

---

**PROJECTO DE INVESTIGAÇÃO  
EM COMPUTAÇÃO GRÁFICA**  
(INTEGRADO NO PLANO DE COLABORAÇÃO IST-UNL)

---

JULHO A DEZEMBRO 74

II. TOMO

**PROJECTO DE INVESTIGAÇÃO  
EM COMPUTAÇÃO GRÁFICA**  
(INTEGRADO NO PLANO DE COLABORAÇÃO IST-UNL)

Relatório Semestral  
(Julho a Dezembro de 1974)

Supervisor: Madalena Quirino  
Responsável: J. A. Legatheaux Martins

## ÍNDICE

1. INTRODUÇÃO
2. DESCRIÇÃO FUNCIONAL DE IBID
3. GRAVAÇÃO DE BONDAS EXPERIMENTAIS
4. SOLUÇÃO FINAL EM FORTRAN PARA DESCARGA DO DISPLAY-FILE NA BANDA
5. SOLUÇÃO FINAL DE PGBD
6. CONCLUSÕES

## 1. INTRODUÇÃO

No relatório anterior, referente à actividade desenvolvida durante o período de Abril a Junho de 1974, foram deixadas em aberto algumas questões que se podem resumir em:

- a) A utilização ou não da densidade de gravação de 800 BPI;
- b) A utilização ou não de códigos especiais para indicação do fim e do início do bloco de informação;
- c) A utilização do FORTRAN como linguagem para escrita da informação na banda;

A estas questões podem ainda ser acrescentadas as seguintes:

- d) A utilização ou não do disco como suporte do DISPLAY-FILE;
- e) A solução final para o problema do refrescamento.

Este conjunto de questões exigiam uma maior colaboração Hard/Soft no que toca à discussão das soluções do hardware para que a resposta final pudesse entrar em consideração com os 2 pontos de vista.

Os resultados dessa colaboração são descritos nos capítulos 2 e 3 do presente relatório e permitiram resolver algumas questões que estavam em aberto, nomeadamente a) e b).

No que se refere à questão colocada em c) resolveu-se explorar até ao fim as potencialidades da escrita da banda em FORTRAN. Os resultados e as conclusões desse trabalho são descritas no capítulo 4.

Finalmente, a experiência e as conclusões que se tiraram da utilização do FORTRAN como linguagem para escrever na banda, assim como a crítica a que foi sujeito PGBD.\* Estes resultados são descritos no capítulo 5 deste relatório.

No capítulo 6 são sistematizadas todas as questões resolvidas, as conclusões e ainda as questões que não estão totalmente resolvidas. A mais importante destas é o problema do refrescamento, que só poderá ser resolvido pela experimentação prática ainda impossível neste momento. No entanto a solução obtida parece-nos bastante satisfatória e não se prevê que a experiência vá conduzir a alterações de fundo.

O presente relatório será seguido de um relatório final em que somente os resultados obtidos serão descritos e no qual será incluído um manual para o utilizador.

---

\* Pelos colaboradores da U.N.L., permitiu corrigir os erros e introduzir novos melhoramentos que conduziram à sua forma final.

## 2. DESCRIÇÃO FUNCIONAL DE IBID

A colaboração da U.N.L. na montagem e discussão do hardware do IBID, permitiu definir melhor uma evolução para o projecto software.

Foi também reforçada a colaboração do NEEM na discussão da solução software mais adequada para o problema do refrescamento e tempo de exposição.

Com vista a permitir ao leitor compreender perfeitamente as soluções adoptadas para os diversos problemas de software é em seguida feito um resumo descritivo de IBID.

A informação encontra-se registada na banda na forma de bytes, como se apresenta na figura. A banda é percorrida horizontalmente por 9 pistas, das quais a nona contém o bit de paridade (a paridade utilizada é ímpar).

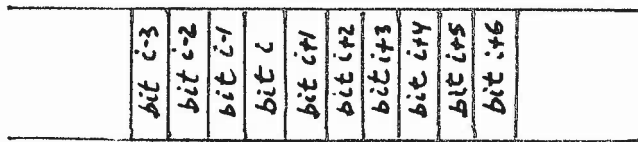


Fig. 1 - Informação na banda

A informação tem como suporte físico a variação de fluxos magnéticos. As unidades de bandas magnéticas da IBM comportam 2 sistemas básicos de registo de informação. O primeiro destes sistemas é o Phase encoding (PE) que usa uma densidade de gravação de 1600 BPI (100 bytes per inch).

Em PE um 0 numa pista é assinalado pela transição do fluxo magnético de tal forma que a sua derivada seja negativa e 1 é assinalado de tal forma que a derivada da transição seja positiva. O sistema dispõe de um relógio de tal forma que só são consideradas como informação as variações presentes em determinados intervalos de tempo precisos.

Exemplo:

A sequência 001010011 tem a seguinte forma:

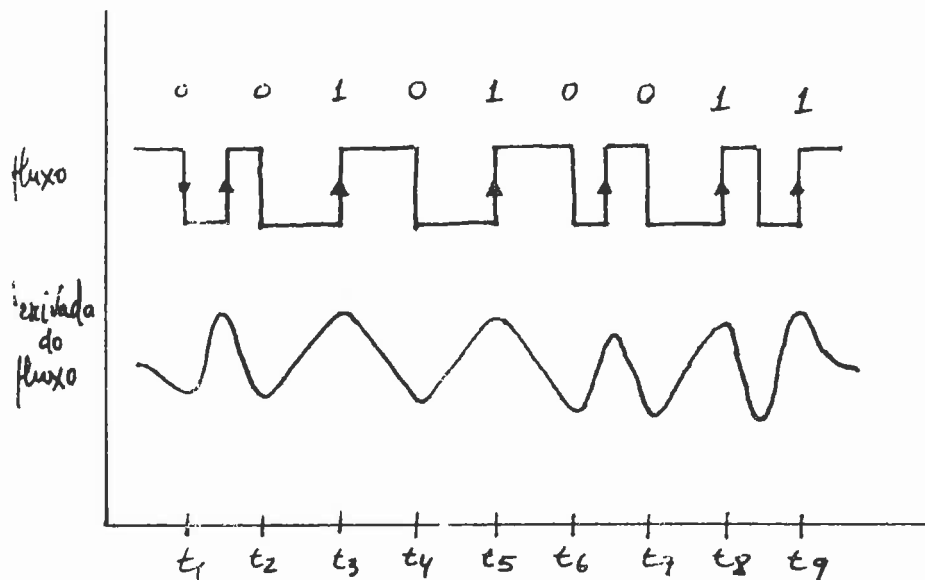


Fig. 2 - Exemplo de escrita em PE

Quando a banda se desloca diante da cabeça magnética a tensão induzida nos terminais das bobinas desta tem a forma da derivada do fluxo.

O sistema de inscrição PE torna-se inadequado à utilização pela IBID, principalmente pelas seguintes razões:

- 1 - É necessário sincronizar perfeitamente o relógio usado na inscrição com o relógio usado na leitura.
- 2 - É necessário sincronizar perfeitamente o início da escrita com o início da leitura.
- 3 - Qualquer variação de velocidade que provoque um atraso ou avanço da banda magnética da ordem de  $(t_i - t_{i-1})/2$  provoca erros na leitura. Repare-se que estas variações são da ordem de  $1/(1600 \times 2)$  polegada.

Por estas razões e visto que a unidade de banda magnética de que se dispunha para ligar a IBID era mecânica muito deficiente, este modo de gravação foi posto de parte.

O outro sistema de gravação é o designado por Non Return to Zero Inverter (NRZI). Neste sistema o 1 é assinalado por uma variação de sentido arbitrário do fluxo magnético e o 0 pela ausência de variação. Evidentemente que se pressupõe que existe sincronização da escrita e da leitura, o que é conseguido com a utilização de 2 relógios de frequências iguais.

Tomemos o exemplo anterior.

A seqüência 00 10011 aparece em NRZI com o seguinte aspecto

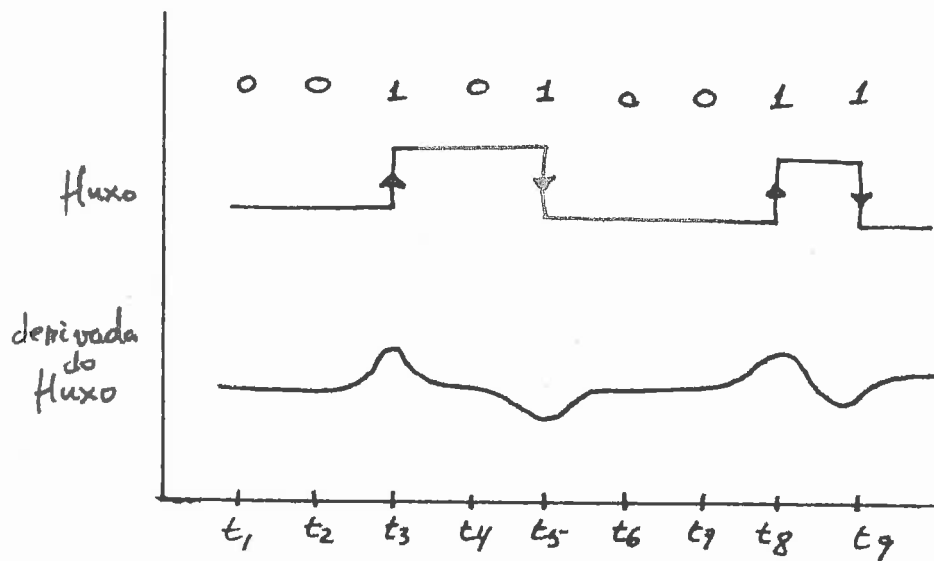


Fig. 3 - Exemplo de escrita em NRZI

A observação da figura mostra que é necessário dispôr-se de um relógio para saber se no intervalo de dois 1'S existe ou não um zero, por esta razão o trabalho com o código NRZI vai exigir que a banda magnética corra com a velocidade absolutamente constante para que a sincronização se mantenha.

