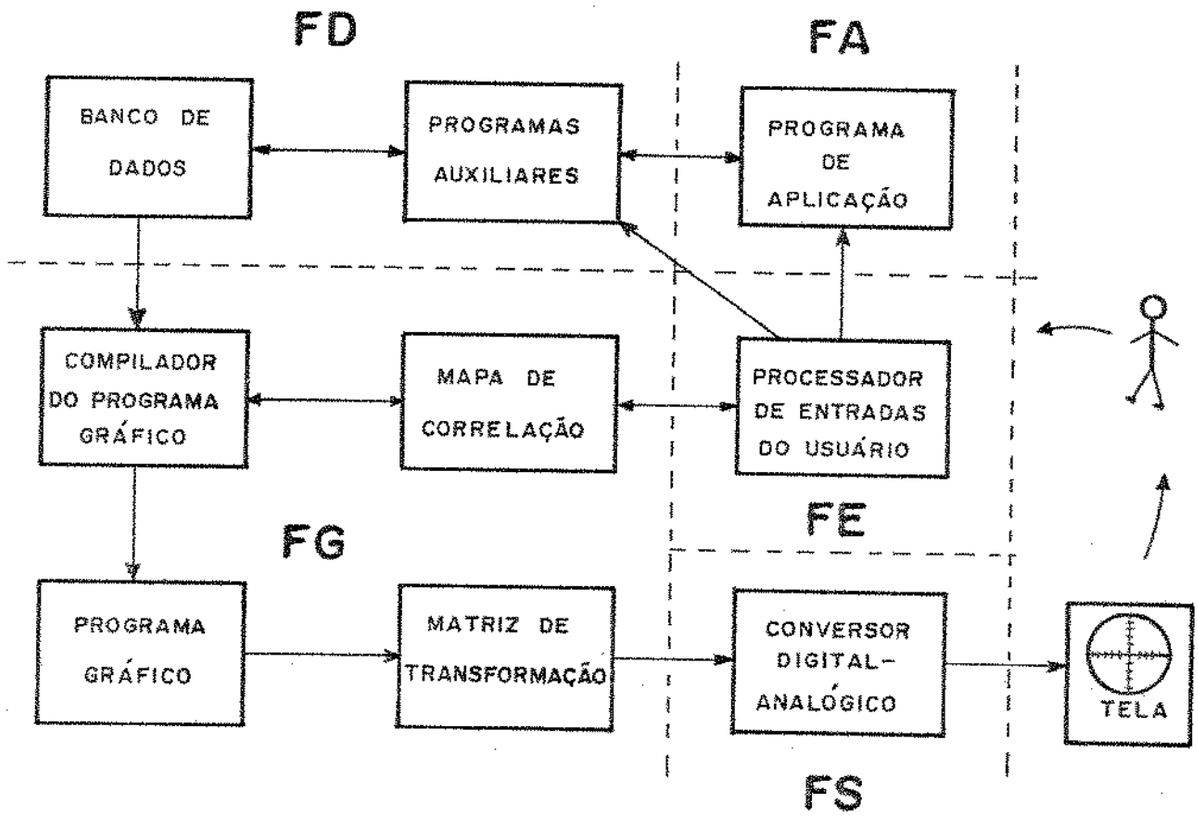


FIG. 1.4

1.2.1 Função de Aplicação

A função de aplicação é realizada pelo programa de aplicação do usuário. Este programa processa os cálculos para a geração das abcissas e ordenadas que darão origem à imagem. Geralmente este programa é feito em FORTRAN IV ou outra linguagem de alto nível.

1.2.2 Função de Dados

A função de dados consta de um banco de dados e um conjunto de programas auxiliares que fazem a ligação entre o programa de aplicação e o banco de dados. Este deve ter uma estrutura de dados compatível com o programa gráfico a ser executado. É o programa de aplicação do usuário que gera a estrutura de dados conveniente para o processo, pela chamada sucessiva dos programas auxiliares.

1.2.3 Função Gráfica

A função gráfica é realizada pelo programa gráfico, seu compilador, matriz de transformação e mapa de correlação. O compilador codifica as instruções do programa, que será executado pelo processador gráfico, gerando a imagem.

Esta imagem pode ser modificada pela matriz de transformação que realiza translação, rotação, projeção e escalamento.

Cada uma das transformações é realizada do seguinte modo: (referência [1]).

Seja $[P] = [x \ y \ z \ 1]$ o ponto original e $[P] = [x'y'z'1]$ o ponto transformado:

I) Translação

$$[P'] = [P]\,[T]$$

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_o & y_o & z_o & 1 \end{bmatrix}$$

II) Escalamento

$$[P'] = [P]\,[S]$$

$$[S] = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

III) Projeção no eixo z

$$[P'] = [P]\,[P_z]$$

$$[P_z] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & z_o & 1 \end{bmatrix}$$

IV) Rotação

$$[P'] = [P]\,[R]$$

$$[R] = [R_x]\,[R_y]\,[R_z]$$

Rotação em torno do eixo x:

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \text{sen}\,\alpha & 0 \\ 0 & -\text{sen}\,\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotação em torno do eixo y

$$[R_y] = \begin{bmatrix} \cos\beta & 0 & -\text{sen}\,\beta & 0 \\ 0 & 1 & 0 & 0 \\ \text{sen}\,\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotação em torno do eixo z:

$$[R_z] = \begin{bmatrix} \cos\gamma & \text{sen}\,\gamma & 0 & 0 \\ -\text{sen}\,\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A matriz de transformação será:

$$[MT] = [T]\,[S]\,[P_z]\,[R]$$

A transformação poderá modificar a estrutura da imagem ou então modificar diretamente os dados.

O mapa de correlação faz a ligação entre o banco de dados e o programa gráfico, permitindo, a partir do programa de aplicação, achar um dado no programa gráfico ou então achar a localização correspondente, na estrutura do banco de dados, de um ponto (x,y) do programa gráfico. Este mapa de correlação é muito importante na modificação da imagem a partir dos dados.

1.2.4 **Função de Saída**

A função de saída é realizada por um conversor digital-analógico. O processador gráfico carrega dois registros (x e y) com o valor das respectivas abcissas e ordenadas. O conversor digital-analógico decodifica os valores dos registros e envia o sinal para os defletores do tubo de raios catódicos.

1.2.5 **Função de Entrada**

A função de entrada é realizada pelos periféricos gráficos ("light-pen", "tablett", "mouse"). Consta de subrotinas que são chamadas sempre que ocorre a interação do usuário com o processo, através dos periféricos gráficos.

Estas subrotinas podem agir no programa de aplicação, nos programas auxiliares ou então modificar diretamente o programa gráfico ou banco de dados, através do mapa de correlação.

1.2.6 **Classificação dos Terminais**

Conforme a capacidade de executar estas funções básicas, os terminais gráficos podem ser classificados conforme indicado na Fig. 1.5.

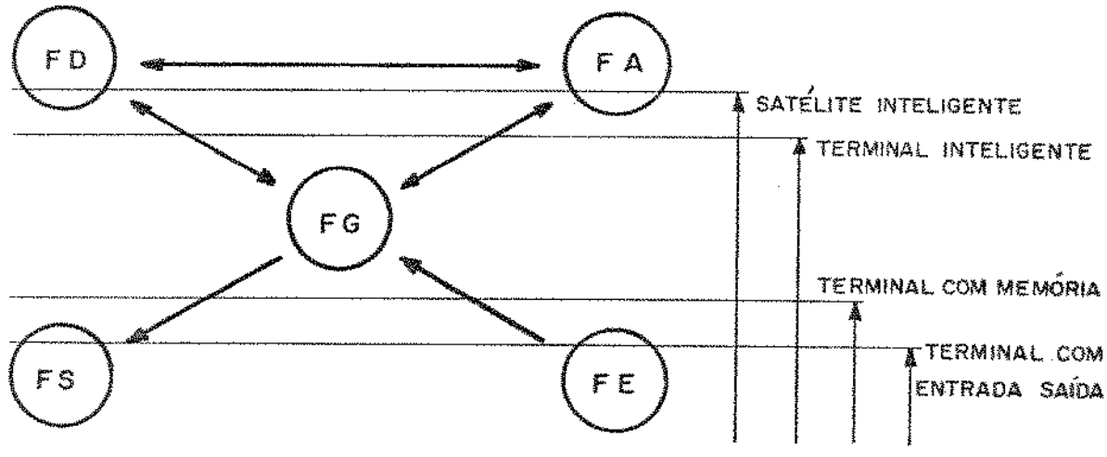


FIG. 1.5

O terminal inteligente, além de possuir memória e DPU, é capaz também de executar instruções não gráficas, pois possui uma unidade de processamento geral (CPU). Deste modo, qualquer modificação no programa gráfico pode ser feita localmente e depois enviada ao computador central, tornando o terminal mais rápido na interação.

O satélite inteligente é um terminal que possui um mini-computador com sistema operacional e compiladores, de modo a poder receber instruções em linguagem de alto nível. Além disso, pode possuir blocos em "hardware" para executar determinadas funções, como a transformação de um ponto pela matriz de transformação. Deste modo, todas as etapas do processamento gráfico podem ser executadas no terminal.

1.3 "Hardware" e "Software"

O estudo de um sistema de computação pode ser dividido basicamente em dois ítens: "Hardware e Software".

Em "Hardware", um sistema de computação gráfica consta de um processador, um tubo de raios catódicos, uma periferia não gráfica e uma periferia gráfica.

Em "Software", o mesmo sistema consta de um programa de aplicação, um banco de dados, um programa gráfico e um compilador do programa gráfico.

1.3.1 "Hardware"

O processador executa o programa gráfico gerando uma imagem no tubo de raios catódicos.

A periferia não gráfica consta de dispositivos de entrada e saída para o programa de aplicação do usuário e memórias auxiliares de massa (disco, fitas magnéticas, ...).

A periferia gráfica consta de dispositivos que permitem a interação do usuário com o processo.

Os periféricos gráficos mais comuns são:

- "light pen";
- "tablett";
- "mouse";

O terminal com entrada e saída é um periférico comum. Consta somente de um tubo de raios catódicos e um conversor digital-analógico que decodifica os registros x e y em sinais para os defletores da tela. Do mesmo modo as entradas do usuário são codificadas e enviadas ao computador. (Fig. 1.6)

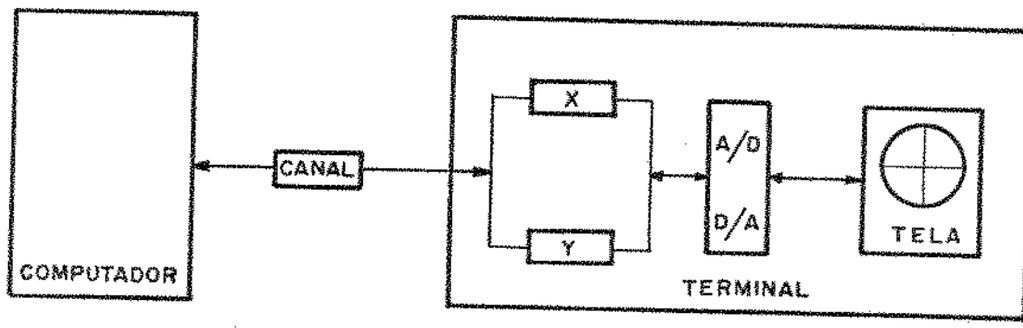


FIG. 1.6

É um terminal lento, todas as etapas do procesamento gráfico são realizadas no computador central e depois enviadas ao terminal.

O terminal com memória é usado para o tipo "refreshing tube device". Este tipo de terminal possui uma memória onde estará gravado o programa gráfico, e um processador gráfico (DPU) capaz de executar instruções gráficas. Apesar de poder executar o programa gráfico localmente, qualquer modificação deste é feita através do computador central. Sendo assim, este tipo de terminal será rápido na execução e lento na interação. (Fig. 1.7)

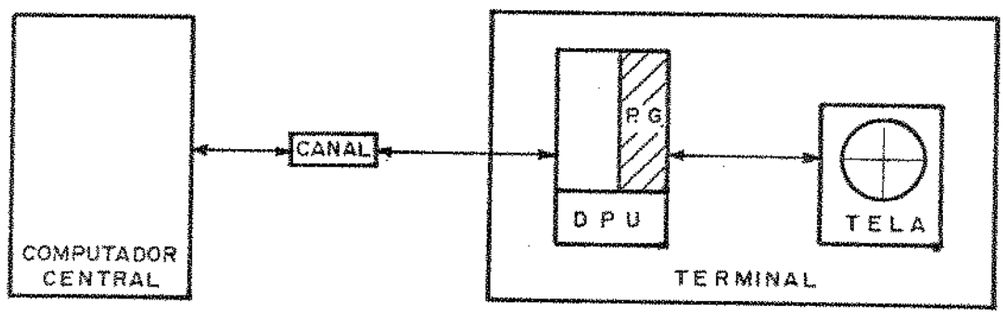


FIG. 1.7

O "light pen" é um dispositivo que possui uma célula fotosensível. Quando excitado, gera uma interrupção que trunca o processamento do programa gráfico. Deste modo temos acesso ao valor das coordenadas do ponto que gerou a interrupção, pesquisando-se onde o programa gráfico foi truncado. (Fig. 1.8)

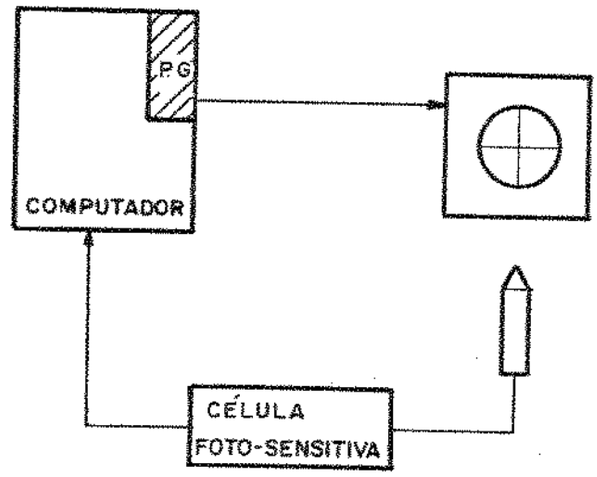


FIG. 1.8

1.3.2 "Software"

Dado um sistema "hardware" e uma certa linguagem de programação devemos escolher uma estrutura para os objetos gráficos e seus dados. Esta estrutura (e seus dados) pode ser fornecida como entrada ou então ser o resultado de cálculos realizados pelo programa de aplicação.

