PC Assembler

USANDO O BIOS

Daniel G. A. QUADROS

ÍNDICE

ÍNDICE **2**

PREFÁCIO DA EDIÇÃO ELETRÔNICA **4**

PREFÁCIO DA EDIÇÃO ORIGINAL **5**

1 INTRODUÇÃO **6**

1.1 O QUE É O BIOS 6
1.2 POR QUE USAR O BIOS 6
1.3 OBJETIVOS DESTE LIVRO 6
1.4 O ACESSO AO BIOS 7
1.5 MÓDULOS DO BIOS 7
1.6 UMA RÁPIDA REVISÃO DE CONCEITOS 7
1.7 UMA VISÃO RÁPIDA DO HARDWARE DO PC 8

2 ACESSO AO TECLADO **11**

2.1 CARACTERÍSTICAS DO HARDWARE 11
2.2 ESTRUTURA DE DADOS 11
2.3 FUNÇÕES DISPONÍVEIS 11
2.4 EXEMPLO 12

3 ACESSO AO VÍDEO **13**

3.1 CARACTERÍSTICAS DO HARDWARE 13
3.2 ESTRUTURA DE DADOS 15
3.3 FUNÇÕES DISPONÍVEIS 16
3.4 EXEMPLO 20

4 ACESSO AO DISQUETE **21**

4.1 CARACTERÍSTICAS DO HARDWARE 21
4.2 ESTRUTURA DE DADOS 21
4.3 FUNÇÕES DISPONÍVEIS 22
4.4 EXEMPLO 25

5 ACESSO AO DISCO RÍGIDO **26**

5.1 CARACTERÍSTICAS DO HARDWARE 26
5.2 ESTRUTURA DE DADOS 26
5.3 FUNÇÕES DISPONÍVEIS 27
5.4 EXEMPLO 31

6 ACESSO À IMPRESSORA **32**

6.1 CARACTERÍSTICAS DO HARDWARE 32
6.2 ESTRUTURA DE DADOS 32
6.3 FUNÇÕES DISPONÍVEIS 32
6.4 EXEMPLO 33

7 ACESSO ÀS INTERFACES SERIAIS **34**

7.1 CARACTERÍSTICAS DO HARDWARE 34
7.2 ESTRUTURA DE DADOS 34

7.3 FUNÇÕES DISPONÍVEIS 34
7.4 EXEMPLO 36

8 ACESSO AO RELÓGIO 37

8.1 CARACTERÍSTICAS DO HARDWARE 37
8.2 ESTRUTURA DE DADOS 37
8.3 FUNÇÕES DISPONÍVEIS 38
8.4 EXEMPLO 38

9 OUTRAS FUNÇÕES DO BIOS 39

9.1 INFORMAÇÃO DA MEMÓRIA DISPONÍVEL 39
9.2 INFORMAÇÃO DOS DISPOSITIVOS DISPONÍVEIS 39
9.3 IMPRESSÃO DE TELA 39
9.4 LEITURA DO CARREGADOR DO SISTEMA OPERACIONAL 40
9.5 TESTES INICIAIS 40

10 UMA ROTINA PARA ACESSO AO ALTO-FALANTE 42

10.1 INTRODUÇÃO 42
10.2 CARACTERÍSTICAS DO HARDWARE 42
10.3 DESCRIÇÃO DA ROTINA 43
10.4 LISTAGEM DA ROTINA 43

11 PROGRAMAS EXEMPLO 48

11.1 COMO ENTRAR COM ESTES PROGRAMAS EM UM PC 48
11.2 QUADRO DE AVISOS 48
11.3 DUPLICADOR DE DISCO 57

APÊNDICE A – MAPA DA MEMÓRIA 69

APÊNDICE B – CÓDIGOS DO TECLADO 70

APÊNDICE C – SÍMBOLOS DO VÍDEO 73

PREFÁCIO DA EDIÇÃO ELETRÔNICA

É com grande prazer e imenso atraso que eu disponibilizo o meu segundo livro em "formato eletrônico".