No nosso caso a unidade de leitura não garantia esta sincronização. Este problema foi então resolvido pela equipe de hardware da seguinte forma:

Dado que a paridade utilizada é ímpar, em cada byte existe pelo menos um dos bits cujo valor é 1. Se se fizer um "OU" de todos os bits obtêm-se assim um sinal que apresenta uma sucessão de 1'S sincronizados com frequência de escrita.

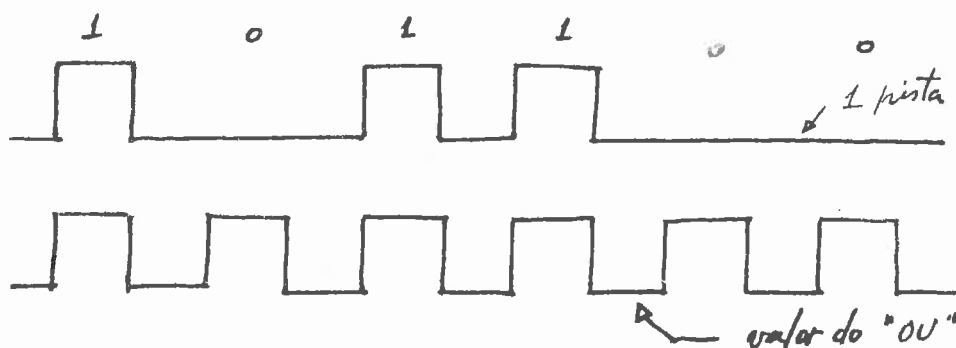


Fig. 4 - Obtenção do relógio

Os bytes são escritos na banda em blocos de informação separados por espaços de fita sem qualquer informação chamados "gaps". A organização da informação na banda é:

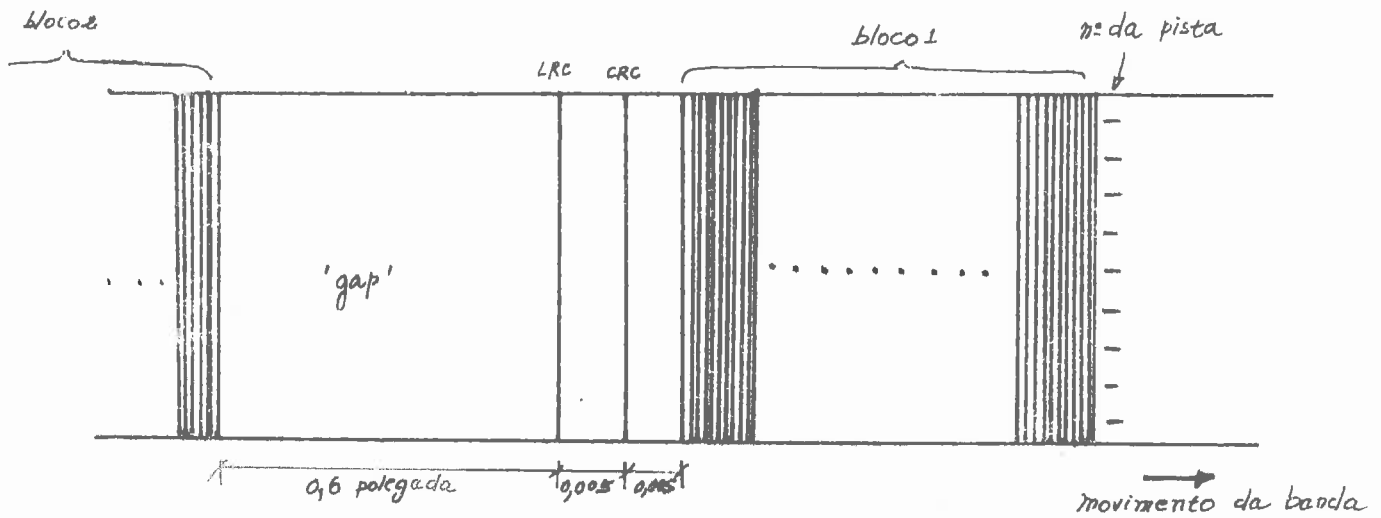


Fig. 5 - Organização da informação na banda

A gravação é feita na frequência de 30.000 bytes por segundo. Os bytes LRC e CRC são bytes de teste da informação do bloco. O tamanho do gap é fixo e é o indicado na figura anterior.



A organização de IBID é a seguinte:

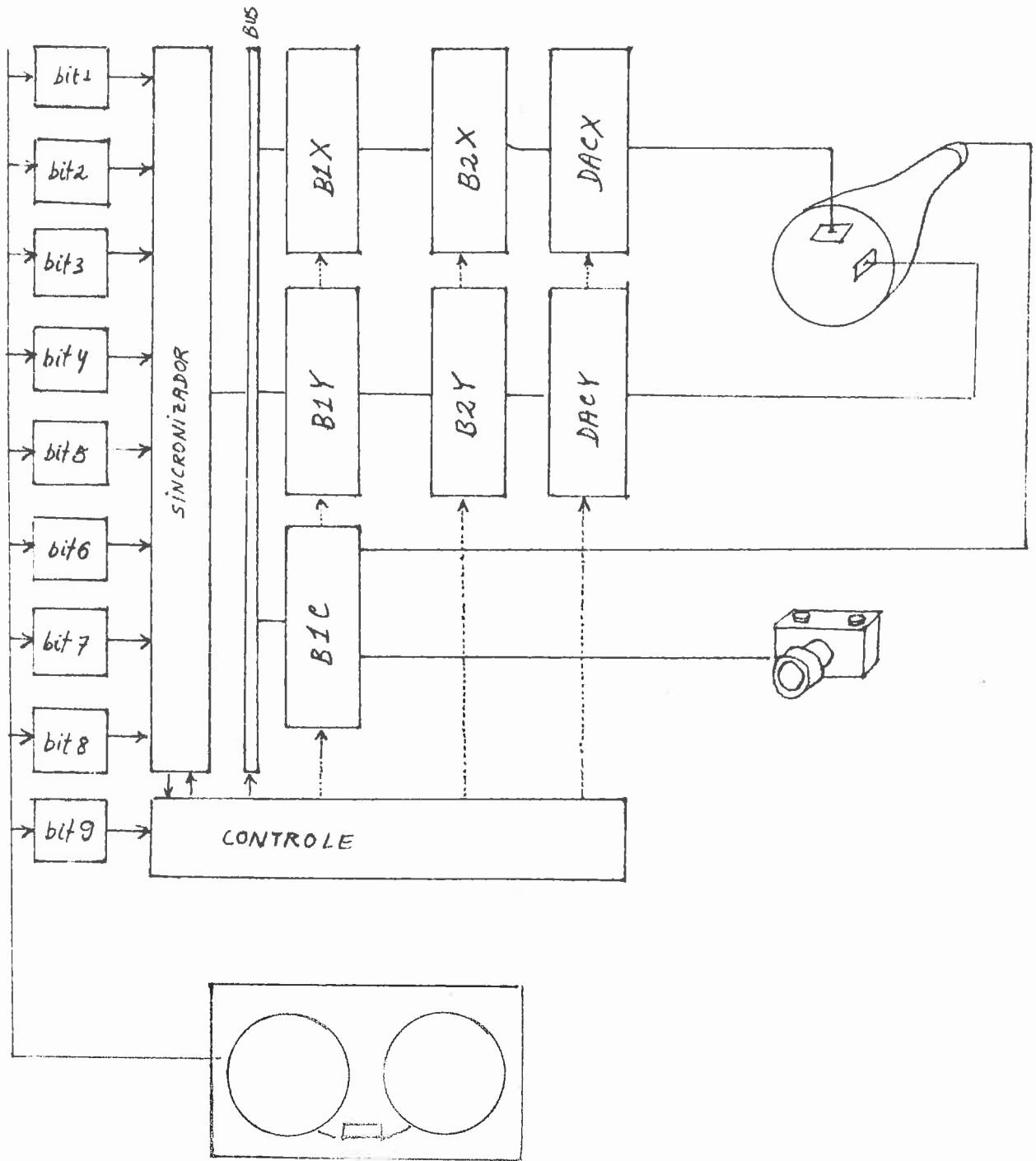


Fig. 6 - Organização em blocos de IBID

As unidades assinaladas com a indicação de bit têm como função a digitalização da informação. A unidade designada por "SINCRONIZADOR" tem como função a sincronização dos diferentes bits de forma a obter-se à saída o byte com todos os bits em simultâneo e sem ruído pelo meio.

Através do BUS a informação vai sendo transferida para os buffers da seguinte forma:

Primeiro o byte X é acumulado em B1X; em seguida o byte Y é acumulado em B1Y mantendo-se o X em B1X. Finalmente o byte C é acumulado em BC e nessa altura a informação dos buffers B1X e B1Y é transferida respectivamente para B2X e B2Y.