Há dois tipos de estruturas básicas:

a) Estrutura com funções para geração da imagem

Neste tipo de estrutura podemos definir o espaço virtual no computador onde a imagem é gerada pelas funções

$$EV: \mathbb{R} \times \mathbb{R}$$

e o espaço real na tela

$$ER: [0:k_x] \times [0:k_y]$$

sendo k_x o número de pontos possíveis na tela em abcissas e k_y o número de pontos possíveis em ordenadas.

A transformação de uma variável pertencente ao espaço virtual em uma variável pertencente ao espaço real chama-se mapeamento:

$$w \in \mathcal{R} \times \mathcal{R}$$

$$w: W \rightarrow [0:k_x] \times [0:k_y]$$

Um exemplo para utilização desta estrutura é a obtenção de gráficos.

b) Estrutura por elementos

Neste tipo de estrutura são considerados como elementos básicos:

- Primitivos, que são ponto, linha, caracter, arco e superfície;
- Itens, definidos como o conjunto ordenado de primitivos que formam as partes isoladas da figura.
- Figuras, que são conjuntos de itens que formam a imagem.

A imagem da Fig. 1.9 será formada, neste tipo de estrutura pelo programa da Fig. 1.10.

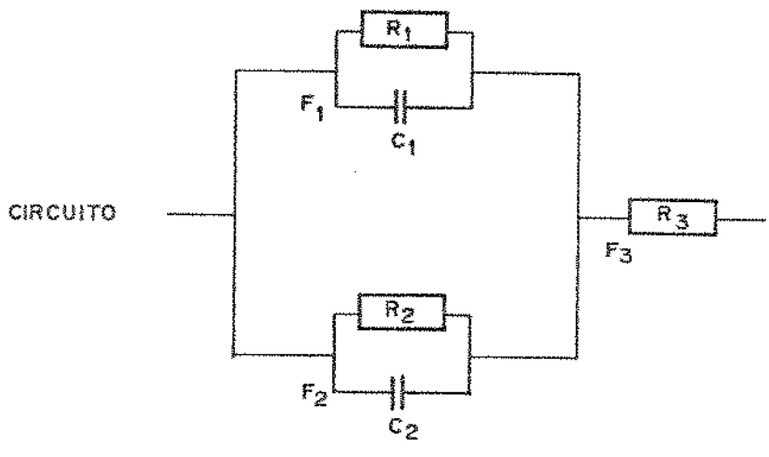


FIG. 1.9

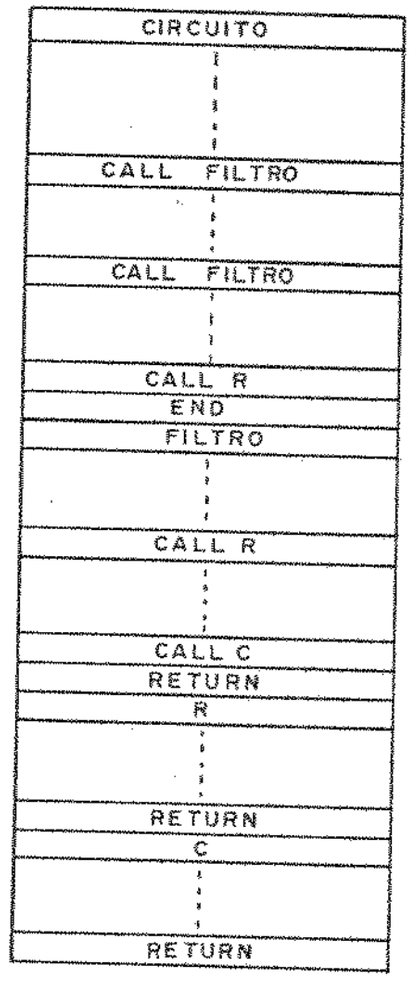


FIG. 1.10

A imagem que representa o circuito é formada por três figuras, dois filtros e uma resistência. As figuras são formadas por itens que são as resistências e capacitores, os quais, por sua vez, são conjuntos ordenados de linhas.

Uma vez definida a estrutura dos objetos gráficos, é necessário uma escolha apropriada do tipo de dados, estrutura de dados, estrutura sintática e estrutura de controle.

Ao preparar o programa de aplicação, o usuário escolherá o tipo de dados necessários para o processamento gráfico. Para problemas diferentes, o programa de aplicação solicitará diferentes tipos de dados.

O banco de dados deve ter uma estrutura que facilite a interação com o programa gráfico. Havendo uma lei de formação, não será necessário montar o mapa de correlação.

Conforme o tipo de imagem desejada, pode ser criada uma estrutura sintática que facilite a montagem do programa gráfico. Deste modo serão preparadas subrotinas específicas para geração de pontos, vetores, caracteres, arcos e gráficos que quando convenientemente chamadas, formarão a imagem desejada.

A estrutura de controle deve ser tal que comande todo o processo, desde a geração da imagem, até a sua modificação através de um comando do usuário.

Finalmente o "software" deve incluir um sistema de interação para:

a) Geração de imagens - O usuário poderá gerar imagens através dos periféricos gráficos. Para isso deverão existir subrotinas, que interpretam o comando do usuário gerando ou eliminando pontos e linhas, para formar a imagem;

b) Transformação da imagem - As transformações podem ser de dois tipos: transformações afins são aquelas que usam a matriz de transformação (rotação, translação, escalamento e perspectiva) e transformações no domínio são as que modificam a estrutura dos dados. Um exemplo é o "clipping", que elimina todos os pontos não pertencentes ao espaço real (referência [1]);

c) Construção do banco de dados - É feita através de subrotinas que interpretam os comandos do usuário, estruturando os dados fornecidos.

CAPÍTULO II - TRATAMENTO DA INFORMAÇÃO GRÁFICA PELO SISTEMA DEC

O sistema em operação consta de um computador central, PDP-10, funcionando em multiprogramação. A informação gráfica é obtida através do terminal GT-40.

2.1 Descrição do GT-40

O GT-40 é um terminal inteligente que consiste de um sistema gráfico, por tubo de raios catódicos e um mini-computador PDP-11/05.

2.1.1 O mini-computador PDP-11/05

Na configuração atual o PDP-11 possui 8k de memória com palavra de 16 bits. Sua unidade central de processamento (CPU) possui um extenso repertório de instruções com capacidade de operação com uma palavra, meia-palavra (byte) ou bit. Esta característica é muito útil na manipulação da informação gráfica.

As instruções podem ser divididas em:

a) Instruções de dois operandos:

ADD x,y	soma	$y \leftarrow y + x$
SUB x,y	subtração	$y \leftarrow y - x$
CMP x,y	comparação	
MOV x,y	substituição	$y \leftarrow x$
BIS x,y	soma lógica	$y \leftarrow y \vee x$
BIT x,y	produto lógico	$y \leftarrow y \cdot x$

Com exceção da soma e subtração, as demais instruções possuem a propriedade de operar com "bytes".

Exemplo:

MOVB x,y

b) Instruções de um operando:

INC x	incrementa	$x \leftarrow x + 1$
DEC x	decrementa	$x \leftarrow x - 1$
CLR x	faz igual a zero	$x \leftarrow 0$
COM x	complementa	$x \leftarrow \overline{x}$
TST x	testa com zero	
ADC x	soma "carry"	$x \leftarrow x + c$
SBC x	subtrai "carry"	$x \leftarrow x - c$
ASR x	deslocamento de 1 "bit" à direita	
ASL x	deslocamento de 1 "bit" à esquerda	
ROR x	rotação de 1 bit à direita	
ROL x	rotação de 1 "bit" à esquerda	
SWAB x	troca de "bytes"	

Todas estas instruções operam com "bytes" também.

c) Instruções de quebra de sequência:

Existem várias instruções de quebra de sequência que testam diferentes condições; se o resultado de uma operação é positivo, negativo ou zero, se houve ou não "overflow", se houve ou não "carry".

d) Instruções para subrotinas:

JRS	R_n, SUB	execute a subrotina
RTS	R_n	retorne da subrotina

e) Instruções gerais:

A instrução HALT trunca o processamento do programa.

A instrução WAIT trunca o processamento do programa até que ocorra uma interrupção. Terminado o processamento da subrotina de serviço de interrupção, após a execução da instrução RTI (retorno da interrupção), o processamento do programa inicial terá continuidade.

Qualquer tipo de endereçamento usado nas instruções faz uso de um dos registros existentes, sendo seis destes de utilização geral, um apontador do "stack" (SP) e um contador de programa (PC).

A capacidade de formar pilhas do tipo "Last in - First out" ("stacks") dão grande versatilidade na programação. No salto para uma subrotina ou interrupção, o endereço de volta é colocado no "stack" cujo endereço é apontado pelo registro SP.

2.1.2 O sistema gráfico

Sendo o GT-40 um terminal gráfico, possui uma unidade de processamento gráfico (DPU) capaz de executar um conjunto de instruções específicas, colhendo comandos e dados da memória e executando esses comandos com todos os cálculos necessários para a geração de um vetor ou de um caracter.

Estas instruções permitem ao programador ter várias opções na forma da informação gráfica. Os vetores podem ser contínuos ou tracejados, os caracteres podem

ser letras maiúsculas, minúsculas, itálicas ou normais, caracteres especiais e símbolos matemáticos. Além disso, tanto os vetores como os caracteres podem ter diferentes níveis de luminosidade e a opção de cintilar.

São cinco as instruções gráficas:

1) Modo Gráfico:

"bit"	
15	"1" indica instrução
14 a 11	0000 - caracter 0001 - vetor curto 0010 - vetor longo 0011 - ponto 0100 - gráfico em x 0101 - gráfico em y 0110 - ponto relativo
10	"1" permite a entrada dos "bits" 9 a 7 no registro de intensidade
9 a 7	valor da intensidade (0 a 7)
6	"1" permite a entrada do "bit" 5 no registro de interrupção
5	"1" permite interrupção do "light pen"
4	"1" permite a entrada o "bit" 3 no registro de cintilamento
3	"1" permite cintilamento
2	"1" permite a entrada dos "bits" 1 e 0 no registro de linhas
1 a 0	00 - linha sólida 01 - linha tracejada longa 10 - linha tracejada curta 11 - linha ponto - traço

Esta instrução define várias opções na forma da informação gráfica.

a) Caracter - CHAR: cujo formato da palavra de dado é representado na Fig. 2.1.

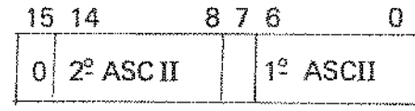


Fig. 2.1

b) Vetor curto - SHORTV: a palavra de dado do vetor curto deverá conter o deslocamento em abcissa e ordenada no formato da Fig. 2.2.

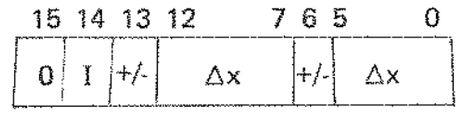


Fig. 2.2

O deslocamento é dado em valor absoluto; os bits 13 e 6 indicarão se este é positivo ou negativo. Havendo disponibilidade de apenas 6 bits para Δx e Δy, o maior vetor que obtemos neste modo gráfico será de 63 unidades de tela em abcissa e ordenada.

O bit 14 (I), quando igual a "1" indica que o vetor terá luminosidade (INTx); quando zero o vetor não aparecerá na tela.

c) Vetor longo - LONGV: Os dados são os deslocamentos Δx e Δy em valor absoluto, ocupando duas palavras consecutivas (Fig. 2.3).

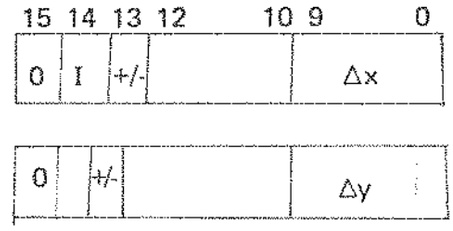


Fig. 2.3

O GT-40 possui 1024 unidades de tela em abcissa e 768 unidades em ordenadas. Devido a este fato, são necessários dez bits para os deslocamentos Δx e Δy, a fim de que este modo gráfico descreva qualquer vetor na tela.

d) Ponto - POINT: a palavra de dado deverá conter o valor da abcissa e ordenada do ponto em unidades absolutas de tela (Fig. 2.4).

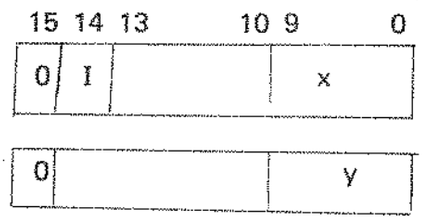


Fig. 2.4

e) Gráfico em x - GRAPHX: Este modo supõe que o incremento da ordenada seja constante, sendo necessário apenas o valor da abcissa. (Fig. 2.5).

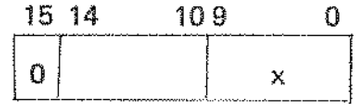


Fig. 2.5

f) Gráfico em y - GRAPHY: Do mesmo modo, é suposto um incremento constante em abcissa (Fig. 2.6).

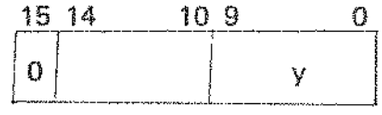


Fig. 2.6

g) Ponto Relativo - RELATV: Difere do vetor curto apenas por traçar um ponto no lugar de um vetor.

Na mesma instrução podemos ter ainda opções:

- em níveis de luminosidade, com escolha de oito níveis, de zero a sete (INT);
- na escolha de cintilamento ou não (BLKON, BLKOFF);
- na escolha do tipo de linha; podendo ser sólida, tracejada longa ou curta e ponto traço (LINE 0 a LINE 3);
- na escolha da utilização do "light pen" (LPON, LPOFF).

2) Instrução de Salto - DJMP: Exige duas palavras. A primeira contém o código da instrução, a segunda o endereço da próxima instrução (Fig. 2.7).

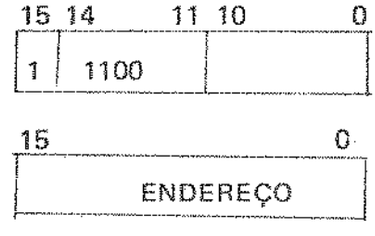


Fig. 2.7

3) Não operação - NOP: Fig. 2.8.

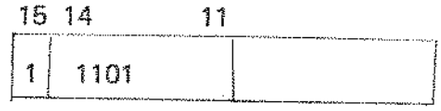
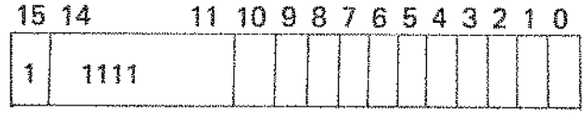


Fig. 2.8

4) Carregamento do registro A - STATSA: Fig. 2.9.