Em 1986, com o relativo sucesso do primeiro volume do **PC Assembler**, acertei rapidamente a publicação deste segundo volume, **PC Assembler – Usando o BIOS**. Novamente o livro foi escrito em um Unitron (clone do Apple][) e os originais foram enviados impressos para serem redigitados e compostos na editora.

O livro também teve o seu sucesso: foram impressos 5305 livros em três edições, do segundo semestre de 87 até o início de 98, quando a Editora Campus decidiu não mais reeditar a obra e me devolveu os direitos de publicação.

Esta "edição eletrônica" segue a formatação que adotei com o primeiro volume, publicado há quase dois anos atrás.

Espero que seja útil para alguém.

Daniel Quadros

Abril/2012

PREFÁCIO DA EDIÇÃO ORIGINAL

Este livro é resultado da minha experiência profissional com micros compatíveis com o PC-IBM, em particular meu envolvimento no projeto do programa monitor do microcomputador Nexus. Gostaria, portanto, de dedicá-lo a Alberto Koga, Chen, Eduardo Fujii, Elisabeth Yoshida, Luiz Claudio Navarro, Marcio Dondon e a todos os demais amigos da Scopus que direta ou indiretamente estiveram ligados a este projeto.

Gostaria também de agradecer à **Humana Informática** o apoio fornecido, sem o qual esta obra não teria sido viável.

Finalmente, um agradecimento especial à paciência de minha esposa e sua colaboração na revisão do original.

Daniel Quadros,

São Paulo 11 de dezembro de 1986

1 INTRODUÇÃO

1.1 O QUE É O BIOS

O ROM BIOS (*Read Only Memory based Basic Input Output System*) ou simplesmente BIOS, é um conjunto de rotinas armazenadas em memória não volátil nos computadores da linha PC.

Estas rotinas interagem diretamente com o hardware, sendo responsáveis pela iniciação e acesso às interfaces instaladas.

1.2 POR QUE USAR O BIOS

O programador Assembler tem no PC-IBM três maneiras de acessar o hardware:

- Diretamente, através de instruções de entrada e saída
- Através do BIOS
- Através do sistema operacional

O acesso via BIOS permite obter um desempenho e um controle próximos ao do acesso direto, sem a necessidade de um conhecimento profundo do hardware. O BIOS é capaz também de mascarar diferenças de hardware, aumentando a portabilidade do programa que o acessa.

Portanto, usa-se o BIOS quando:

- Necessita-se de maior controle
- Necessita-se maior desempenho
- Deseja-se aumentar a portabilidade

Um exemplo típico é o acesso ao vídeo. O acesso via sistema operacional e relativamente lento e não permite, por exemplo, o posicionamento do cursor. O acesso direto é extremamente rápido, porém exige o conhecimento do hardware e tem variações conforme a interface de vídeo utilizada. O acesso via BIOS representa um bom compromisso, com a obtenção de uma boa velocidade e funções extras em relação ao sistema operacional, acessadas de forma independente da interface utilizada.

1.3 OBJETIVOS DESTE LIVRO

O objetivo principal deste livro é fornecer ao programador assembler informações para que ele possa usar as funções do BIOS em seus programas.

Juntamente com a descrição das funções são fornecidos exemplos e sugestões, bem como lembretes quanto a restrições existentes.

Embora uma apresentação profunda do hardware do PC fuja do escopo deste livro, cada capítulo contem uma curta descrição do hardware envolvido.

1.4 O ACESSO AO BIOS

O acesso às funções do BIOS é sempre feito através de interrupções de software. Os parâmetros de entrada e os resultados são normalmente passados através de registradores. Normalmente, apenas os registradores utilizados para resultados são alterados; entretanto deve-se tomar cuidado, pois existem algumas exceções.