Através dos DACX, DACY (Digital to analog couvaters) e BC a informação da instrução é passada ao tubo de raios catódicos ficando B1X e B1Y prontos a receber a nova instrução e assim sucessivamente.

tempo 1	tempo 2	tempo 3
byte X entra em B1X	byte Y entra em B1Y	byte C entra em BC; B2X recebe o valor de B1X; B2Y recebe o valor de B1Y; a infor mação é passada ao tu bo através das DAC'S A instrução fica a ser executada e vol- ta-se ao tempo 1.
Feixe está a transitar para o ponto cujas coordenadas estão em (B2X e B2Y).		

Fig. 7 - Funcionamento de IBID.

Outras características importantes de IBID são:

Deteção automática do início e fim do bloco de informação; e durante esse período entra informação nos buffers.

Cada duas instruções, ou seja cada dois pontos a representar estão separados no tempo pelo período necessário ao aparecimento de 3 bytes, isto é  $3 \times \frac{1}{30.000} \text{ s} = 0.1 \text{ ms}$ , sendo assim a frequência máxima de representação 10.000 pontos por segundo.

Uma instrução provoca a transição do feixe do ponto em que este se encontra para o novo ponto. Este percurso pode ser maior ou menor, no entanto o tempo de transição é sempre inferior a 0.1 ms.

### 3. GRAVAÇÃO DE BANDAS EXPERIMENTAIS

Ao longo da construção e ensaios de IBID foi necessário dispor de bandas contendo informação adequada, com vista a ensaiar as diferentes partes da interface.

Essas bandas foram gravadas com auxílio de um programa em FORTRAN e contêm blocos de 252 bytes idênticos.

Foram construídas 4 bandas:

- 1 - 'Banda a 1'S U.N.L.' contendo 20.038 blocos de 252 bytes em que todos os bits têm o valor 1.
- 2 - 'Banda triângulo + 1' contendo 20,018 blocos de 252 bytes iguais nove a nove e contendo a informação necessária para representar um triângulo de coordenadas  
(52,21) , (81,51) e (21,101)
- 3 - 'Banda circunferência' contendo 8461 blocos de 252 bytes cada um dos quais constituído por 84 pontos que representam uma circunferência.
- 4 - 'Banda estrela' contendo 8734 blocos de 252 bytes sendo cada um destes blocos constituído por 9 conjuntos de 27 bytes representando uma figura poligonal complexa com 9 pontos.

O programa para obter estas bandas é o mesmo, variando apenas de caso para caso a parte que se refere à construção do bloco.

A estrutura de dados utilizada pelo programa é a seguinte:

nome	tipo	função	estrutura
BUF1 (252)	integer * 2	conter o bloco em inteiros	vector
BUF2 (504)	logical * 1	construção do bloco em bytes	vector
BUF3 (252)	logical * 1	conter o bloco a colocar na banda	vector

Fig. 8 - Estrutura de dados

Os diferentes vectores BUF ocupam todos a mesma zona de memória. Encontram-se ligados através das instruções EQUIVALENCE (BUF1(1), BUF2(1)) e (BUF2(1), BUF3(1)).

As técnicas para se conseguir construir o bloco são as mesmas usadas na SUBROUTINE DISP já referida no relatório trimestral anterior.

```
INTEGER * 2 BUF1 (252)
LOGICAL * 1 BUF2 (504), BUF3 (252)
EQUIVALENCE (BUF1(1), BUF2(1))
EQUIVALENCE (BUF2(1), BUF3(1))
REWIND *

------(construção do bloco)-----

J = 2
DO 20 I = 1,252
  BUF2 (I) = BUF2 (J)
20 J = J + 2
DO 30 (I) = 1,20000
  WRITE (4, 4, END = 40) BUF3
30 CONTINUE
4  FORMAT (252 A1)
40 C* L IMPRIM
END
```

Texto do programa de construção de bandas especiais

Apresentam-se em seguida os diferentes blocos correspondentes a cada imagem.

1 - 'Banda 1 U.N.L.'

```
DO 10 I=1,252
  BUF1 (I)=255
10 CONTINUE
```

2 - 'Banda circunfrência'

```
ALFA=0
DO 10 I=1,252,3
  BUF1 (I)=COS (ALFA) * 100.+100
  BUF1 (I+1)=SIN (ALFA) + 100.+100
  BUF1 (I+2)=0
  ALFA=ALFA + 0.074
10 CONTINUE
```

3 - 'Banda triângulo+1'

```
N0=1
N20=21
N50=51
N80=81
N100=101
DO 10 I=1,252,9
  BUF1 (I)=N50
  BUF1 (I+1)=N20
  BUF1 (I+2)=N0
  BUF1 (I+3)=N80
  BUF1 (I+4)=N50
  BUF1 (I+5)=N0
  BUF1 (I+6)=N20
  BUF1 (I+7)=N100
  BUF1 (I+8)=N0
10 CONTINUE
```

4 - 'Banda estrela'

```
N20=20
N220=220
N0=0
N170=170
N70=70
N120=120
DO 10 I=1,243,27
  BUF1 (I)=N20
  BUF1 (I+1)=N20
  BUF1 (I+2)=N0
  BUF1 (I+3)=N220
  BUF1 (I+4)=N20
  BUF1 (I+5)=N0
  BUF1 (I+6)=N170
  BUF1 (I+7)=N70
  BUF1 (I+8)=N0
  BUF1 (I+9)=N220
  BUF1 (I+10)=N70
  BUF1 (I+11)=N0
  BUF1 (I+12)=N220
  BUF1 (I+13)=N120
  BUF1 (I+14)=N0
  BUF1 (I+15)=N120
  BUF1 (I+16)=N120
  BUF1 (I+17)=N0
  BUF1 (I+18)=N120
  BUF1 (I+19)=N170
  BUF1 (I+20)=N0
  BUF1 (I+21)=N20
  BUF1 (I+22)=N120
  BUF1 (I+23)=N0
  BUF1 (I+24)=N70
  BUF1 (I+25)=N70
  BUF1 (I+26)=N0
10 CONTINUE
  DO 15 I=244,252
    BUF1 (I)=N0
15 CONTINUE
```

O conteúdo das bandas foi testado por uma subrotina especial que lê o número especificado de blocos da banda e escreve o seu conteúdo na impressora.

Esta subrotina tem o nome IMPRIM.

```
SUBROUTINE IMPRIM
INTEGER * 2 BUF1 (252)
LOGICAL * 1 BUF2 (504), BUF3 (252), BUF4 (50)
EQUIVALENCE (BUF1(1), BUF2(1))
DO 5 I=1,252
  BUF1 (I)= 0
5 CONTINUE
10 READ (1,1000) N, BUF4
1000 FORMAT (I2,50A1)
  WRITE (3,1010) BUF4
1010 FORMAT (1H1, 50A1//)
  REWIND 4
  DO 30 I=1,N
    READ (4,1020) BUF3
1020 FORMAT (252 A1)
    J=504
    K=252
    20 BUF2 (J) BUF3 (K)
      J=J-2
      K=K-1
      IF (K.GT.0) GOTO 20
    WRITE (3,1030) BUF1
1030 FORMAT (1H1, I2 (21 I5//))
    30 CONTINUE
      REWIND 4
      RETURN
      END
```



A estrutura de dados desta subrotina é a seguinte:

NOME	TIPO	DIMENSÃO	FUNÇÃO	ESTRUTURA
BUF1	Integer * 2	252	Conter o bloco a pôr na impressora	vector
BUF2	Logical * 1	504	Tratamentos intermédios	vector
BUF3	Logical * 1	252	Conter o bloco da banda	vector
BUF4	Logical * 1	50	Conter o array identificativo da banda	vector

Nota: BUF1, BUF2 e BUF3 coexistem na mesma área de memória.

Fig. 9 - Estrutura de dados da subrotina IMPRIM.

#### 4. SOLUÇÃO FINAL EM FORTRAN PARA DESCARGA DO DISPLAY FILE NA BANDA

A solução que em seguida é descrita aproveita ao máximo o tamanho de bloco FORTRAN. Assim, a zona da banda onde é escrita informação é a seguinte:

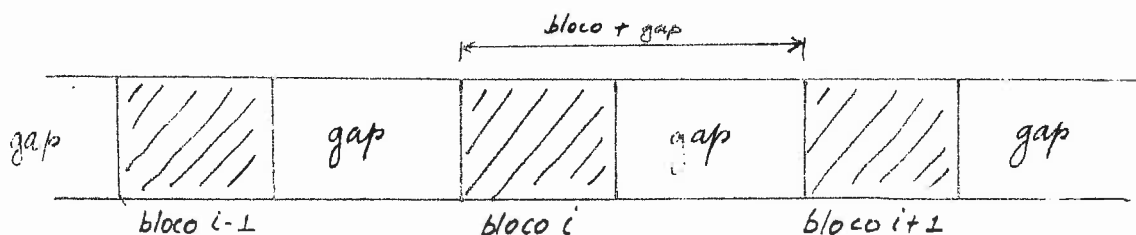


Fig. 10 - Informação na banda em FORTRAN.

O tamanho de cada bloco é constante e igual a 252 bytes e o tamanho do gap é equivalente a 548 bytes, visto que estamos a trabalhar com a densidade de gravação de 800 BPI.

A cada conjunto "bloco + gap" corresponde um tempo de representação de  $800/30\ 000$  s ou seja 0,0266 s, visto que a velocidade de gravação é de 30 K bytes/s.