"bit"	Comentário
10	"1" trunca o processamento gráfico
9	"1" permite entrada do bit 8 no registro de interrupção
8	interrompe ou não quando trunca o processamento
7	"1" permite entrada do bit 6 no registro de intensidade
6	ilumina ou não o ponto de interação com o "light pen"
5	"1" permite entrada do bit 4 no registro de forma
4	1 - itálicas 0 - normal
2	trunca o processamento da DPU e reinicializa em sincronismo com a frequência da linha

Nesta instrução podemos ter opção:

- de truncar o processamento do programa gráfico (DSTOP), seguido ou não de interrupção (SINON, SINOFF);
- de permitir que o ponto de interação com o "light pen" seja intensificado ou não;
- na escolha do tipo de caracter, itálico ou normal (ITALO ou ITAL 1).

5) Carregamento do registro de estado B - STATSB

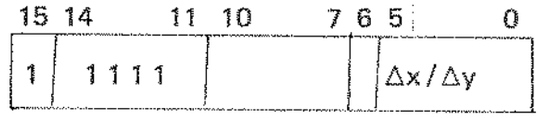


Fig. 1.10

Esta instrução contém 6 bits onde estará o incremento constante, em x ou y, para o modo gráfico GRAPHX e GRAPHY.

O "bit" 6 quando igual a "1" permite a entrada dos bits 5 a 0 no registro de incremento.

O programa gráfico é iniciado por uma instrução seguida de quantos dados forem necessários. Uma vez dadas as opções, não será necessário repeti-las na próxima instrução, a menos que haja modificações.

Exemplo de programa gráfico:

```
ST:  POINT + INT 4 + LPOFF + BLKOFF + LINEO;  ponto de intensidade 4, não
                                              ; permite interrupção do "light pen"
                                              ; sem cintilação, linha sólida
     500                                      ; ponto sem luminosidade
     500                                      ; com x= 500 e y = 500
     LONGV                                    ; vetor longo
     200 + INTx                               ; Δx =200 , Δy =0
     0                                        ; com  luminosidade
     0 + INT x                                ; Δx = 0 , Δy = 200
     200                                      ; com luminosidade
     200 + INT x + MINUS                      ; Δx = - 200, Δy = 0
     0                                        ; com luminosidade
     0 + INT x                                ; Δx = 0  Δy = - 200
     200 + MINUS                              ; com luminosidade
     DJMP                                     ; salto para
     ST                                       ; recomeçar
```

2.1.3 Sistema CPU+DPU

A interação entre CPU, DPU e memória pode ser vista na Fig. 2.1.1.

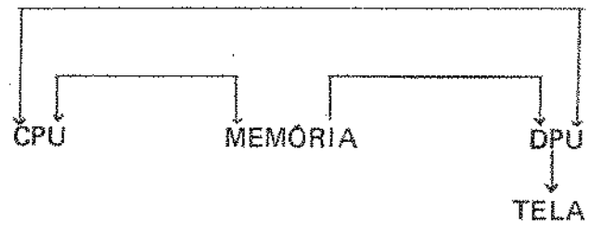


Fig. 2.11

O GT-40 deve ser visto como um computador que possui dois processadores programáveis separadamente. A CPU colhe instruções da memória e as executa lendo ou modificando os dados armazenados na mesma memória. A DPU colhe instruções e dados em posições contíguas da memória que formam o programa gráfico. Não há instrução gráfica que modifique o programa gráfico; a comunicação DPU-memória é unidirecional.

A DPU é inicializada através de instruções executadas pela CPU; a partir deste ponto os dois processadores seguem seus programas independentemente.

```
MOV     # 500, SP      ; inicializa o stack
MOV     ST,DPC         ; carrega o contador de programa
WAIT                   ; espera interrupção
BR      . - 2          ; volta a esperar interrupção
```

Este programa é executado pela CPU e inicializa a DPU com a instrução MOV ST, DPC. A partir deste momento a DPU passa a executar seu próprio programa

```
ST:  POINT + INT4 + LPOFF + BLKOFF + LINEO
     500
     500
     LONGV
     200 + INT x
     0
```

```
0 + INT x
200
200 + INT x + MINUS
0
0 + INT x
200 + MINUS
DSTOP + SINON          ; trunca o processamento da
DJMP                   ; DPU e permite interrupção
ST.
```

Ao ser executada a instrução gráfica DSTOP+SINON, a CPU toma conhecimento da interrupção e faz um salto automático para a subrotina de serviço da interrupção. Esta subrotina de serviço se encontra em uma posição de memória especificada no vetor de interrupção. O vetor de interrupção possui duas posições fixas de memória para cada periférico; a primeira posição deverá conter o endereço da subrotina de serviço, a segunda o novo estado da CPU durante a execução da subrotina.

A subrotina de serviço de interrupção é feita pelo usuário.

Exemplo:

```
. = DPUI          ; posição do vetor de interrupção da DPU
  SUB             ; endereço da subrotina de serviço
  ESTADO          ; estado da CPU

SUB:  MOV  #1,DPC ; dá continuidade ao processamento da DPU
      RTI         ; retorno da interrupção
```

2.1.4 "Light-Pen"

Uma das vantagens do tratamento da informação gráfica por tubo de raios catódicos do tipo "refreshing" é a possibilidade do usuário modificar o processo em execução, tomando decisões a partir da informação gráfica obtida e, interagindo diretamente na tela, obter a modificação necessária.

No GT-40 o periférico gráfico de interação é o "light pen". A interação ou não interação do "light pen" com o processo é programada através das instruções gráficas.

O "light pen" é um instrumento sensível à luz que quando excitado

trunca o processamento da DPU e gera uma interrupção a ser processada pela CPU.

Deste modo teremos acesso a três dados importantes:

- O contador de programa da DPU(DPC), que mostrará onde o programa gráfico foi interrompido;
- Uma vez truncado o processamento da DPU, os registros x e y conterão as coordenadas do ponto onde o programa gráfico parou no instante da interrupção, ou seja, as coordenadas do ponto que excitou o "light pen".

A interrupção, gera um salto automático para a subrotina de serviço do "light pen". Esta subrotina é elaborada pelo programador de modo a tomar decisões a partir dos dados disponíveis e assim modificar o processo localmente, ou então enviar uma mensagem ao computador central para que este tome as decisões.

Exemplo de subrotina de serviço do "light pen":

```
            RX = 172004          ; registro que contém a coordenada x da tela
            Ry = 172006          ; registro que contém a coordenada y da tela
            P100S = 17 5614      ; registro que contém o estado do computador
                                 ; central para receber dados
            P100B = 175616       ; registro que contém o dado a ser
                                 ; enviado ao computador central

SUB:        MOV  Dx,P100C        ; envia a
            JSR  PC, OUTPUT      ; coordenada x
            MOV  Dy, P100C       ; envia a
            JSR  PC, OUTPUT      ; coordenada y
            RTI
P100C:      0
OUTPUT:     TSTB P100S           ; testa se o computador central
            BPL  OUTPUT          ; está receptivo
            MOV P100C, P100B     ; envia o dado
            RST  PC
```

2.1.5 Interface

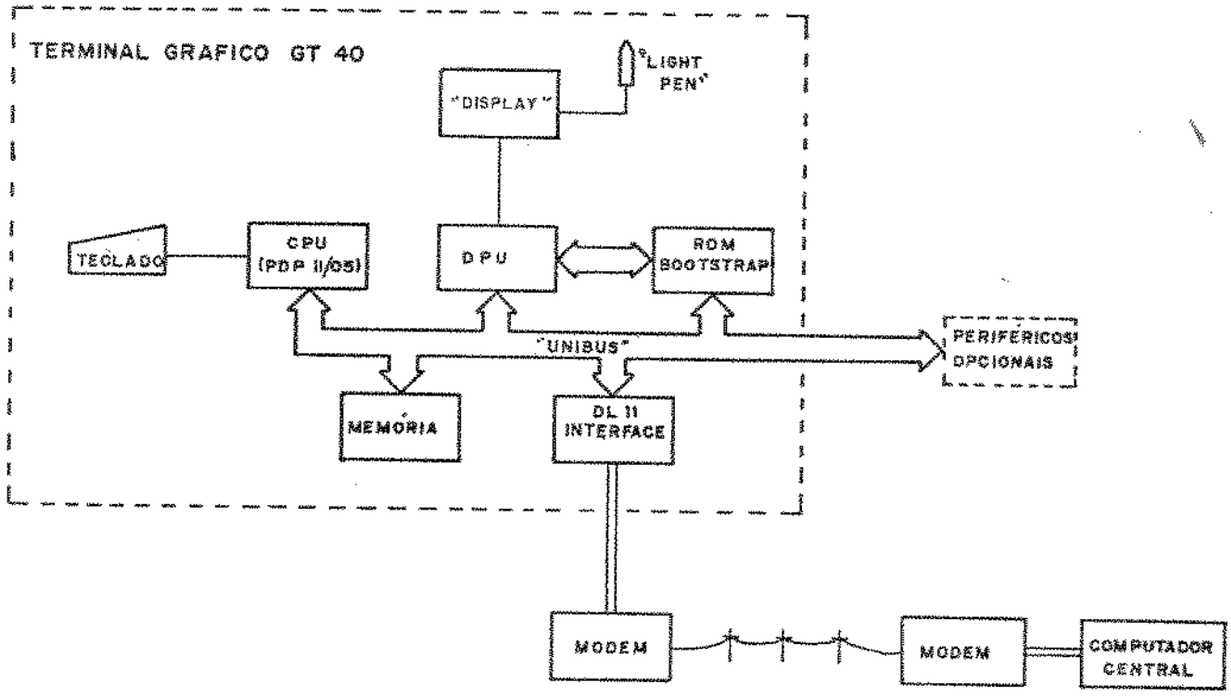
A ligação entre o GT-40 e o computador central é feita por uma interface assíncrona que transfere e recebe oito bits via "modem".

A velocidade de transmissão é de 1500 bauds.

Existe a possibilidade do oitavo bit aparecer como teste de paridade. Esta opção é programada no computador central.

2.1.6 "UNIBUS"

Uma das principais características do PDP-11 é a "UNIBUS", uma barra bidirecional que liga todos os componentes do computador como se fossem periféricos.



A comunicação entre dois periféricos é feita através da barra com uma relação "master/slave". O periférico "master" tem o controle da barra enquanto comunica com o periférico "slave". Esta propriedade é importante pois permite uma fácil e rápida comunicação entre os mesmos.

A cada periférico está associado um nível de prioridade para requisição do controle da barra. A CPU é o único periférico que tem o nível de prioridade programável, podendo o usuário escolher um entre oito níveis (0 a 7). Se a CP U estiver em um nível superior a um periférico, este não consegue o controle da barra; é um artifício para impedir interrupção proveniente de determinado periférico.

Quando um periférico ganha o controle da barra ele realiza uma ou mais transferências de dados ou requisita uma interrupção. A transferência de dados é NPR ("non processor request"), ou seja, por um canal de acesso direto à memória, não havendo interferência do processador. O periférico "master" não passa o controle da barra antes de haver completado uma transferência de dados ou uma interrupção. De modo geral, em cada bloco, é transferido um único dado, visto que ao se manter o controle da barra não se permite que outros periféricos façam uso da mesma.

2.2 Software

Sendo o PDP 11/05 um computador pequeno, não possui facilidades como, sistema operacional ou compiladores. Porém ele pode ser programado em assembler pois o compilador central possui o compilador necessário.

O GT-40 possui um "bootstrap" em "hardware" que inicializa seu funcionamento como terminal. Estando este funcionando como terminal, o computador central possui um carregador que transfere o programa em assembler compilado para a memória do PDP-11.

Deste modo, a DEC preparou dois programas para utilização do terminal gráfico. Cada um deles consta de um programa em assembler a ser carregado no PDP-11 e uma série de subrotinas em FORTRAN–IV que o usuário utiliza no seu programa para montar a informação gráfica.

CAPÍTULO III - ALGORÍTIMO PROPOSTO PARA O TRATAMENTO DA INFORMAÇÃO GRÁFICA

3.1 Introdução

Os dois programas fornecidos pela DEC para o tratamento da informação gráfica são: GT40 A/I e BOOK. O segundo é superior ao primeiro em dois aspectos:

- Faz uso do "light pen", sendo possível assim a desejada interação do usuário com o processo;
- Seu programa em assembler a ser carregado no PDP-11 do GT40 é muito menor, havendo portanto mais espaço para o programa gráfico.

O BOOK é um programa completo; faz uso de todos os recursos do GT40 para a realização da informação gráfica. A imagem é montada pelo usuário através de chamadas sucessivas das subrotinas do BOOK.

3.2 Book

O BOOK está baseado em 4 tipos de estruturas: "Pictures", "Figures", "Graphs" e "Tables".

A "Picture" é composta de pontos (POINT), vetores longos (LONG V) e chamadas para outras "Pictures", "Figures", "Graphs" e "Tables". Cada elemento da "Picture" ocupa três linhas, ou seja, três palavras na memória do computador.

Um ponto é inserido no programa gráfico através da chamada.

CALL DOT (x,y)

o BOOK montará no programa gráfico

POINT
x
y

As chamadas para outras "Pictures", "Figures", "Graphs" e "Tables" são como saltos para subrotinas; terminada a execução da estrutura chamada a DPU volta à "Picture" original e executa a instrução seguinte.

A "Figure" é composta de vetores curtos, os "Graphs" de gráficos em x ou em y e as "Tables" de caracteres. Em todas as três estruturas cada elemento ocupa somente uma linha.

Sendo o BOOK um programa completo, seu programa em assembler é complexo, ocupando bastante memória no PDP-11. O programa gráfico gerado não é otimizado; uma sequência de pontos seria montada na forma

POINT
x1
y1
POINT
x2
y2

havendo portanto instruções supérfluas que diminuem a capacidade de armazenamento de informação gráfica na memória.

Devido a estes fatos o BOOK não pode executar gráficos de muitos pontos por não haver memória suficiente para conter todos os dados.

3.3 "Software" Proposto

Para resolver o problema de gráficos de muitos pontos é proposto um algorítmo com as seguintes restrições:

- O programa em assembler residente no GT40 deve ser simples de modo a não ocupar muita memória;
- O programa gráfico gerado deve ser otimizado, não havendo repetições desnecessárias de instruções.

Sendo o programa em assembler simples, não há possibilidade de variação no tipo de informação; o programa tratará somente de gráficos. Serão usadas somente duas instruções: POINT e SHORTV', sendo os dados colocados em sequência após a instrução SHORT V de modo a não haver repetição desnecessária de instrução.