Uma chamada típica ao BIOS é feita da seguinte forma:

- Salvam-se os registradores alterados pela função e cujo conteúdo não se deseja perder
- Colocam-se os parâmetros de entrada nos respectivos registradores
- Chama-se a função, através de uma instrução INT
- Recuperam-se os registradores salvos

1.5 MÓDULOS DO BIOS

O BIOS está organizado como uma série de grupos de rotinas, aqui denominadas módulos. Os módulos que serão apresentados neste livro são os seguintes:

- Acesso ao teclado
- Acesso ao vídeo
- Acesso ao disquete
- Acesso ao disco rígido (winchester)
- Acesso à impressora
- Acesso às interfaces seriais
- Acesso ao relógio
- Informação da memória disponível
- Informação dos dispositivos disponíveis
- Impressão da tela (*hardcopy*)
- Leitura do carregador do sistema operacional
- Testes iniciais

Em alguns modelos de PC, existem ainda outros módulos que não serão aqui discutidos, como, por exemplo, as rotinas de acesso à fita cassete existentes no PC-IBM original.

1.6 UMA RÁPIDA REVISÃO DE CONCEITOS

O microprocessador 8088 manipula dados de 8 bits (**BYTE**s), 16 bits (**WORD**s) e, em algumas ocasiões especiais, 32 bits (**DOUBLE WORD**s). Os *double words* normalmente armazenam endereços, que no 8088 são constituídos de duas partes: um **SEGMENTO** e um **OFFSET**, ambos de 16 bits. O endereço "físico" de uma posição é obtido multiplicando o segmento por 16 e somando o offset, de forma a se obter um endereço de 20 bits. Os words são armazenados na memória com o byte MENOS significativo no endereço

MENOR; os double words são armazenados com o word MENOS significativo no endereço MENOR. Por exemplo, o endereço 1234H:5678H seria armazenado a partir do endereço 2000H da seguinte forma:

Endereço	conteúdo
2000H	78H
2001H	56H
2002H	34H
2003H	12H

(Neste livro adotamos a convenção de indicar números hexadecimais através do sufixo "H", como no exemplo acima.)

Os bits de um dado são numerados a partir de 0, que é o bit menos significativo. Desta forma o bit mais significativo de um byte é o bit 7 e o mais significativo de um word é o 15.

O 8088 possui 13 registradores de 16 bits: DS, ES, CS, SS, PC, SP, BP, SI, DI, DX, CX, BX e AX; os quatro últimos podem ser manipulados como pares de registradores de 8 bits (DH, DL, CH, CL, BH, BL, AH e AL). Os quatro primeiros são os REGISTRADORES de SEGMENTO, que contêm os segmentos de dados (DS e ES), código (CS) e pilha (SS).

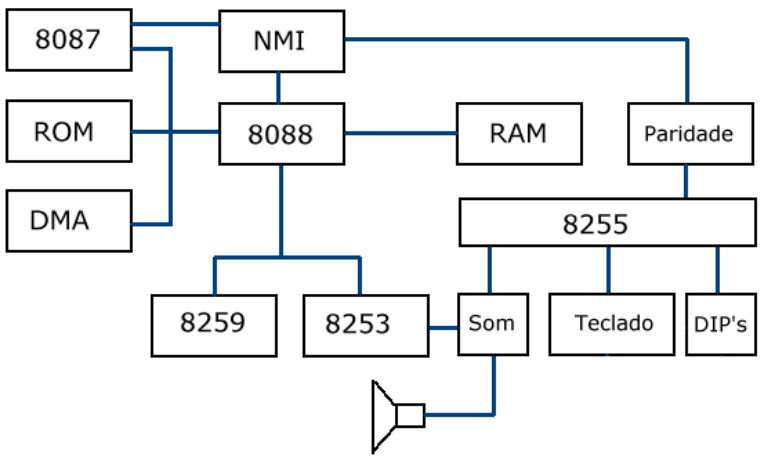
Vamos nos referir com frequência a INTERRUPÇÕES de SOFTWARE e HARDWARE. As primeiras são desvios causados pela execução da instrução INT e as segundas são desvios causados pela sinalização de um dispositivo externo. O 8088 dispõe de 256 interrupções, armazenando os endereços das rotinas que as tratam em double words, a partir do início da memória (endereço 0). O endereço da rotina que trata a interrupção "i" está armazenado no double word de endereço "4*i".

Para conhecer maiores detalhes sobre o 8088 sugere-se a leitura do primeiro volume do "PC Assembler".