A escrita é feita com o formato 252A1 visto que se os blocos fossem escritos por instruções não formatadas ocupariam mais 8 bytes com informação extra que provocaria interferências na imagem.

As características de PGBD especificavam que cada imagem poderia ter um mínimo de 1 ponto e um máximo de 1.000 pontos. A informação correspondente a um único percurso do feixe sobre uma imagem pode ocupar de 3 a 3.000 bytes ou seja de 1/84 do bloco até 13 blocos o que corresponde a um máximo de tempo, para um refrescamento, igual a 0,35 s.

Chega-se assim à conclusão de que a saída de imagens com 1000 pontos com a DISP em FORTRAN é praticamente inviável por permitir um máximo de 3 refrescamentos por segundo o que é manifestamente insuficiente para um tubo de raios catódicos normal.

Dado um determinado tempo de permanência da imagem no écran o número de blocos que na banda conterão pontos dessa imagem é dado por:

$$NBLO = \left\lceil (T/2.66) \right\rceil + 1$$

O conjunto de NBLO blocos é preenchido com os pontos da imagem sequencialmente como se não existissem gaps como a seguir se especifica:

Seja NBLO o número de blocos que ocupa uma imagem e n o número de pontos da imagem.

Seja ainda por hipótese

$$NBLO \gg \frac{n}{84}$$

Consideremos agora os seguintes casos:

1º caso:  $n < 84$

2º caso:  $n = 84$

$$\begin{aligned} 3^\circ \text{ caso: } & \begin{cases} n = 84 \\ n \neq 84 \end{cases} \\ 4^\circ \text{ caso: } & \begin{cases} n = 84 \\ n = 84 \end{cases} \end{aligned}$$

Seja ainda a imagem constituída por:

$$X_1, Y_1, C_1, X_2, Y_2, C_2, \dots, X_n, Y_n, C_n$$

A disposição da imagem na fita é a seguinte:

1º caso:

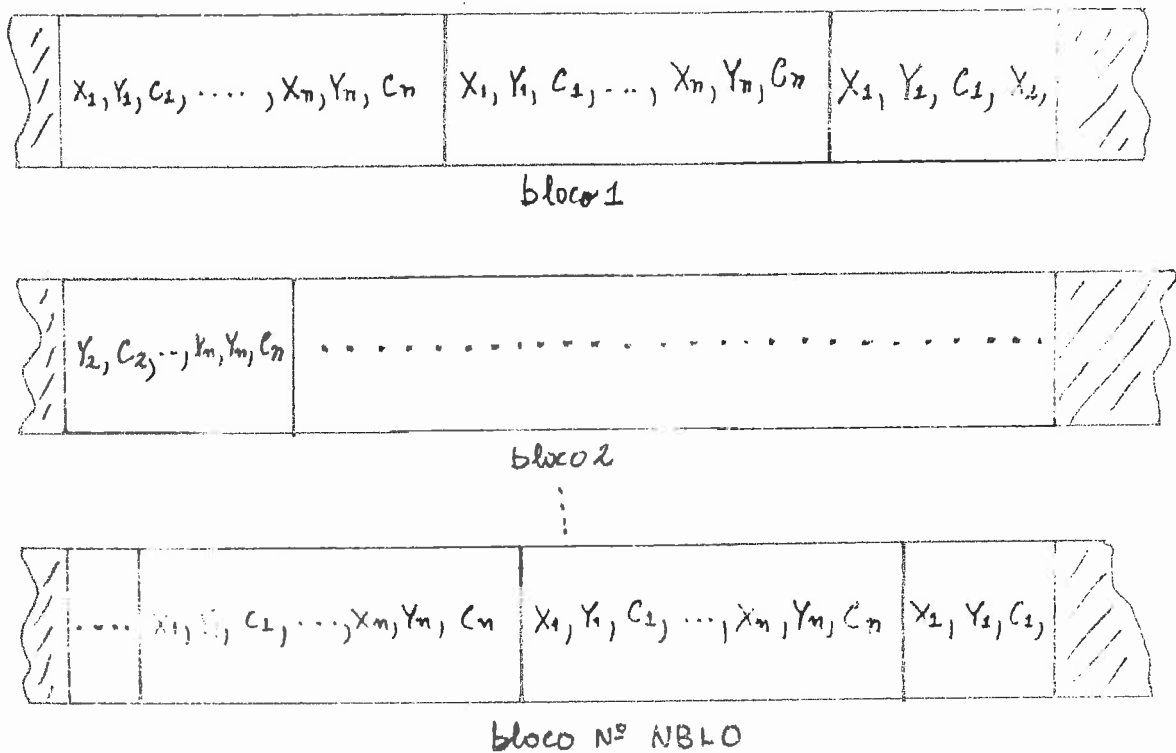


Fig. 11 - Disposição da imagem na fita no 1º caso.

Neste caso podemos observar o aparecimento de 2 tipos de problemas:

- 1) A imagem ficar partida entre 2 blocos;  
O resultado desta anomalia é uma pequena intermitência de um dos pontos da imagem. Como este ponto varia ao longo da representação não traz problemas de maior
- 2) A última imagem não ficar completa;  
O resultado desta anomalia é o último refrescamento ser incompleto o que também não traz problemas.

No 2º caso temos:

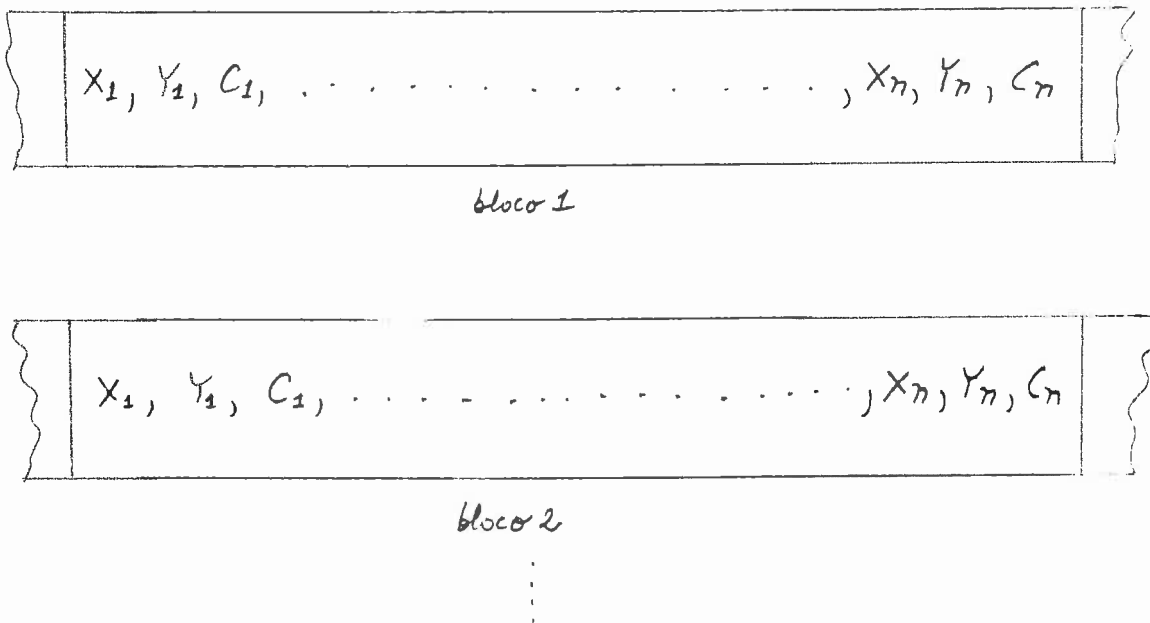
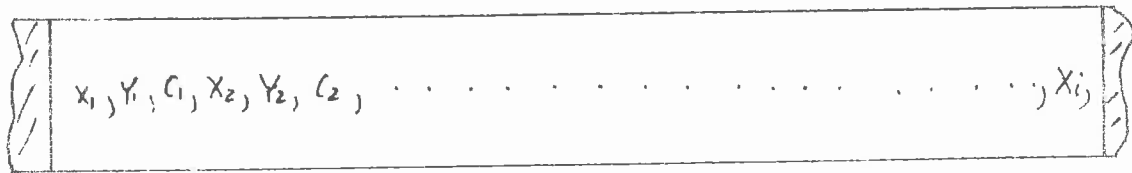
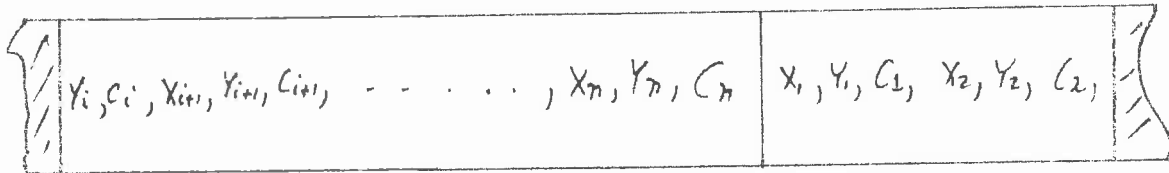


Fig. 12 - Disposição da imagem na fita no 2º caso.