Um teletipo (LA-30) da DEC foi acoplado ao GT40 em funcionamento na FEC, de modo que a função de entrada e saída do terminal seja feita pelo teletipo e não pela tela. Para isto foi desenvolvido um programa a partir do "bootstrap" original.

A partir deste "bootstrap" modificado foi desenvolvido um monitor que

será a parte residente no GT40 do programa de gráficos. A parte a ser executado no computador central (PDP-10) foi feita em FORTRAN IV e consta da subrotina GRAF, que será chamada no programa do usuário.

Um dos problemas de um terminal ligado a um computador central à distância é a lentidão da comunicação. Como solução para o problema, foi estudada a possibilidade de se acoplar um disco ao PDP-11. Este disco será de muita utilidade, para o caso do programa gráfico gerado no computador central ser muito grande. O computador central transmitirá todo o programa gráfico, que ficará gravado no disco, e um programa local o executará por partes. Esta operação será muito mais rápida que a comunicação direta com o computador central, deste modo será possível obter gráficos razoavelmente estáveis.

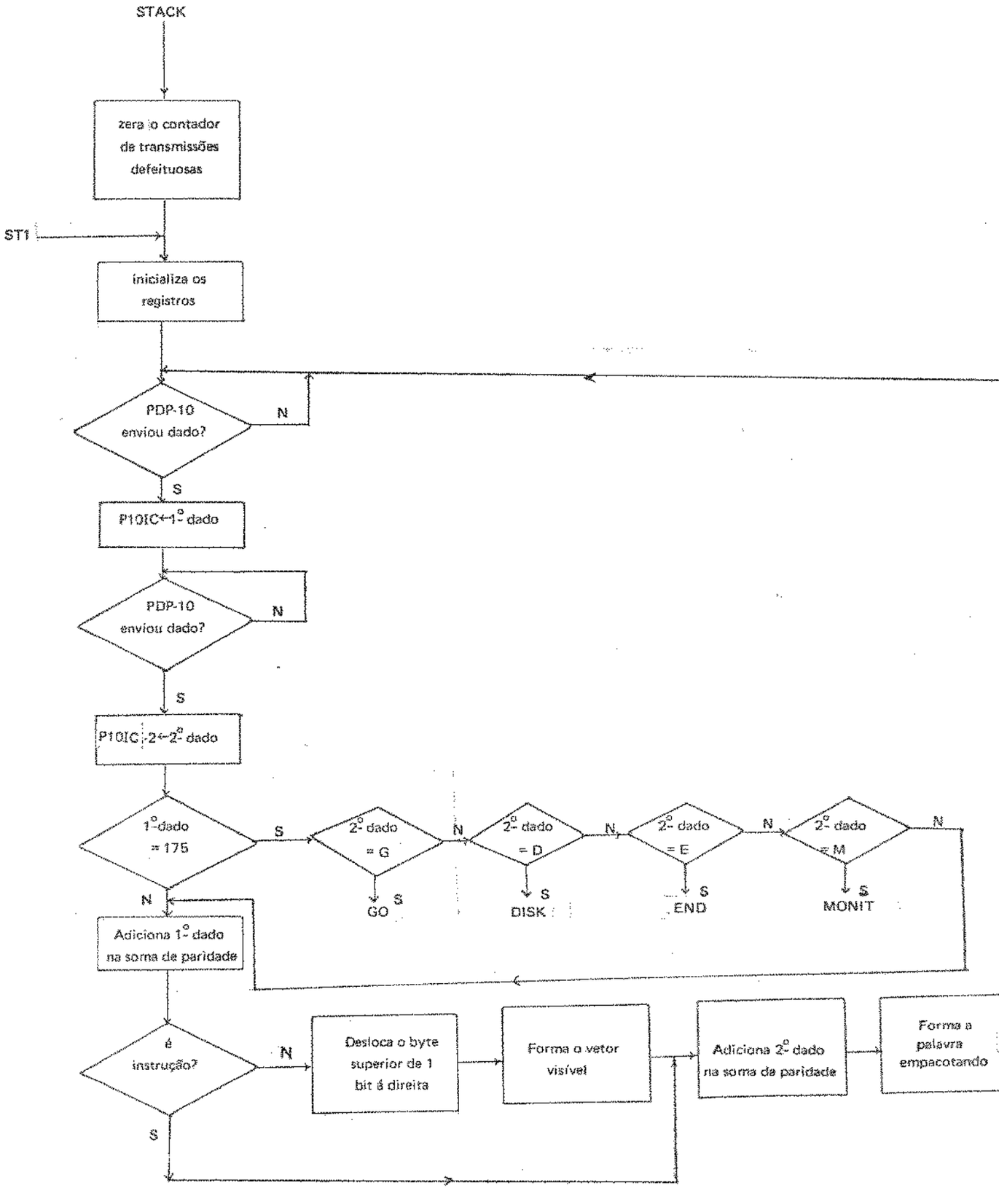
Foi escolhido um disco muito rápido (RJS04-DEC) com tempo de transferência de 4μ, e sua utilização incorporada ao monitor e subrotina GRAF.

3.3.1 Monitor

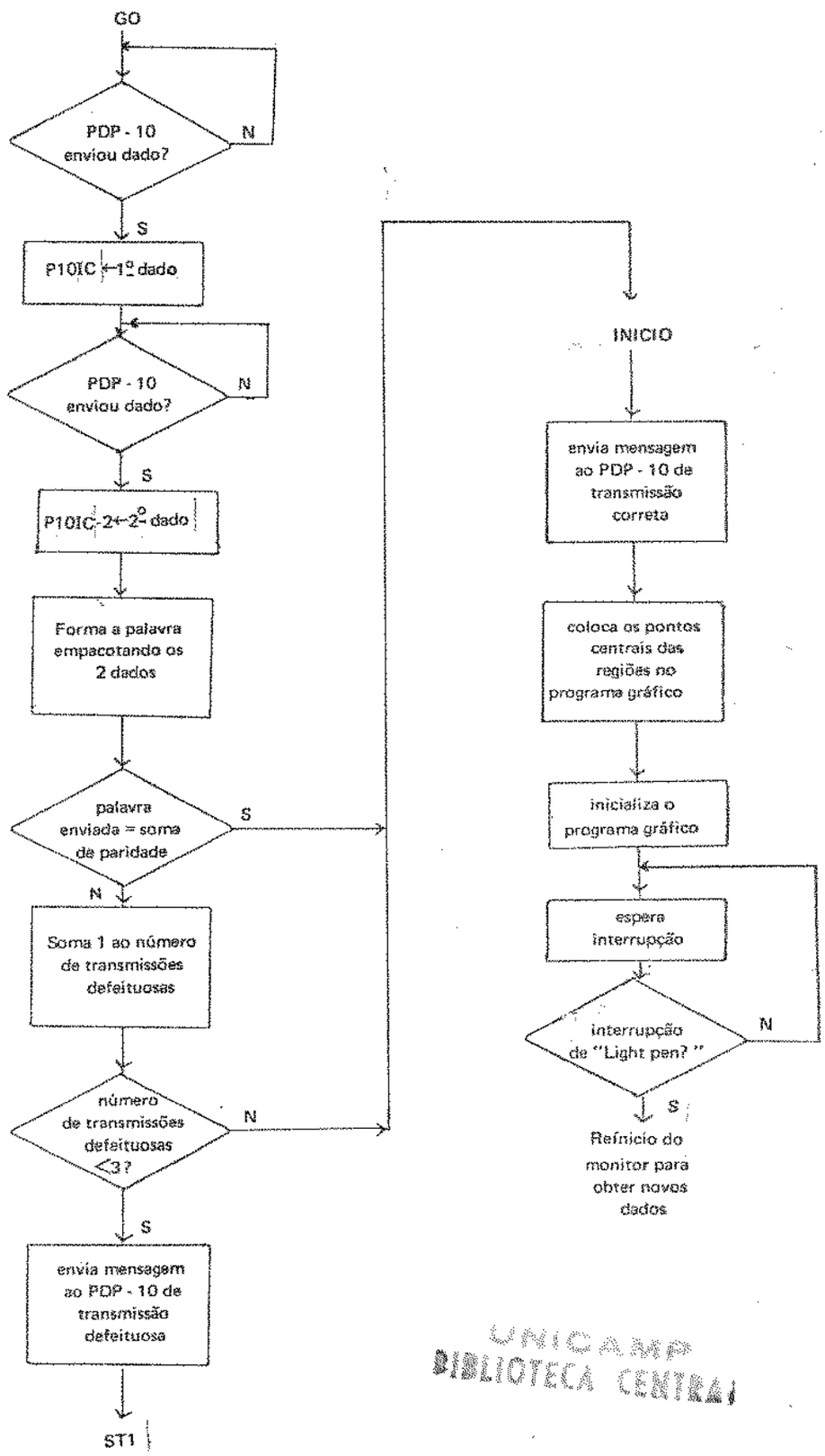
Com a utilização do disco, o monitor terá várias chamadas para diferentes segmentos, conforme a dimensão do programa gráfico. Se a dimensão for tal que a memória disponível seja suficiente, o monitor gravará o programa gráfico na memória e o executará imediatamente. Caso contrário, o monitor gravará os blocos do programa gráfico no disco e chamará o segmento que o executa por partes.

Antes de ser chamada a subrotina GRAF no programa do usuário, o monitor estará executando o segmento que inicializa o GT40 como terminal através do teletipo, ou seja, estará executando o "bootstrap" modificado.

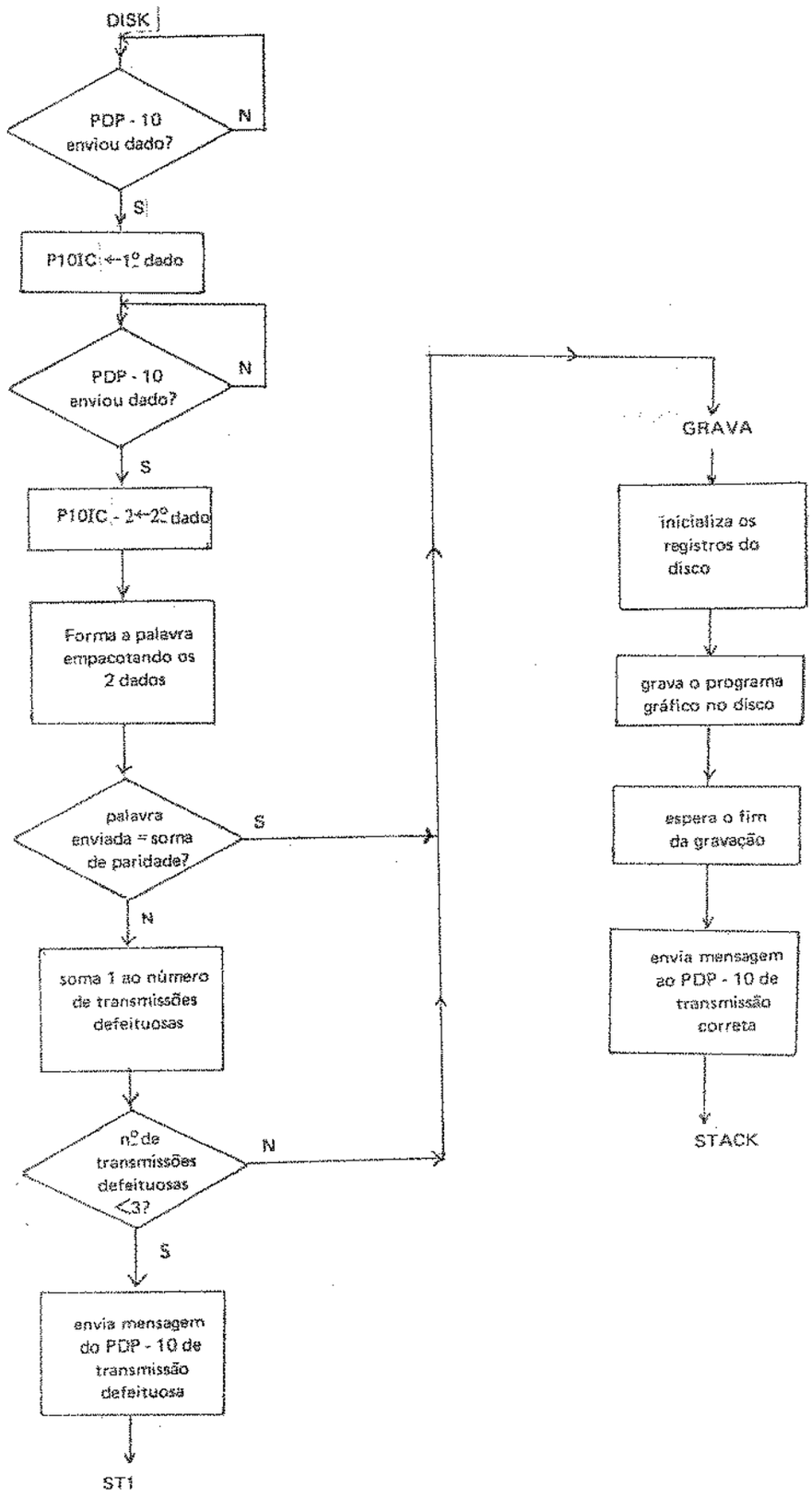
Ao ser chamada, a subrotina GRAF envia uma mensagem ao monitor fazendo com que este salte para o segmento que monta o programa gráfico na memória.