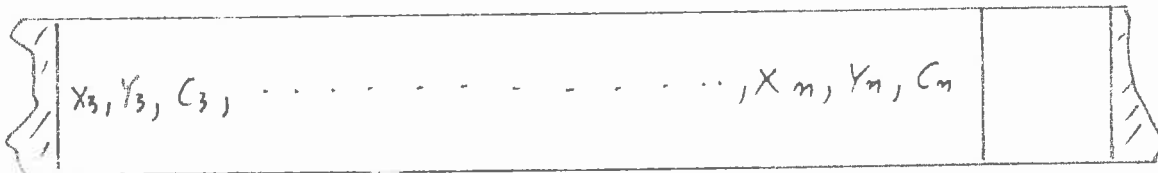
No 3º caso temos:



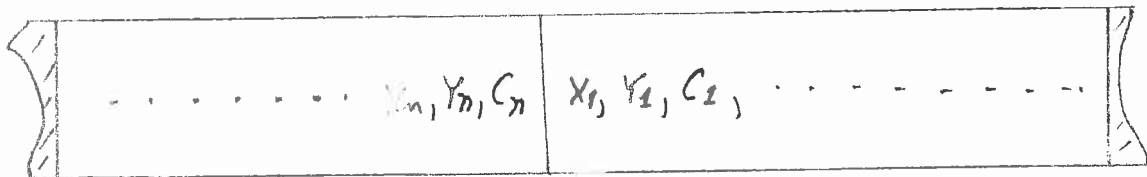
*bloco 1*



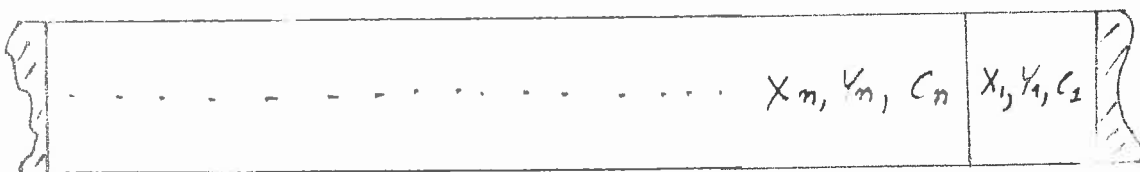
*bloco 2*



*bloco 3*



*bloco NBLO-1*



*bloco NBLO*

Fig. 13 - Disposição da imagem na fita no 3º caso.

Os problemas que podem eventualmente surgir neste caso são idênticos ao 1º caso.

O 4º caso é idêntico ao 2º.

Diversas experiências feitas com o tubo de raios catódicos levaram-nos à conclusão de que a frequência do refrescamento nunca deve ser inferior a 10/s para se obter uma imagem visível. Entre 10/s e 30/s a imagem embora visível apresenta intermitência e acima de 30 refrescamentos por segundo é boa.

Assim o número de pontos da imagem não deve ser superior a:

$$0,1/0,0266 \cong 4 \text{ blocos}$$

Ou seja não convém representar imagens que ocupem mais do que  $4 \times 86 = 344$  pontos, se se usar DISP em FORTRAN.

Como o número de blocos colocados na banda para representar uma imagem é dado por:

$$NBLO = INT (T/2.66) + 1$$

sendo T o tempo de representação desejado pelo utilizador em centésimas de segundo.

Assim, sendo T' o tempo de representação do número de blocos necessários a pelo menos um refrescamento da imagem podem suceder várias situações.

a)  $T' > T$

Neste caso a imagem nunca chega a ser totalmente refrescada pelo menos uma vez

b)  $T' < T$  com T' da mesma ordem de grandeza de T.

Neste caso a imagem é refrescada um número de vezes pouco superior a um o que é manifestamente insuficiente

c)  $T' \ll T$

Neste caso a representação é boa.

A discussão dos problemas de representação e permanência de imagens no écran será estudada no capítulo 6 de uma forma mais aprofundada.

A estrutura de dados de DISP é a seguinte:

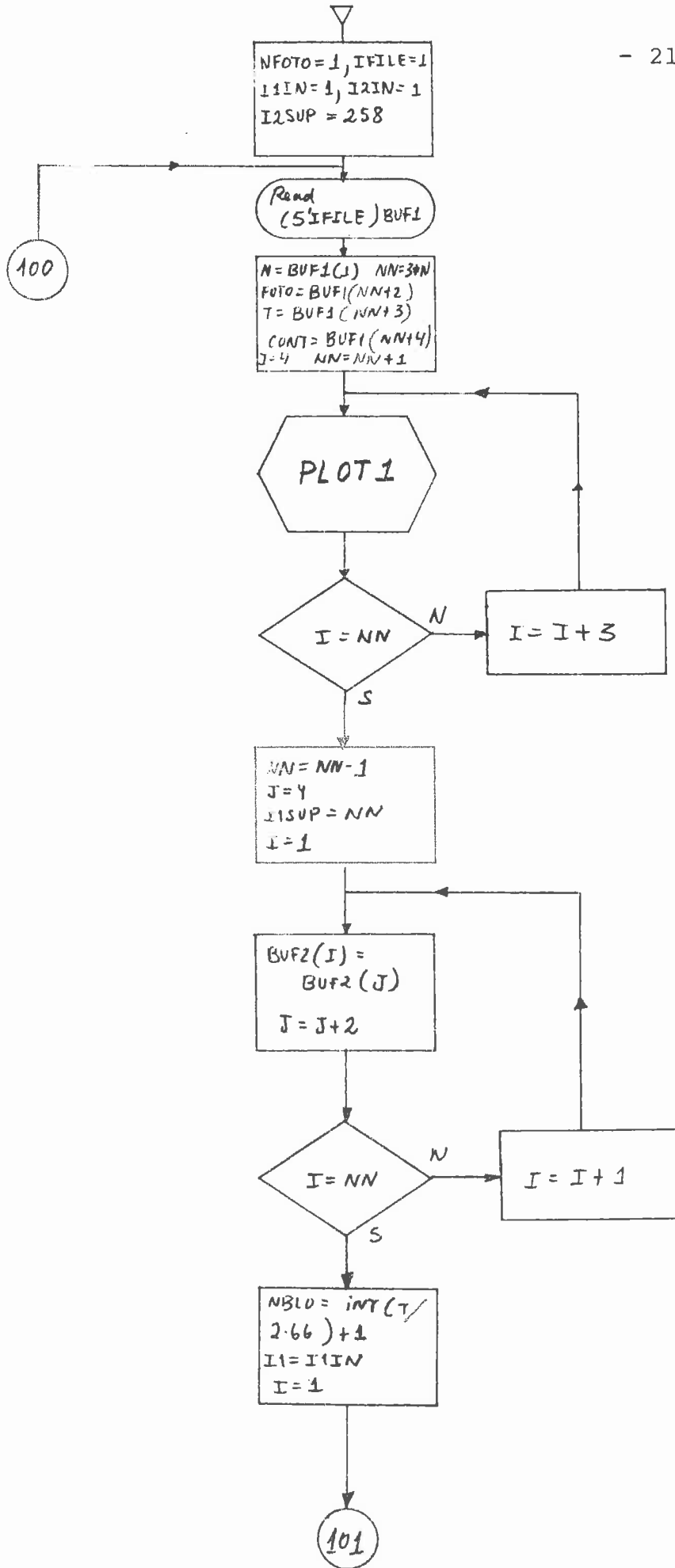
NOME	TIPO	TAMANHO	ESTRUTURA	FUNÇÃO
BUF1	Integer * 2	3004	vector	conter o registo do disco
BUF2	Logical * 1	6008	vector	construção da imagem a colocar na banda
BUF3	Logical * 1	252	vector	bloco a colocar na banda
N	Integer * 2		Inteiro	número de pontos
FOTO	Integer * 2		Inteiro	contém ou não fotografia
T	Integer * 2		Inteiro	indicação do tempo
CONT	Integer * 2		Inteiro	há ou não outra imagem a seguir

Nota: BUF1 e BUF2 coexistem na mesma área de memória.

Fig. 14 - Estrutura de dados de DISP.

Nesta subrotina o problema da fotografia foi resolvido da seguinte forma: a indicação de que deve ser tirada é automaticamente colocada no enésimo bloco da imagem. A fixação desse n é da subrotina e pode ser afinada com as experiências.

A acção da subrotina é descrita pela figura número 15.





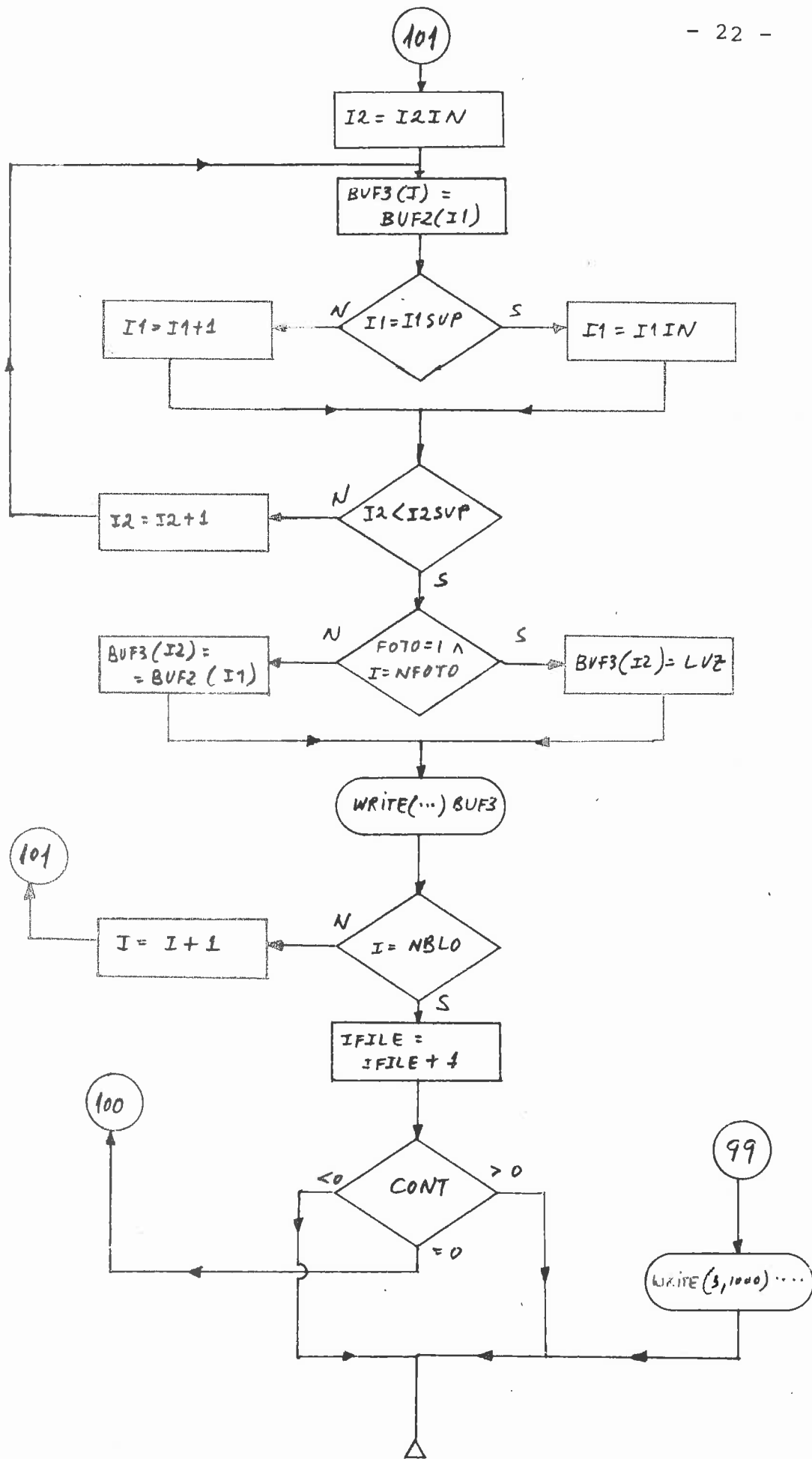


Fig. 15 - Organigrama de DISP.

A solução adoptada é como é óbvio extremamente deficiente no que respeita ao aproveitamento das potencialidades da banda, visto que cerca de 2/3 desta não contém informação. Torna-se assim evidente que é necessário utilizar a linguagem ASSEMBLER para a representação, esta solução será tratada no capítulo 5.

5. SOLUÇÃO FINAL DE PGBD

A experiência e a crítica a que foi sujeito PGBD assim como a solução final da escrita na banda utilizando a linguagem ASSEMBLER são descritos neste capítulo.

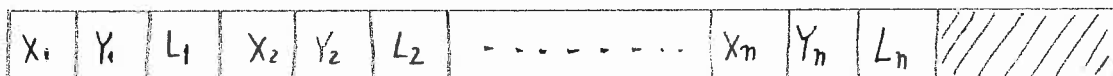
As alterações fundamentais são as seguintes:

O DISPLAY-FILE (DF) residia em disco e era constituído por um máximo de 20 registos de 6008 bytes sendo colocada toda a informação sobre uma dada imagem em cada um deles.

Este DF era constituído pela subrotina CLOSEI e excepcionalmente pela subrotina CLOSED quando fechava a utilização do G.D. sendo neste caso o último registo do DF de forma a conter a indicação de imagem final.

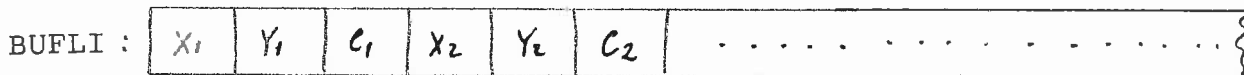
Esta forma de armazenamento da informação foi reconhecida como dispendiosa no que respeita ao tempo de I/O. Por outro lado as vantagens que proporcionava em relação à portabilidade do package eram extremamente limitadas. Assim foi abandonada e substituída pela solução que em seguida se descreve:

A imagem ao ser construída é colocada num vector (BUF) de comprimento 3.000 e do tipo INTEGER \* 2. Esse registo tem a forma:



em que cada  $X_i$  e  $Y_i$  são as coordenadas do ponto  $i$  e  $L_i$  a indicação sobre a sua luminosidade a qual pode tomar os valores 0,1 ou 2.

Existe outro vector, coexistindo com este na mesma área de memória (BUFLI) do tipo LOGICAL \* 1 e de comprimento 6.000. Este vector por um procedimento já conhecido recebe a imagem na forma com que vai ser colocada na banda.



em que:

X<sub>i</sub> e Y<sub>i</sub> são as coordenadas do ponto e c<sub>i</sub> o byte de controle.

A subrotina CLOSEI ao ser chamada, constrói um registo do D.F. e colocava-o de seguida no disco, recorrendo, à subrotina DISP que despeja o disco na banda.

A subrotina CLOSEI agora construída dá à imagem a forma final assinalada atrás e chama a subrotina DISP que partindo do conteúdo de BUFLI coloca directamente na banda a informação necessária para a representar.

Finalmente a última alteração respeita às subrotinas CLOSEI e CLOSED. A segunda foi pura e simplesmente suprimida e a primeira recebeu um novo parâmetro, END, que se tomar o valor 0 significa que não se trata da última imagem e se receber o valor 1 significa que se trata da última imagem e cessa então a utilização do G.D.

De todas as subrotinas de PGBD aquelas que sofreram alterações de fundo foram as subrotinas CLOSEI e DISP. Estas subrotinas serão apresentadas em pormenor. As alterações das outras subrotinas são de pormenor e não serão referidas senão sumariamente.

As subrotinas do package são:

- OPENDI
- OPENIM
- PLOT
- LINE
- CLOSEI
- DISP
- PLOT 1
- conjunto de subrotinas em ASSEMBLER

As estruturas de dados utilizadas por estas subrotinas são as seguintes:

NOME	TIPO	FUNÇÃO	ESTRUTURA ABSTRACTA
BC	1000.integer*2	conter as indicações de luminosidade dos pontos	vector
BX	1000.real*4	conter as coordenadas em X dos pontos	vector
BY	1000.real*4	conter as coordenadas em Y dos pontos	vector
IMAX	integer	número máximo de pontos da figura	_____
ALPON	logical	alarme de pontos	_____
NPONT	integer	número de pontos de uma figura	_____
BUF	3000.integer*2	conter o bloco que contém a figura	vector
BUFLI	6000.logical*1	conter o bloco que contém a figura	vector
BUFFER	15000.logical*1	BUFFER para escrita na banda	vector
CSCALE	integer	conter SCALE	_____
CTX	real	conter TX	_____
CTY	real	conter TY	_____
CCX	real	conter CX	_____
CCY	real	conter CY	_____
CT2X	real	conter T2X	_____
CT2Y	real	conter T2Y	_____
NFIG	integer	número de figuras	_____
N1	integer	número de pontos de uma figura	_____
FOTO1	integer	indicação sobre a fotografia	_____
T1	integer	tempo de representação da figura	_____
END1	integer	indicação sobre se é ou não a última figura	_____
NCONT	integer	apontador do vector BUFFER	_____

Nota: BUFLI e BUF coexistem na mesma área de memória através de uma instrução EQUIVALENCE.

Fig. 16 - Estruturas de dados de PGBD

Estas variáveis estão agrupadas em 2 áreas COMMON:

```
COMMON/XYZW1/IMAX, ALPON, NPONT, BC, BX, BY
```

e

```
COMMON/XYZW2/CSCALE, CTX, CTY, CCX, CCY, CT2X, CT2Y, NFIG,  
N1, FOTOL, T1, END1, BUF, BUFLI, BUFFER, NCONT.
```

A primeira área COMMON é comum às subrotinas, OPENDI, PLOT, LINE, OPENIM e CLOSEI e transmite os dados sobre uma dada figura.

A segunda área COMMON é comum às subrotinas, OPENDI, CLOSEI e DISP.

As funções das subrotinas OPENDI, OPENIM, PLOT, LINE e CLOSEI são essencialmente as mesmas donde as alterações que sofreram são mínimas.

As subrotinas CLOSEI e DISP passaram a ter novas funções. A subrotina CLOSEI toma as indicações sobre a figura contidas em BX, BC e BY e constrói o bloco a colocar na banda. A subrotina DISP vai colocando na banda os blocos sobre uma dada figura.

A subrotina PLOT1 permaneceu absolutamente inalterada.

Descrevemos em seguida as subrotinas CLOSEI e DISP.

```
SUBROUTINE CLOSEI (FOTO, T, END)  
INTEGER FOTO, T, END  
LOGICAL ALPON  
INTEGER * 2 BC (1000)  
DIMENSION BX (1000), BY (1000)  
INTEGER CSCALE, N1, FOTOL, T1, END1  
INTEGER * 2 BUF (3000)  
LOGICAL * 1 BUFLI (6000), BUFFER (15 000)  
EQUIVALENCE (BUFLI (1), BUF(1))  
COMMON/XYZW1/...  
COMMON/XYZW2/...
```

A acção desta subrotina consiste em tomar a imagem contida em BX, BY e BC, executar-lhe as transformações geométricas indicadas pelos parâmetros de OPENDI e armazená-la primeiro em BUF e depois da compactação já conhecida, em BUFLI.