STACK
zera o contador de transmissões defeituosas
ST1
inicializa os registros
PDP-10 enviou dado?
N
S
P10IC←1° dado
PDP-10 enviou dado?
N
S
P10IC-2←2° dado
1° dado = 175
S
2° dado = G
N
2° dado = D
N
2° dado = E
N
2° dado = M
N
S
GO
S
DISK
S
END
S
MONIT
N
Adiciona 1° dado na soma de paridade
é instrução?
N
Desloca o byte superior de 1 bit á direita
Forma o vetor visível
Adiciona 2° dado na soma de paridade
Forma a palavra empacotando
S

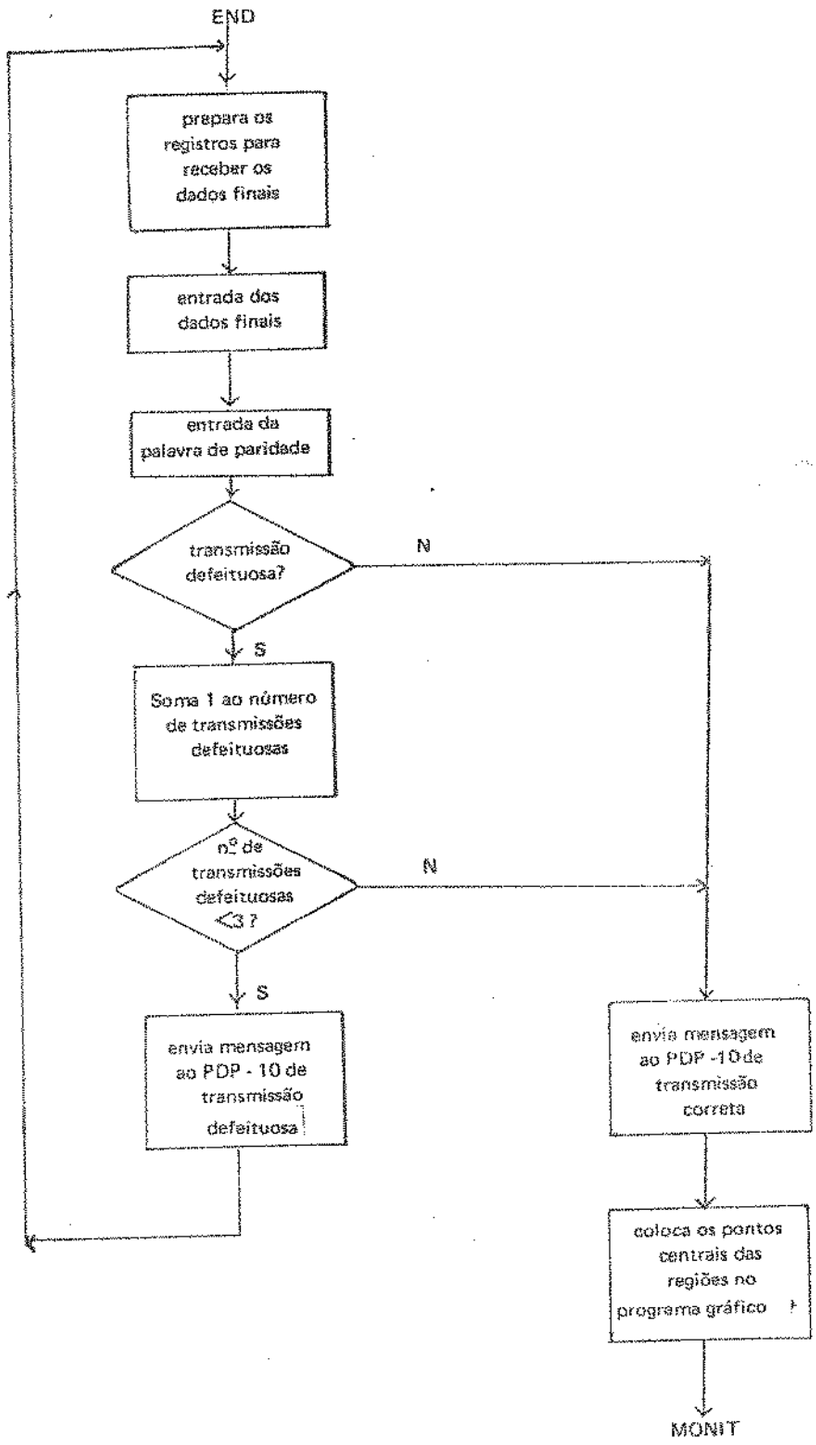
Se a memória disponível no GT40 for suficiente para conter todo o programa gráfico, a subrotina GRAF enviará uma mensagem ao monitor fazendo-o saltar para o segmento que executa o programa gráfico. Antes de começar a execução, o monitor testa se os dados enviados estão corretos. Caso não estejam, há uma nova tentativa para um carregamento correto. Serão feitas três tentativas.



Se a memória não for suficiente, a mensagem enviada será de comando para gravar o programa gráfico no disco. Antes de gravar o monitor testa se os dados enviados estão corretos.



Após enviar o último bloco do programa gráfico para gravar no disco, a subrotina GRAF envia uma mensagem de fim de transmissão. O monitor recebe então os dados finais e salta para o segmento que executa o programa gráfico por partes.

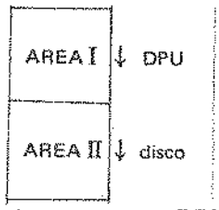


O segmento do monitor que executa o programa gráfico por partes chama-se MONIT.

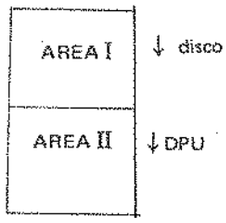
Quando a memória é insuficiente para conter o programa gráfico, este é dividido em um número par de blocos. A última palavra de cada bloco é a instrução DSTOP, que trunca o processamento da DPU, com opção de interrupção. A subrotina de serviço da interrupção é uma instrução que dá continuidade à execução da DPU.

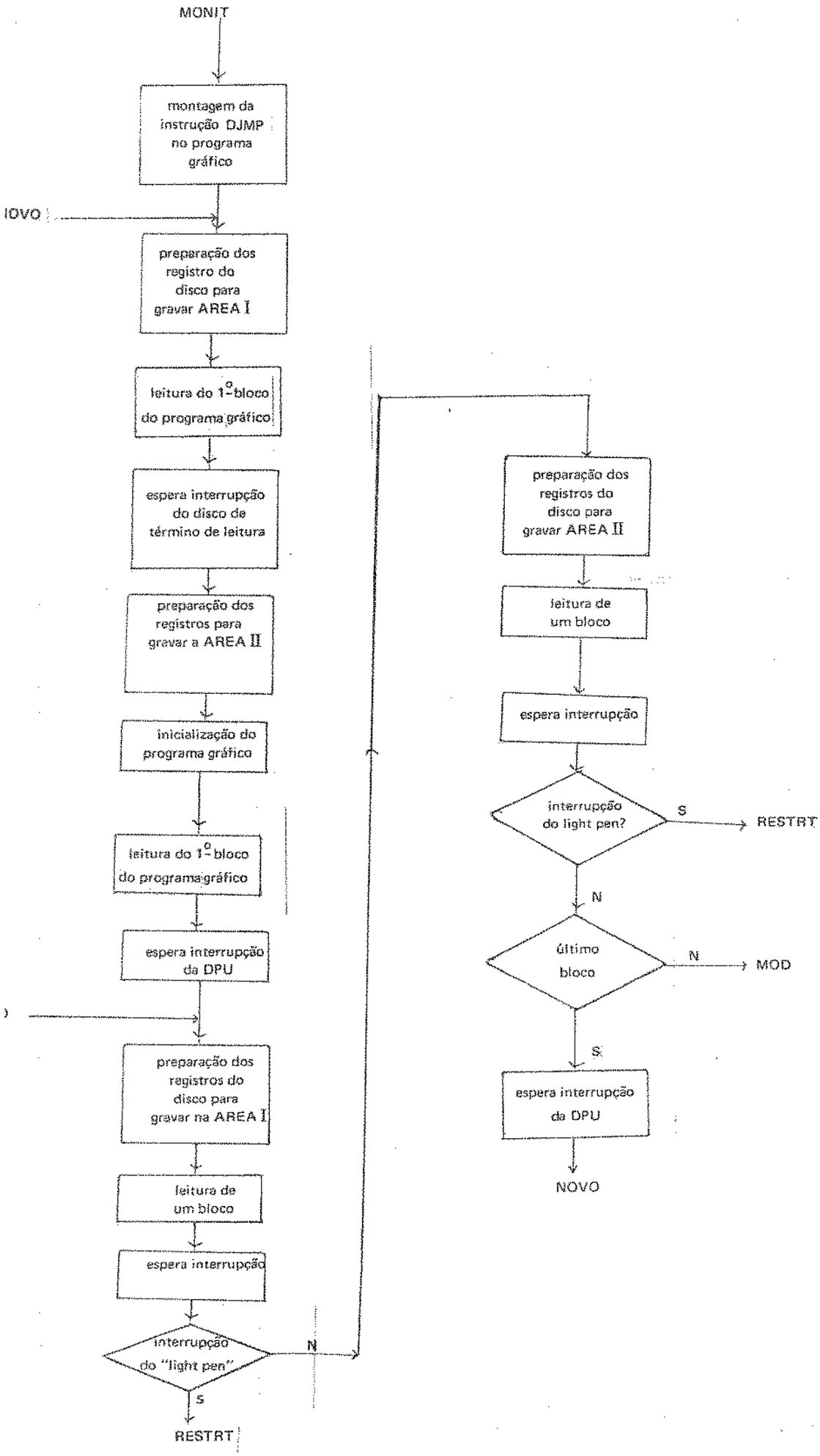
Na memória estão sempre dois blocos. Inicialmente é lido do disco o primeiro bloco do programa gráfico e gravado na memória na posição AREA I. Durante a leitura o monitor fica parado na instrução WAIT. Terminada a leitura, o disco gera uma interrupção que dará prosseguimento ao monitor. Este inicializa a DPU para executar a ÁREA I e lê do disco o bloco seguinte do programa gráfico, gravando-o na posição ÁREA II. A partir deste passo não será mais permitida interrupção do disco.

O disco é muito mais rápido que a DPU. A velocidade de transferência do disco é da ordem de 4μs por palavra. A DPU necessita de 23μs para executar um vetor curto. Terminada a leitura, o monitor espera, na instrução WAIT a interrupção que ocorrerá quando a DPU executar a instrução DSTOP.



No próximo passo, a DPU executará a ÁREA II e o monitor gravará o próximo bloco do programa gráfico na posição ÁREA I. O monitor continuará nesta sequência até que termine os blocos do programa gráfico, quando a DPU será truncada por uma instrução DSTOP sem opção de interrupção. O monitor voltará então ao início deste segmento para ler do disco o primeiro bloco do programa gráfico.




MONIT
montagem da instrução DJMP no programa gráfico
IOVO
preparação dos registro do disco para gravar AREA I
leitura do 1°-bloco do programa gráfico
espera interrupção do disco de término de leitura
preparação dos registros para gravar a AREA II
inicialização do programa gráfico
leitura do 1°-bloco do programa gráfico
espera interrupção da DPU
preparação dos registros do disco para gravar na AREA I
leitura de um bloco
espera interrupção
interrupção do "light pen"
N
S
RESTRT
preparação dos registros do disco para gravar AREA II
leitura de um bloco
espera interrupção
interrupção do light pen?
S
RESTRT
N
último bloco
N
MOD
S
espera interrupção da DPU
NOVO

Como a finalidade do trabalho é a obtenção da imagem de um gráfico de muitos pontos, foi feita uma paginação para que, por meio do "light pen", seja possível a ampliação de uma parte do gráfico, para melhor visualização de detalhes.

A tela foi dividida em regiões. Inicialmente todas as regiões aparecem na tela formando o gráfico completo. Cada região possui um ponto central de luminosidade intermitente, sensível ao "light pen". Tocando este ponto, ocorre uma interrupção e o monitor salta para a subrotina de serviço do "light pen". Esta subrotina foi elaborada para detetar qual a região escolhida e enviar ao computador central, após o que, o monitor volta ao seu início para receber os dados da nova imagem.

3.3.2 Subrotina GRAF

A parte a ser executada no computador central é a subrotina GRAF, chamada no programa do usuário. Uma vez que o gráfico final pode ser constituído de vários subgráficos, a subrotina GRAF será chamada tantas vezes quantos forem os subgráficos.

Os parâmetros são:

IFIRST - igual a "1" se for a primeira chamada
ILAST - igual a "1" se for a última chamada
N - número de pontos do subgráfico
x - matriz que contém as abcissas
y - matriz que contém as ordenadas
xMIN - abcissa mínima do gráfico total
yMIN - ordenada mínima do gráfico total
xMAX - abcissa máxima do gráfico total
yMAX - ordenada máxima do gráfico total

As matrizes x e y devem conter os pontos já ordenados para a construção dos gráficos. A subrotina usa os pontos máximos e mínimos para escalar o gráfico, de modo que a tela seja inteiramente aproveitada.

Como o gráfico será dividido em regiões, a subrotina calcula inicialmente as abcissas e ordenadas que definirão as fronteiras entre as regiões.

3.3.2.1 Montagem do Programa Gráfico

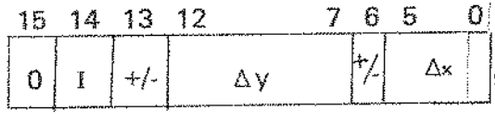
O programa gráfico é inteiramente montado pela subrotina, inclusive as instruções. Estas estão definidas no início do programa pelo octal correspondente ao seu código de máquina.

Lembrando que estas instruções ocupam uma palavra de 16 bits no PDP-11 e que a interface transmite oito bits de cada vez, vemos que a instrução deve ser desmontada em duas palavras; uma conterá os oito bits (byte) superiores e a outra os oito inferiores. Esta operação é realizada pela subrotina MONT.

Deste modo a transmissão do programa gráfico será feita sempre por pares de dados. Para isso foram definidas duas matrizes LX e LY que conterão o programa gráfico. Na transferência de dados será transmitido primeiro o dado contido em LX depois o dado contido em LY, sempre nesta sequência, para formar os pares de bytes que o monitor montará em uma palavra de 16 bits.

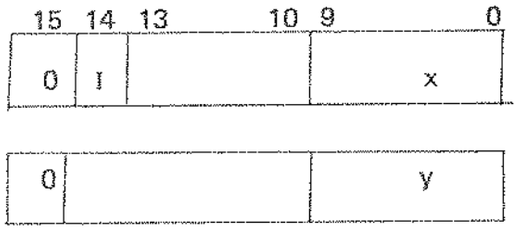
Se o par de dados for uma instrução LX conterá o byte superior e LY o byte inferior. O monitor somente empacotará os dois bytes.

Para o caso dos vetores, LX conterá o deslocamento ΔX e LY conterá ΔY. Lembrando o formato da palavra de dado do vetor curto, vemos que o monitor deverá primeiro deslocar o byte superior de 1 bit à direita e depois empacotar. O bit 14 será feito igual a "1" para o vetor ser visível.



O monitor deteta os dois diferentes casos pelos bit 15. Se a palavra é uma instrução o bit 15 é **um**, se a palavra é um dado o bit 15 é **zero**. Isto vai gerar um problema na montagem de um ponto.

Os dados de uma instrução POINT são pares de palavras como abaixo. Como o bit 15 é zero, o monitor fará o deslocamento antes de empacotar. Foi necessário então desmontar a palavra de forma diferente. A subrotina MONTPT divide o dado (X ou Y) em sete bits inferiores que estarão na matriz LY e os três superiores restantes que estão em LX.



3.3.2.2. Paginação

Para uma possível ampliação em busca de detalhes a tela foi dividida em regiões; do mesmo modo o gráfico foi paginado.