Finalmente CLOSEI chama a subrotina DISP que tomando a in-

formação continua em BUFLI vai colocar na banda a informação necessária à representação da imagem.

O texto é o seguinte:

```
SUBROUTINE CLOSEI (FOTO, T, END)
INTEGER FOTO, T, END
LOGICAL ALPON
INTEGER * 2 BC (1000)
DIMENSION BX (1000), BY (1000)
INTEGER CSCALE, N1, FOTOL, T1, ENDL
INTEGER * 2 BUF (3000)
LOGICAL * 1 BUFFER (15000)
COMMON/XYZW1/IMAX, ALPON, NPONT, BC, BX, BY
COMMON/XYZW2/CSCALE, CTX, CTY, CCX, CCY, CT2X, CT2Y, NFIG,
C N1, FOTOL, T1, ENDL, BUF, BUFFER, NCONT
NPONT=NPONT+1
N1=NPONT
FOTOL=FOTO
T1=T
ENDL=END
IF (CSCALE-1) 20, 10, 30
10 XMIN=1.E20
YMIN=1.E20
XMAX=-1.E20
YMAX=-1.E20
DO 15 I=1, NPONT, 1
IF (BX(I).LT.XMIN) XMIN=BX (I)
IF (BX(I).GT.XMAX) XMAX=BX (I)
IF (BY(I).LT.YMIN) YMIN=BY (I)
15 IF (BY(I).GT.YMAX) YMAX=BY (I)
D1=XMAX-XMIN
D2=YMAX-YMIN
IF (D1-D2) 16, 17, 17
16 D=D2
GOTO 18
17 D=D1
18 CCX=250/D
CCY=CCY
```

```
CTX=- (XMAX+XMIN)/2.  
CTY=- (YMIN+YMAX)/2.  
CT2X=125.  
CT2Y=125.  
20 DO 25 I=1, NPONT  
   BX (I)=CCX * (BX(I)+CTX)+CT2X  
25 BY (I)=CCY * (BY(I)+CTY)+CT2Y  
30 I=1  
   DO 40 J=1, NPONT  
   BUF (I)=BX (J)  
   BUF (I+1)=BY (J)  
   BUF (I+2)=BU (J)  
40 I=I+3  
   NN=3* NPONT  
   DO 50 I=3, NN, 3  
   CALL PLOT1 (BUF(I))  
50 CONTINUE  
   CALL DISP  
   NFIG=NFIG+1  
   RETURN  
   END
```

Texto da subrotina CLOSEI.

A sub-rotina DISP tem por cabeça:

```
SUBROUTINE DISP  
INTEGER CSCALE, N1, FOT01, T1, END1  
INTEGER * 2 BUF (3000)  
LOGICAL * 1 BUFLi (6000), BUFFER (15000), LUZ, LUZ1  
DATA LUZ/Z10/, LUZ1/Z00/  
EQUIVALENCE (BUFLI(1), BUF(1))  
COMMON/XYZW2/.....
```

A subrotina DISP começa por determinar o número de Bytes que precisam de ser ocupados com informação da imagem para que esta seja representada durante o tempo T1.

Em seguida esses bytes são preenchidos por repetição dos bytes que compõem a imagem sendo a escrita na banda feita por grupos de 15.000, tamanho de BUFFER.

BUFFER tem um apontador, NCONT, que indica até onde está ocupado. Isto porque uma dada representação de uma imagem não ocupa necessariamente um número de bytes múltiplo de 15.000. No início da acção, os bytes da imagem começam a ser arrumados de NCONT para a frente. No fim do processamento NCONT terá um valor que é actualizado.

Quando se pretende tirar uma fotografia à imagem um dos bytes de controle levará essa indicação. Esse byte deve ser colocado na banda de tal forma que quando a máquina disparar a imagem já se encontra formada no écran. Esse tempo é variável e é indicado pelo valor da variável NFOTO. No capítulo das conclusões este assunto será novamente retomado.

Quando uma imagem é assinalada como sendo a última e o vector BUFFER não fica completamente preenchido com informações da imagem os restantes bytes são preenchidos com zeros.

O número de bytes necessários para representar uma dada imagem é calculado da seguinte forma:

$$NBYTES = T1 * 300$$

visto que 300 bytes correspondem exactamente a um tempo de representação de 0,01 segundos. Este ponto também será retomado no capítulo das conclusões.

As subrotinas em ASSEMBLER são 2, M800 e WRITE, e têm as seguintes funções: M800 fixa a escrita em 800 BPI e WRITE escreve na banda num só bloco a informação contida no vector BUFFER. (\*)

A acção da subrotina DISP é descrita na figura:

(\*) Estas subrotinas ASSEMBLER são da autoria de L. Castro.



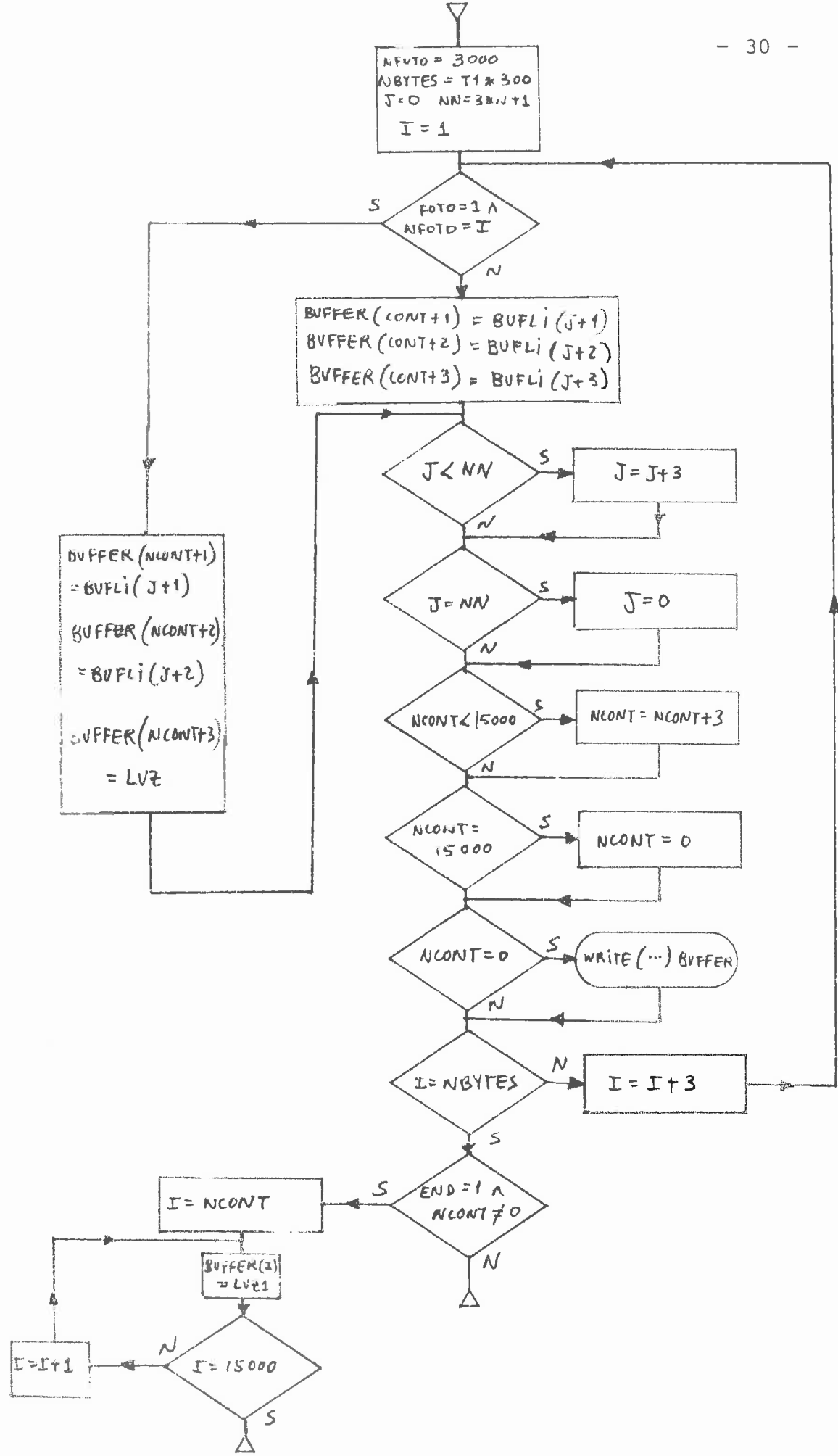


Fig. 17 - Acção da subrotina DISP.

## 6. CONCLUSÕES

No capítulo anterior apresenta-se a forma final de PGBD. Nessa solução a escrita na banda é feita pelas subrotinas WRITE e M800, escritas em ASSEMBLER.

De facto o FORTRAN permitia escrever em blocos de tamanho máximo de 252 bytes mas dado que o 'gap' ocupava uma zona equivalente a um bloco de aproximadamente 540 bytes isso levava a que sensivelmente 2/3 da banda estivessem inaproveitados. Assim, mesmo abandonando a solução inicial de fazer corresponder a uma imagem um bloco, o que limitava o número máximo de pontos de cada imagem a 84, as limitações deste modo de gravação eram evidentes. Considerações rigorosas sobre a capacidade de retenção da imagem do tubo de raios catódicos e sobre a capacidade da visão humana levaram a concluir que neste modo de gravação não seria possível representar sem intermitência imagens com mais de 153 pontos e o desenho animado só seria possível para imagens com um número de pontos extremamente pequeno.