As matrizes LX e LY são dimensionadas na forma LX(M,I,J) e LY(M,I,J). As variáveis I e J indicarão a página e a variável M a posição da palavra (linha) dentro da página.

Na montagem de um vetor, quando o ponto estudado não pertencer à mesma página do ponto anterior, três casos podem ocorrer:

1. A página varia somente em abcissa
2. A página varia somente em ordenada
3. A página varia em abcissa e ordenada

Em todos os casos deve-se considerar que existirão uma ou mais novas páginas. Cada nova página será inicializada com um ponto, que terá as coordenadas da fronteira entre as páginas, seguido de uma sequência de vetores. Para não ocorrer a destruição de uma página já inicializada, a subrotina BUSCAM procurará a primeira posição não utilizada na nova página (variável M).

A última subrotina usada é a INTER, que fará a interpolação linear, quando o deslocamento ΔX ou ΔY de um vetor for maior que 63 unidades de tela. Isto porque a palavra de dado do vetor curto tem somente 6 bits para cada deslocamento ΔX e ΔY.

3.3.2.3 Segmentação do programa gráfico

O monitor ocupa aproximadamente 1500 posições de memória dos 8k disponíveis no PDP-11. Descontadas as 500 posições da zona zero, onde estão os vetores de interrupção, sobram perto de 6k posições de memória para o programa gráfico. A subrotina GRAF supõe que a memória disponível é de 5k; assim, se o número de palavras do programa gráfico for superior a 5120, este será dividido em um número par de blocos, sendo que, cada bloco consta de 2560 palavras no máximo.

Neste caso, o programa gráfico será gravado em disco e executado por partes. Para minimizar o tempo de acesso do disco, a subrotina GRAF calcula em que setor cada bloco deve ser gravado no disco, fornecendo ao monitor o número de setores adiantados que terá o próximo bloco, em relação ao último setor gravado do bloco anterior.

A DPU leva aproximadamente 23μs para traçar um vetor. O tempo que

a DPU levará para executar um bloco será:

$$TDPU = NPBx23$$ NPB-número de palavras por bloco

O disco leva 4μ para transmitir uma palavra; o tempo de transmissão de um bloco será:

$$TDISK = NPBx4+60$$

A inicialização dos registros do disco leva $60\mu s$.

Cada setor tem 64 palavras, um bloco ocupará S setores.

$$S = NPB/64$$

Após a leitura do 1º bloco o monitor carregará os registros do disco para a leitura do 2º bloco, inicializará a DPU e ficará à espera da interrupção da DPU na instrução de WAIT. A CPU leva aproximadamente $87\mu s$ para executar estas instruções; durante este tempo o disco irá girar 0,34 setores.

O número de setores adiantados do segundo bloco em relação ao primeiro será

$$NSA1 = TS+0,34$$

TS - fração de setor, indica a parte gravada do último setor do bloco.

Sendo o disco mais rápido que a DPU, o tempo de espera do disco será:

$$TE = TDPU-TDISK+30\mu S$$

A CPU levará $30\mu s$ para executar alguns testes antes de preparar o novo bloco.

Uma revolução completa do disco leva $17000\mu s$, portanto o número de revoluções do disco no tempo de espera será:

$$R = TE/17000$$

O disco possui 64 setores por "track", assim o número de setores adiantados será

$$NSA = FS + FR \times 64$$

FR - fração de revolução.

Como o monitor segue uma mesma sequência para os blocos restantes, o número de setores adiantados para todos os blocos, exceto o segundo, será constante.

Quando a memória é suficiente para conter o programa gráfico, o número de setores adiantados é feito igual a zero.

3.3.2.4 Transmissão dos dados

Estando o programa gráfico montado nas matrizes LX e LY e feito todos os cálculos, a subrotina GRAF inicia a transmissão dos dados enviando o comando "S" ao monitor, para este começar a recolher os dados.

Se a memória é suficiente para conter todo o programa gráfico, a subrotina GRAF envia os dois blocos de dados e o comando "G" para o monitor iniciar a execução, ficando a espera da mensagem do monitor que indicará se a transmissão foi satisfatória. Se a transmissão foi defeituosa, a subrotina volta a transmitir os dados, caso contrário, salta para o segmento que executa as mensagens do "light pen".

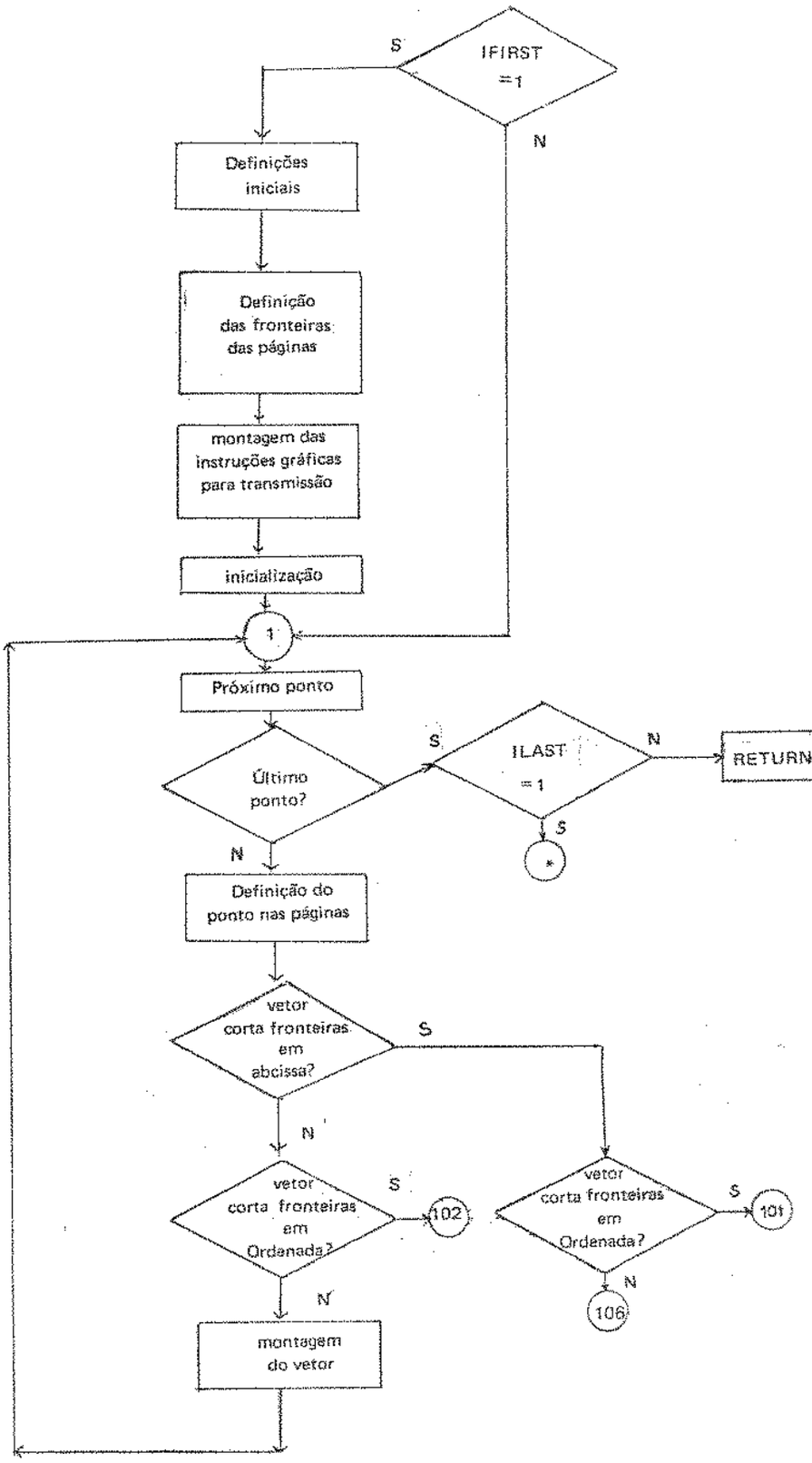
Se a memória não é suficiente, a subrotina envia a mensagem "D" após cada bloco de dados, verificando sempre se a transmissão foi satisfatória. A mensagem "D" fará o monitor saltar para o segmento que grava os dados em disco. Após o último bloco é enviada a mensagem "E" e os dados finais; esta mensagem fará o monitor executar o programa gráfico por partes.

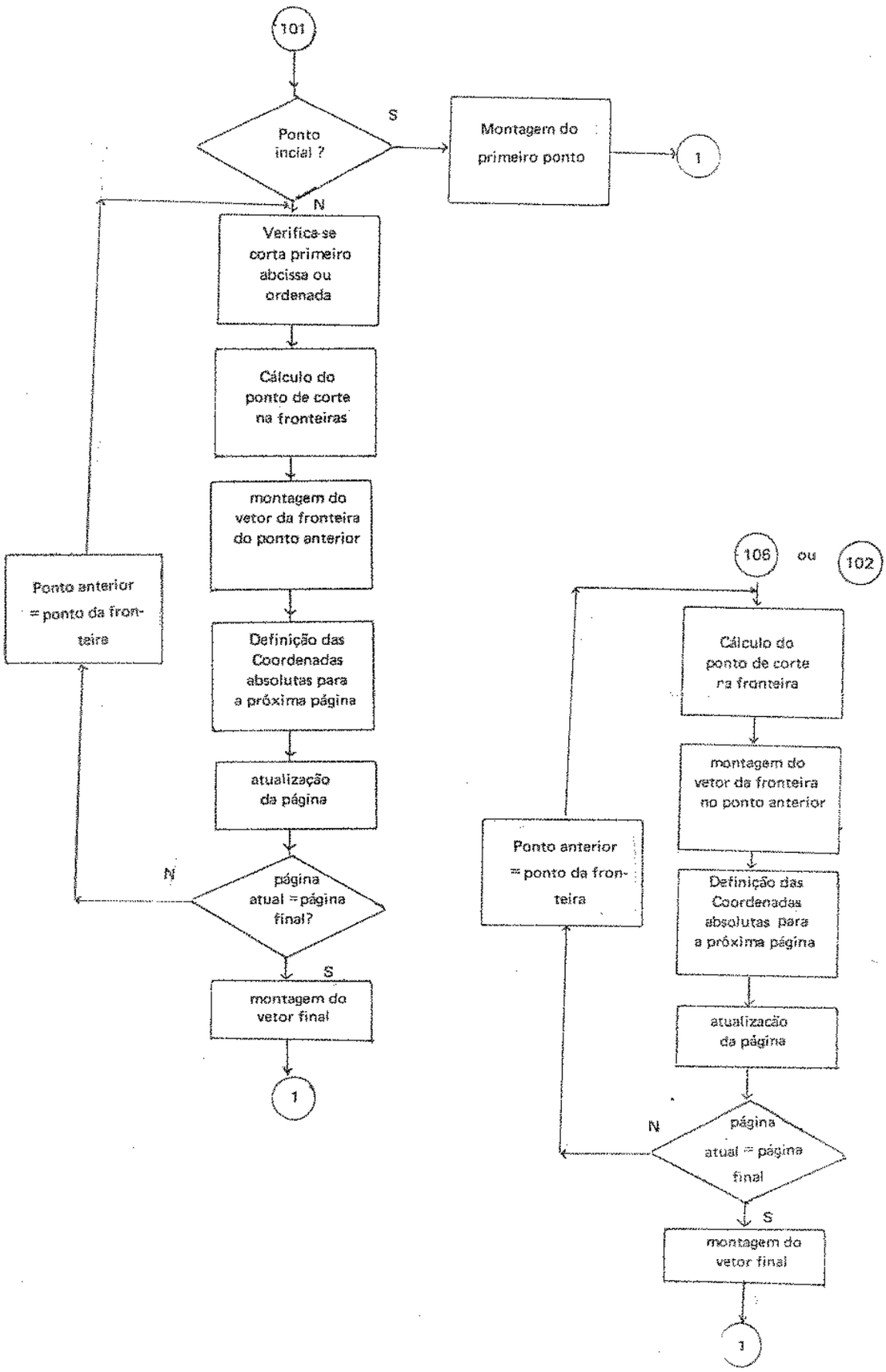
3.3.2.5 Interação pelo "light pen"

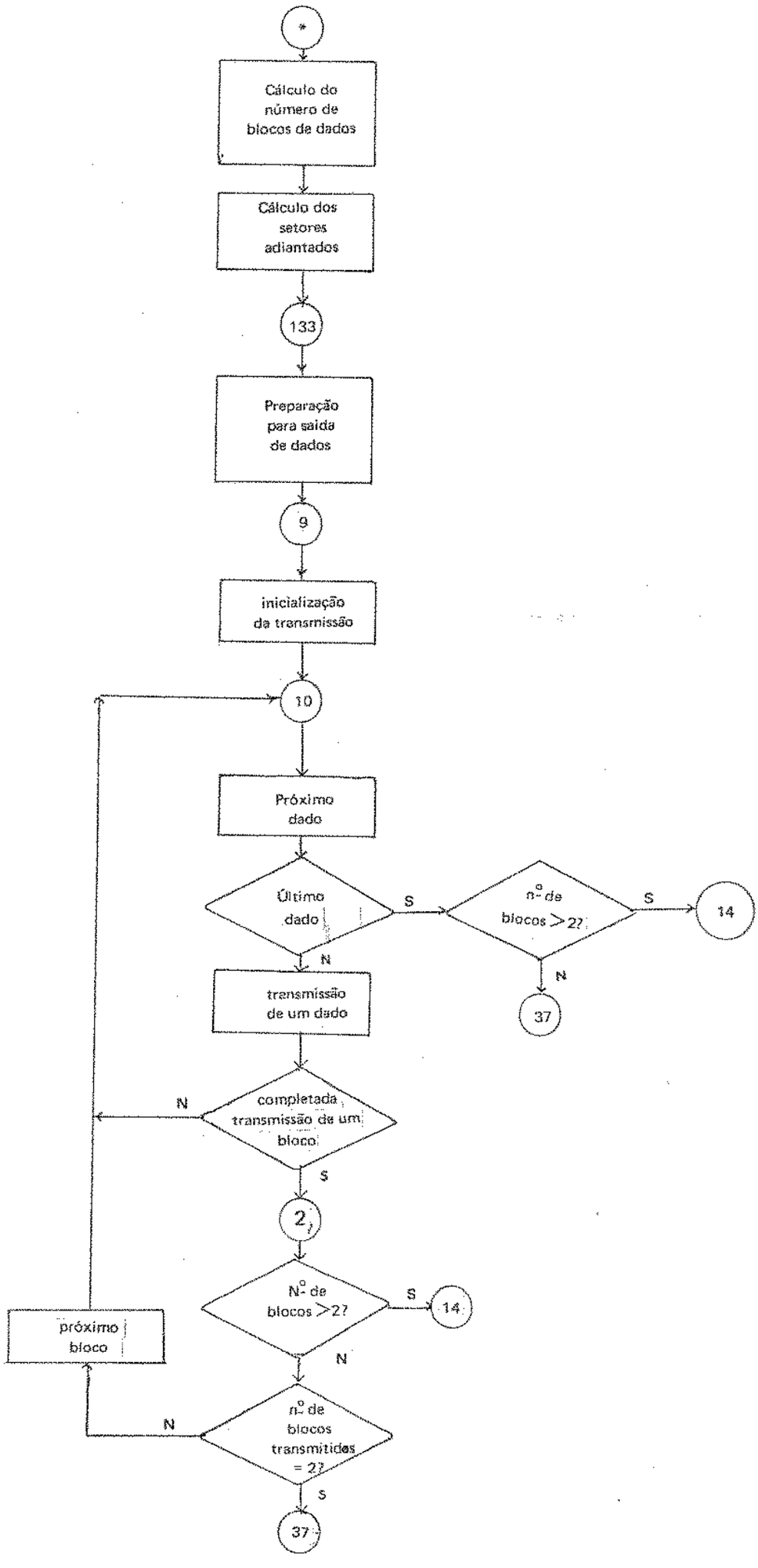
Após a transmissão de todos os dados, a subrotina GRAF salta para o segmento que controla as interações do usuário com o processo através do "light pen". Neste segmento a subrotina ficará a espera de uma mensagem do monitor; esta mensagem será a página escolhida para ampliação, na forma das duas variáveis I e J. A página escolhida será então ampliada e os dados enviados ao terminal seguidos da mensagem "G". Isto implica na hipótese de que a página ampliada não ultrapassa 5k de memória. Terminada a transmissão dos dados a subrotina volta ao início deste segmento ficando à espera de uma nova interação.

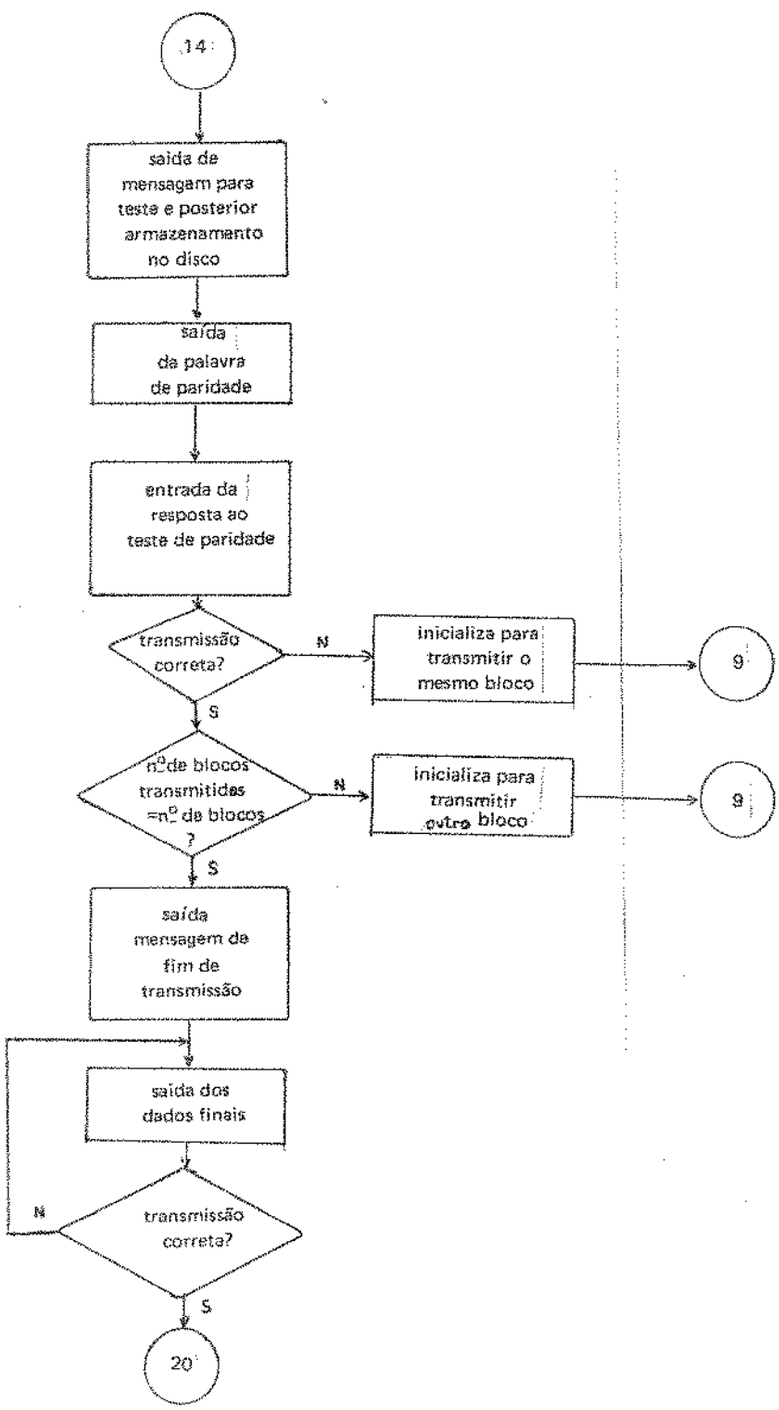
No caso de I ou J ser igual a zero, a subrotina fará a recomposição do gráfico original completo. Se a memória é suficiente para conter o programa gráfico a subrotina fará um salto para o segmento que transmite os dados, caso contrário, enviará a mensagem "M" que fará o monitor executar o programa gráfico (já gravado em disco) por partes.

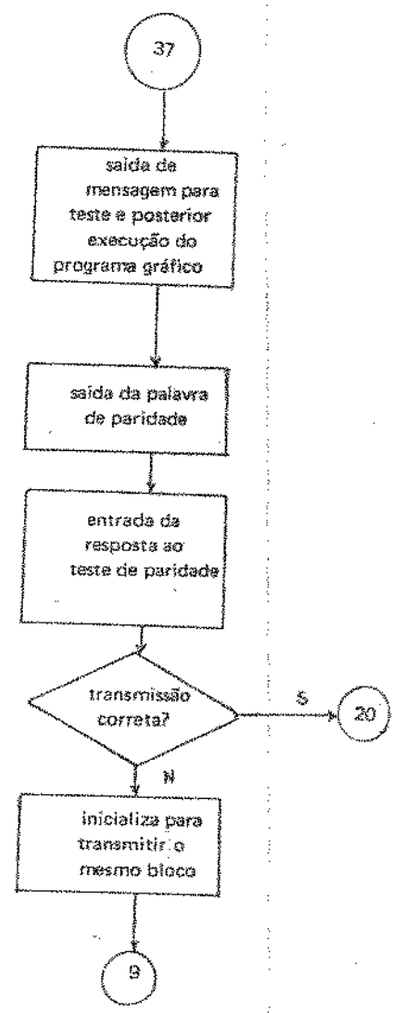
Para terminar a execução da subrotina, foi escolhido um valor para I e J que faz a subrotina ir para RETURN.

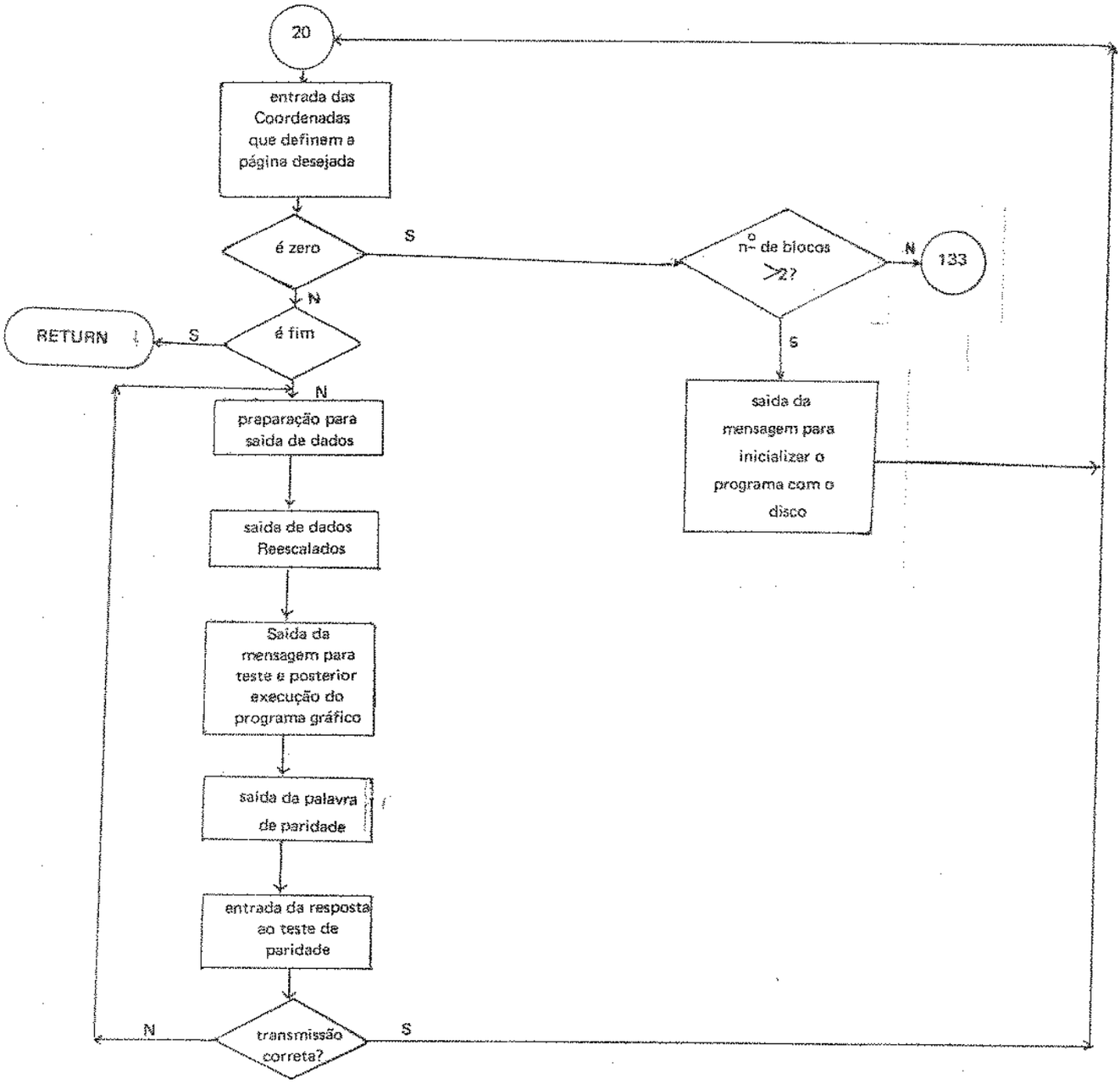

IFIRST
=1
S
N
Definições
iniciais
Definição
das fronteiras
das páginas
montagem das
instruções gráficas
para transmissão
inicialização
1
Próximo ponto
Último
ponto?
S
N
ILAST
=1
S
N
RETURN
*
Definição do
ponto nas páginas
vetor
corta fronteiras
em
abcissa?
S
N
vetor
corta fronteiras
em
Ordenada?
S
102
N
montagem
do vetor
vetor
corta fronteiras
em
Ordenada?
S
101
N
106


101
Ponto inicial ?
S
Montagem do primeiro ponto
1
N
Verifica-se corta primeiro abcissa ou ordenada
Cálculo do ponto de corte na fronteiras
montagem do vetor da fronteira do ponto anterior
Ponto anterior = ponto da fron-teira
Definição das Coordenadas absolutas para a próxima página
atualização da página
N
página atual = página final?
S
montagem do vetor final
1
108
ou
102
Cálculo do ponto de corte na fronteira
montagem do vetor da fronteira no ponto anterior
Ponto anterior = ponto da fron-teira
Definição das Coordenadas absolutas para a próxima página
atualizacão da página
N
página atual = página final
S
montagem do vetor final
1


*
Cálculo do número de blocos de dados
Cálculo dos setores adiantados
133
Preparação para saída de dados
9
inicialização da transmissão
10
Próximo dado
Último dado
S
nº de blocos > 2?
S
14
N
N
37
transmissão de um dado
completada transmissão de um bloco
N
S
2
N° de blocos > 2?
S
14
N
próximo bloco
nº de blocos transmitidos = 2?
N
S
37


14
saída de mensagem para teste e posterior armazenamento no disco
saída da palavra de paridade
entrada da resposta ao teste de paridade
transmissão correta?
N
inicializa para transmitir o mesmo bloco
9
S
nº de blocos transmitidos = nº de blocos ?
N
inicializa para transmitir outro bloco
9
S
saída mensagem de fim de transmissão
saída dos dados finais
transmissão correta?
N
S
20


37
saída de mensagem para teste e posterior execução do programa gráfico
saída da palavra de paridade
entrada da resposta ao teste de paridade
transmissão correta?
S
20
N
inicializa para transmitir o mesmo bloco
9


20
entrada das Coordenadas que definem a página desejada
é zero
S
N
nº de blocos >2?
N
133
S
RETURN
S
é fim
N
preparação para saida de dados
saida de dados Reescalados
Saida da mensagem para teste e posterior execução do programa gráfico
saida da palavra de paridade
entrada da resposta ao teste de paridade
saida da mensagem para inicializar o programa com o disco
N
transmissão correta?
S

CAPÍTULO IV - EXEMPLO DE APLICAÇÃO

Como exemplo de utilização foi escolhida uma superfície do tipo de Gauss com dois picos, definida por

$$y(x1,x2) = g_1(x1,x2) + g_2(x1,x2)$$

sendo

$$g_1(x1,x2) = EXP\left(-\frac{(x1-2)^2}{2} - \frac{(x2-2)^2}{3}\right)$$

$$g_2(x1,x2) = 5 \times EXP\left(-\frac{(x1-5)^2}{5} - \frac{(x2-5)^2}{5}\right)$$

Foram feitos 26 cortes fixando x1 em valores dentro do campo [0,5]. Cada curva resultante constará de 72 pontos variando x2 no campo [-5,13].