A exploração das capacidades máximas da escrita em FORTRAN foi apresentada no capítulo 4 e a solução final, usando o ASSEMBLER, foi feita no capítulo 5.

A escrita na banda, neste segundo modo, é feita por blocos de 15.000 bytes. Atendendo a que cada 'gap' corresponde a sensivelmente 540 bytes, conclui-se que cerca de 1/30 da banda não contém informação para o G.D. Dadas as considerações sobre o refrescamento que em seguida serão apresentadas conclui-se que esta relação, ausência de informação/informação =  $\frac{1}{30}$  é suficiente e o seu efeito pode ser praticamente ignorado pelo utilizador.

No capítulo 2 discutiram-se alguns problemas do hardware que ainda estavam em aberto e cujas soluções finais foram apresentadas. Um destes foi a adopção do modo de gravação em 800 bpi's que por um lado trouxe simplificações técnicas mas por outro tem limitações. Neste modo de gravação a capacidade de transmissão da informação fica reduzida em média a

$$K. 30.000 \text{ bytes/s}$$

em que K é o factor de percas devido à existência do 'gap', ou seja, essa velocidade é sensivelmente

$$\frac{29}{30} \cdot 30.000 = 29.000 \text{ bytes/s} = 9.660 \text{ pontos/s}$$

O significado real deste valor será apreciado em seguida quando forem discutidos os problemas do refrescamento. Outra limitação importante deste modo de gravação é a sua rápida queda em desuso, donde a portabilidade da interface ser menor. Verifica-se no entanto que com a unidade de bandas de que se dispunha e os problemas de hardware que a utilização de 1.600 BPI'S exigia, era absolutamente desaconselhável este modo de gravação.

Problemas de hardware em aberto são os que dizem respeito à velocidade de deslocamento do feixe que, a ser constante, tem como consequência que um segmento mais curto apresente uma luminosidade maior pois o tempo de transição é sempre o mesmo entre 2 pontos consecutivos e imposto pela velocidade da fita magnética.

Se a velocidade for sempre a mesma qualquer que seja a distância haverá pontos mais luminosos quando se percorrer segmentos mais curtos que correspondem a passagens mais prolongadas do feixe.

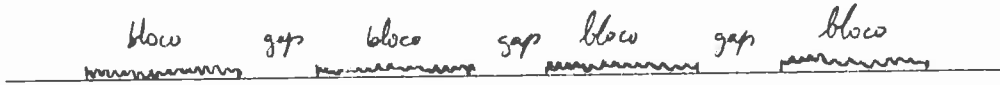
Todas as discussões que se vão seguir sobre o refrescamento ignoram este problema e consideram que o feixe transita sempre no tempo de 0.1 ms deixando a mesma intensidade luminosa, o que é falso, mas que nesta fase em que o G.D. ainda não funciona serve de hipótese de trabalho.

Finalmente um dos problemas hardware que também foi resolvido foi o da deteção do 'gap' que anteriormente era feito por software. Esta solução que decorre do modo de gravação limita a portabilidade da interface de máquina para máquina mas por sua vez simplifica o hardware.

O software agora construído está organizado de forma modular em subrotinas com uma certa hierarquia de funções e zonas de memória comuns para comunicação de dados.

Finalmente apresenta-se no seu estado actual a solução para o problema do refrescamento.

A informação é colocada sequencialmente na banda na seguinte forma:



Sendo cada bloco constituído por 5.000 pontos (15.000 bytes) e cada gap corresponde a 180 pontos (540 bytes). Uma dada imagem constituída por N pontos e a representar em T centésimas de segundo ocupa o seguinte espaço na banda.

Exemplo:



Sendo esse espaço correspondente a um número de pontos, não incluindo o gap, tal que o seu tempo de representação seja exactamente T. Esse número de pontos é dado por:

$$\begin{aligned} \text{NPONTOS} &= T * 100 \text{ ou, em termos de bytes,} \\ \text{NBYTES} &= T * 300 \end{aligned}$$

Estes números não são exactos quando no meio dos NPONTOS existe um 'gap' mas isso será analisado mais tarde.

Dado que a imagem tem N pontos o número NPONTOS será preenchido ciclicamente com sucessivas sequências dos N pontos até pre-  
fazer o número NPONTOS e portanto o número de refrescamentos será NPONTOS/N.

Para analisar o processo do refrescamento e alguns aspectos da visão que o influenciam foi realizada a seguinte experiência num osciloscópio de características médias:

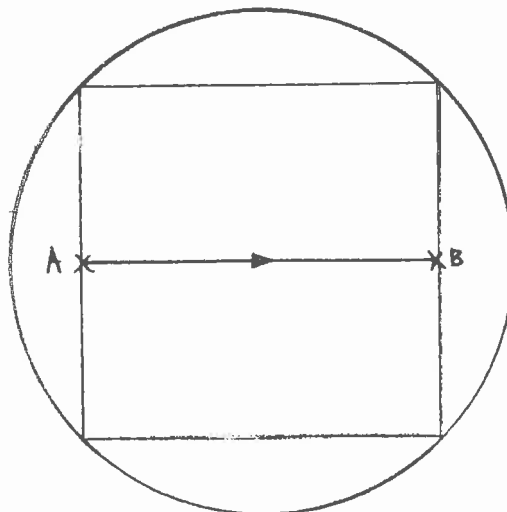


Fig. 18

O feixe era obrigado a percorrer na horizontal o ecran de A para B. O tempo de transição de A para B era fixo e igual a 0,1ms. A frequência das transições era ajustável e a intensidade luminosa também.

Verificou-se que, para uma intensidade luminosa média, o segmento  $\overline{AB}$  apresenta-se nitido e sem intermitência para refrescamentos de frequência superior a 30/s, enquanto que para frequências inferiores se apresentava intermitente.

Isto permite concluir que os segmentos de recta de uma imagem têm de ser refrescados pelo menos 30X por segundo.

Assim, dada uma imagem com N pontos e atendendo a que a capacidade de representação da interface é de 9.660 pontos/s a frequência de refrescamento de uma dada imagem é dada por:

$$f.N = 9.660$$

sendo f a frequência de refrescamento e N o número de pontos de uma imagem.

Se fizermos  $f=30$  (caso limite) conclui-se que o número máximo de pontos que uma imagem fixa pode ter para ser representada sem intermitência é:

$$N = \frac{9660}{30} = 322$$

Ou seja, apenas as imagens com  $N \leq 322$  pontos aparecem com nitidez. Se considerarmos que com uma frequência de refrescamento  $f=10$  mesmo assim a imagem ainda se consegue perceber temos que o número limite de pontos que uma imagem pode ter para ainda ser perceptível é aproximadamente 1000. Como a imagem com mais pontos é mais complexa, portanto ainda mais difícil de perceber, conclui-se que para além de 322 pontos a qualidade da representação decai rapidamente.

Repare-se que estas considerações têm por base o segmento de recta  $\overline{AB}$  que por ser muito grande significa que se está a analisar o caso mais desfavorável.

A vista humana só pode distinguir imagens desde que estas sejam representadas durante pelo menos 3,33 centésimas de segundo visto que a chamada frequência critica de fusão é de 30 imagens/s.

Tomando também em consideração que para se fazer desenho animado se terá que representar um certo número de imagens diferen-

tes por segundo, o utilizador virá a ficar com uma zona de utilização do desenho animado perfeitamente delimitada e que só a experiência poderá permitir definir.

Finalmente, o efeito da existência do 'gap' é o seguinte: Para imagens fixas a representar em tempos grandes o tempo de representação vem afectado de um factor suplementar de  $1/30.T$ . No entanto a imagem não será perturbada pelo 'gap' visto que o tempo do 'gap' é de 1,77 centésimas de segundo, inferior a 3,33 centésimas de segundo (frequência critica de fusão) e portanto a vista não se apercebe daquela ausência de imagem. Para imagens de tempo de representação muito curto, caso tipico do desenho animado, a imagem que receber o 'gap' no meio, vem bastante afectada mas também se poderá tornar mais ou menos indiferente no conjunto do desenho animado.

Quando se pretende obter uma fotografia da imagem há que ter em atenção que:

- 1 - A fotografia só pode ser tirada depois de a imagem estar nitida no écran.
- 2 - É necessário um determinado tempo de exposição.
- 3 - Há que contar com o tempo de reacção da máquina fotográfica.

Para isso existe um parâmetro que dá conta do tempo que vai, numa imagem que deve ser fotografada, desde o inicio da sua representação até ao disparo da máquina. Este parâmetro vai permitir, tomando em consideração aqueles factores atrás expostos, afinar a solução do problema.

Para todas as questões ainda em aberto, adoptou-se a solução de considerar determinadas hipóteses de fundo e criar depois a possibilidade de afinar os pormenores. Esta concepção leva a concluir que o trabalho agora descrito tem a sua forma definitiva no que respeita à sua arquitectura e concepção de fundo, donde a este relatório se seguirá imediatamente o manual do utilizador.

7. BIBLIOGRAFIA

Além da bibliografia referida no primeiro destes relatórios foi ainda usada a seguinte:

- 1 - Human Information Processing  
Lawrence J. Fogel  
Prentice Hall Inc., New Jersey, 1967
- 2 - A Psicologia da Visão  
R. L. Gregory  
Biblioteca Universitária Inova  
Editorial Inova Limitada, Porto, Portugal.