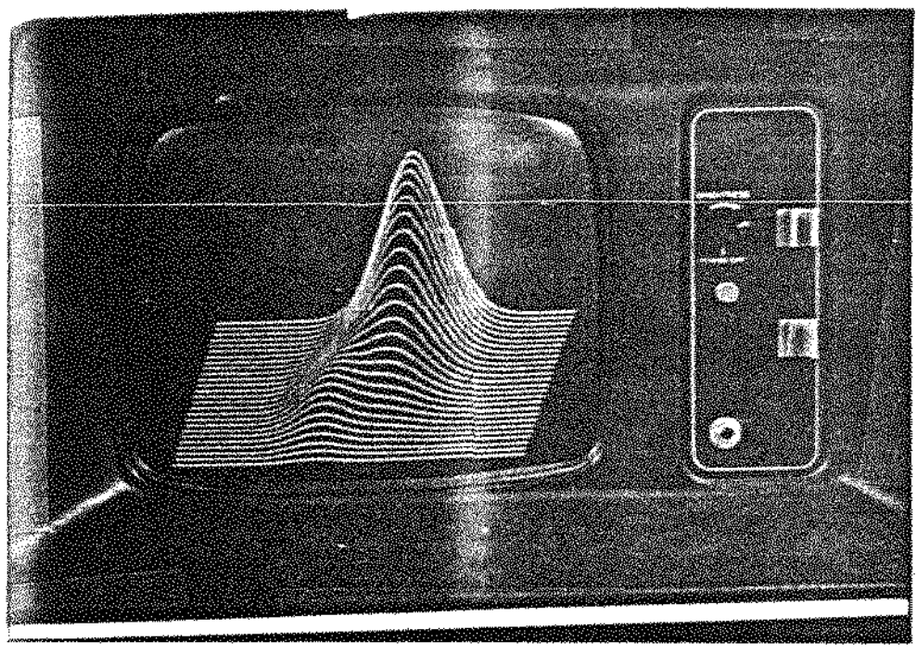
Para cada curva haverá uma chamada para a subrotina GRAF. (Ver Apêndice A).

Na primeira chamada o parâmetro IFIRST será igual a 1, indicando início do gráfico para a subrotina; o parâmetro ILAST será zero, pois não é a última curva. Os parâmetros restantes serão o número de pontos, suas coordenadas e as coordenadas máxima e mínima do gráfico completo. Com estes dados a subrotina fará os cálculos iniciais e montará o programa gráfico com a primeira curva.

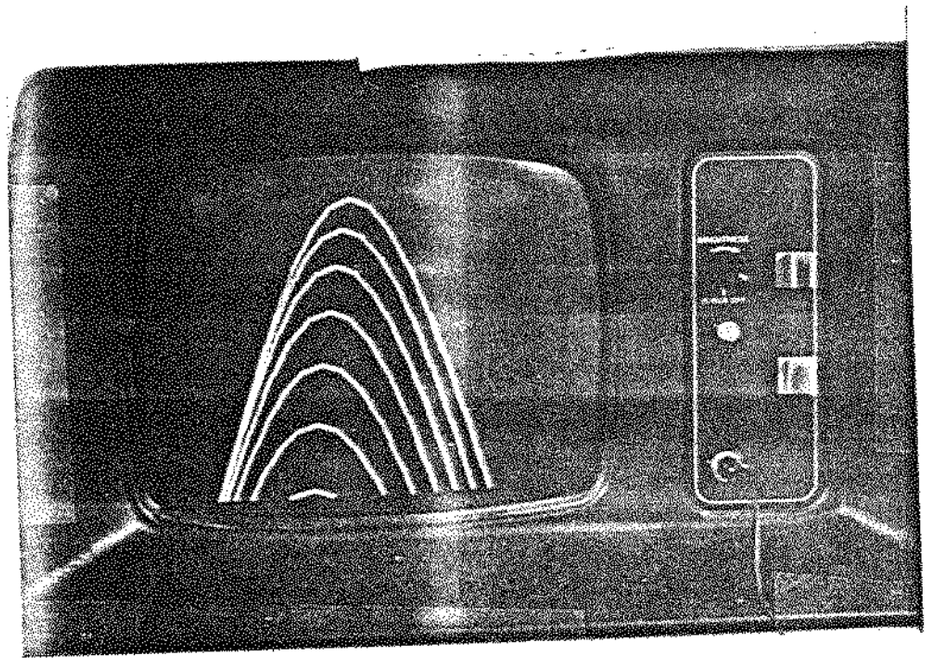
Para as chamadas intermediárias os parâmetros IFIRST e ILAST serão iguais a zero; as coordenadas máxima e mínima serão ignoradas pela subrotina pois os cálculos iniciais só serão executados na primeira chamada (IFIRST igual a um). A subrotina acrescentará cada curva ao programa gráfico.

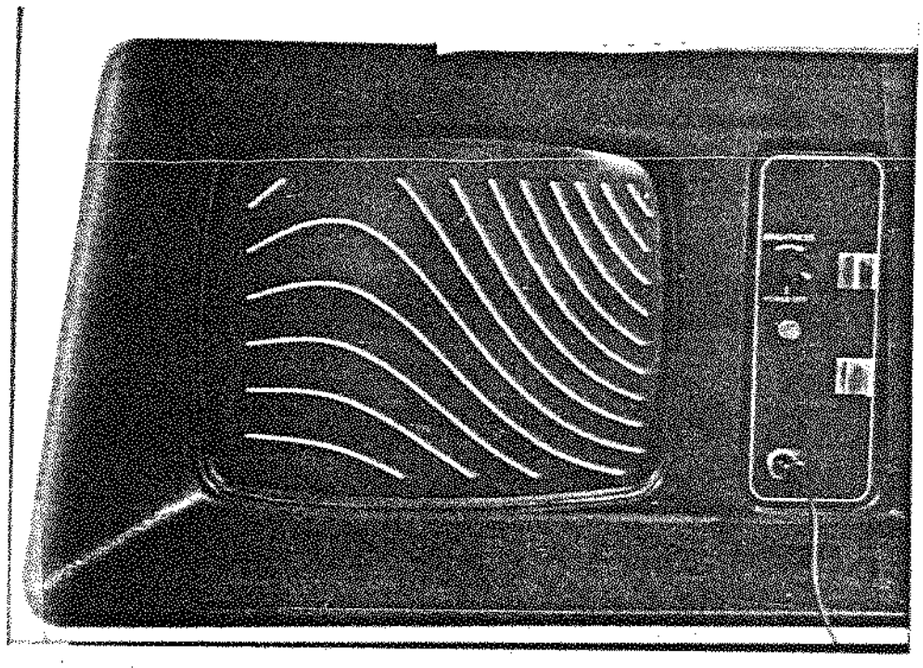
Quando ocorrer a última chamada (ILAST igual a um) a subrotina acrescentará a última curva e iniciará a transmissão do programa gráfico para o terminal. Terminada a transferência de dados,o contrôle do programa continuará com a subrotina, que ficará a espera de uma interação do usuário.

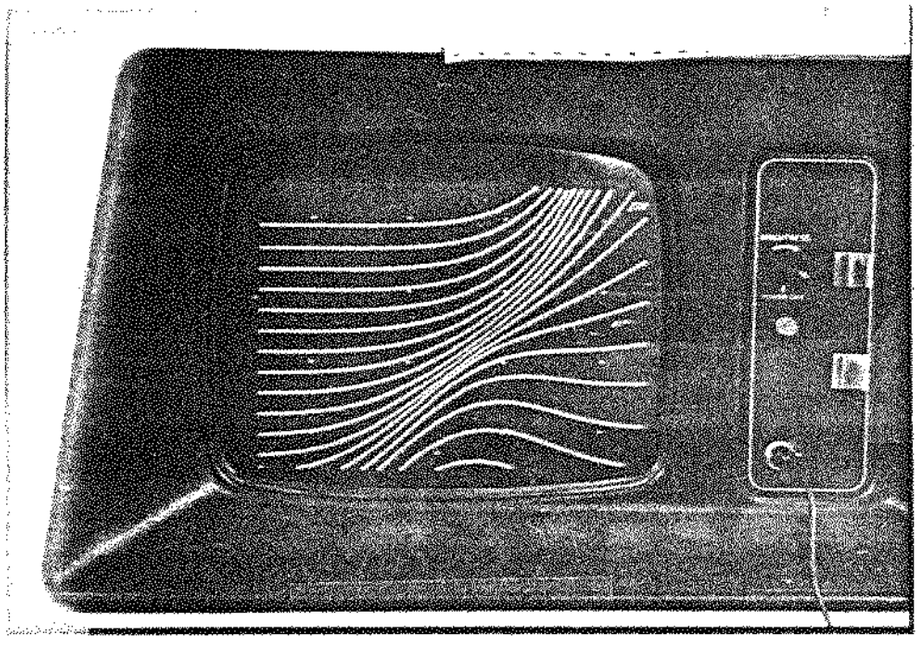
Após receber a mensagem de fim de transmissão, o monitor residente no terminal colocará no programa gráfico os pontos centrais das regiões em que está dividido o gráfico, e inicializará a DPU, aparecendo então a imagem na tela.



Tocando um dos pontos centrais com o "light pen", a região desejada aparecerá ampliada na tela.







O retorno à imagem inicial se fará pelo ponto inferior à esquerda.

O ponto superior à direita fará a subrotina GRAF ir para RETURN, terminando o processamento deste gráfico. Estando o controle novamente com o programa de aplicação do usuário, este poderá processar outros gráficos se necessário.

CONCLUSÃO

O algorítmo proposto, para resolver o problema encontrado no tratamento da informação gráfica, foi implantado e testado.

A tela foi dividida em 16 regiões e as ampliações conseguidas consideradas satisfatórias para os gráficos testados.

Um gráfico de 3000 pontos apareceu nítido na tela, apresentando apenas um ligeiro cintilamento. Para um gráfico com maior número de pontos cujo cintilamento seja muito intenso, será necessário fotografar a tela para a obtenção da imagem.

Os gráficos de muitos pontos que são executados por partes não foram testados, uma vez que não se encontra instalada a memória de massa referida no programa.

A estrutura do programa permite desenvolvimentos futuros que poderão ser introduzidos à medida que houver necessidade. Entretanto deve-se lembrar do compromisso de espaço de memória reservado ao gráfico; o melhoramento não deverá implicar em um número grande de instruções acrescentadas ao programa assembler residente no GT40, de modo a serem mantidas as 5k posições de memória para o programa gráfico.

Fazendo uso dos mesmos algorítmos usados para paginação e montagem do programa gráfico, a subrotina GRAF pode ser escrita em uma linguagem de alto nível que permite uma estrutura de dados mais flexível. Seria desejável que a linguagem permitisse recursividade, havendo assim possibilidade de ampliações consecutivas até o nível desejado pelo usuário.

BIBLIOGRAFIA

J.L. Encarnação **Computer Graphics: Programmierung und Anwendung von Graphischen Systemem**, R. Oldenburg Verlag 1975.

H.S. Stone, D.P. Siewiorek **Introduction to Computer Organization and Data Structures: PDP-11 Edition, McGraw-Hill 1975.**

PDP-11 Peripherals Handbook, **Digital.**

Progressor Handbook PDP-11/05, **Digital.**

GT-40 User's Guide **DEC-11-HGTGA–A–D, Digital.**

BATCH-11/DOS-11 - Assembler (MACRO-11) **DEC-11-OMACA–B–D**, Digital

GT-40 Graphic Display Terminal, Vol. 1 e 2, DEC-11-HGTVA–B–D e DEC-11-HGTMA–A–D, Digital

Picture Book Reference Manual, DEC-11-GPBMA–A–D, Digital.

J. Potts - **Computer Graphics - Whence and Hence** Computer & Graphics, Vol. 1 nº2/3, Setembro 1975.

T.J. Wright - **Practical Computer Graphics for Scientific Users, Philosophy and Implementation**, Computer & Graphics, Vol. 1 nº 2/3, Setembro 1975.

F.R. Stocker, T. Minsker - **Graphics Geometric Perception and Communication**, Computer & Graphics, Vol. 1 nº 2/3, Setembro 1975.

T.A. DeFanti, D.J. Sendin, T.H. Nelson - **Computer Graphics as a Way of Life** - Computer & Graphics, Vol. 1 nº 1, Maio 1975.

L.T. Speecher - **Interative Computer Graphics for Assisting Human Programmers**, Computer & Graphics, Vol. 1 nº 1, Maio 1975.

C.N. Turrill, W.R. Mallgren - XPLG - **Experiences in Implementing an Experimental Interative Graphics Programming System**, Computer & Graphics, Vol. 1 nº 1, Maio 1975.

APÊNDICE A

```
      DIMENSION X11(26),X22(72),X(3000),Y(3000)
C
C     CALCULOS INICIAIS
C
      X11(1)=0.
      X22(1)=-5.
      DO 1 I=2,26
1     X11(I)=X11(I-1)+.2
      DO 2 I=2,72
2     X22(I)=X22(I-1)+.25
      SEN=SIN(3.1415*30./180.)
      COS=COS(3.1415*30./180.)
      XMIN=-5.
      YMIN=0.
      XMAX=13.+X11(26)*SEN
      YMAX=5.+X11(26)*COS
      X1=X11(1)
      DO 3 I=1,72
      X(I)=X22(I)
      X2=X22(I)
3     Y(I)=G1(X1,X2)+G2(X1,X2)
C
C     CHAMADA PARA MONTAR A PRIMEIRA CURVA
C
      CALL GRAF(1,0.,72,X,Y,XMIN,YMIN,XMAX,YMAX)
      DO 4 J=2,25
      X1=X11(J)
      DO 5 I=1,72
      X(I)=X22(I)+X11(J)*SEN
      X2=X22(I)
5     Y(I)=(G1(X1,X2)+G2(X1,X2))+X11(J)*COS
C
C     CHAMADA PARA MONTAR AS CURVAS INTERMEDIARIAS
C
      CALL GRAF(0.,0.,72,X,Y,XMIN,YMIN,XMAX,YMAX)
4     CONTINUE
      X1=X11(26)
      DO 6 I=1,72
      X(I)=X22(I)+X11(26)*SEN
      X2=X22(I)
6     Y(I)=(G1(X1,X2)+G2(X1,X2))+X11(26)*COS
C
C     CHAMADA PARA MONTAR A ULTIMA CURVA
C
      CALL GRAF(0.,1.,72,X,Y,XMIN,YMIN,XMAX,YMAX)
      STOP
      END
      FUNCTION G1(X1,X2)
      G1=EXP(-(X1-2.)**2/20.-(X2-2.)**2/3.)
      RETURN
      END
      FUNCTION G2(X1,X2)
      G2=5.*EXP(-(X1-5.)**2/3.-(X2-5.)**2/5.)
      RETURN
      END
```