1.7 UMA VISÃO RÁPIDA DO HARDWARE DO PC

O hardware do PC é bastante modular, sendo normalmente composto de uma placa de sistema que dispõe de conectores ("*slots*") nos quais são conectadas as várias placas de expansão. Neste item, vamos apresentar rapidamente a placa de sistema; outras placas de expansão serão apresentadas nos capítulos que tratam dos módulos do BIOS que com elas interagem.

A figura abaixo mostra, em grandes blocos, a arquitetura da placa de sistema.



Vamos agora analisar os diversos blocos:

8088 – Microprocessador

É um processador com via interna de 16 bits e via externa de 8 bits, capaz de endereçar diretamente 1 Mbyte de memória e 64 K endereços de E/S. No PC original e na maioria dos compatíveis, trabalha com uma frequência de 4.77 MHz.

8087 – Coprocessador numérico

Permite a rápida execução de operações aritméticas, logarítmicas e trigonométricas, com alta precisão.

ROM – Memória permanente (não volátil)

Contém o BIOS e, no caso do PC IBM, o ROM BASIC (linguagem BASIC sem comandos de acesso a disco).

RAM – Memória de leitura/escrita

Localizada a partir do endereço 00000H, o tamanho desta memória varia de modelo para modelo; os tamanhos mais comuns são 256, 512, 640 e 704 Kbytes. O PC utiliza detecção de erro através de paridade na memória RAM, armazenando cada byte em 9 bits (o nono bit não é acessível diretamente).

DMA – Acesso direto à memória

É composto de um controlador de DMA 8037 da Intel, capaz de endereçar 64 Kbytes e um conjunto de registradores (denominados registradores de página de DMA) que contém os 4 bits mais significativos do endereço de transferência. Por este motivo, uma transferência via DMA deve ter os 4 bits mais significativos de DMA fixos. O 8037 dispõe de 4 canais de DMA, sendo um deles utilizado para o refrescamento da memória RAM, não estando, portanto, disponível para as placas de expansão.

8259 – Controlador de interrupções

Gerencia 8 níveis de interrupção, que são mapeados para os vetores 08H a 0FH. Os dois primeiros níveis são usados internamente à placa de sistema: o nível 0 (INT 8) corresponde a uma interrupção de tempo real gerada pelo temporizador 8253 e o nível 1 (INT 9) corresponde à chegada de um código do teclado. O controle das interrupções é feito através do registrador de máscara da 8259, acessível por instruções de entrada e saída no endereço 21H. Os bits em "1" neste registrador correspondem a interrupções MASCARADAS, isto é, interrupções não permitidas. Ao final do tratamento de uma interrupção deve-se escrever o comando "*End Of Interrupt*" (EOI - 20H) no registrador de comandos da 8259 (endereço de E/S 20H)

8253 – Temporizador

O temporizador/contador 8253 dispõe de 3 canais independentes e recebe como entrada uma frequência de 1.193.180 Hz. O canal 0 está conectado ao nível 0 de interrupção, o canal 1 é utilizado no refrescamento da memória, e o canal 2 está conectado ao circuito de geração de sons. A programação de um canal da 8253 é feito escrevendo-se uma palavra de controle no

registrador de controle (endereço de E/S 43H) e em seguida a contagem inicial no registrador correspondente ao canal (canal 0 - 40H, 1 - 41H, 2 - 42H). O formato da palavra de controle é:

bits 7 e 6	00 - seleciona canal 0
	01 - seleciona canal 1
	10 - seleciona canal 2
bits 5 e 4	00 - prepara leitura do contador
	01 - ler/carregar somente o byte menos significativo
	10 - ler/carregar somente o byte mais significativo
	11 - ler/carregar primeiro o byte menos significativo e depois o mais significativo
bits 3 a 0	0110 - seleciona modo divisor binário

8255 – Interface paralela

Esta interface é programada de forma a ter dois registradores de leitura e um de escrita. O primeiro registrador de leitura (endereço de E/S 60H) é utilizado para ler os códigos enviados pelo teclado. O segundo registrador de leitura (endereço de E/S 62H) permite ler as seguintes informações:

bit 7	1 se ocorreu erro de paridade na RAM da placa de sistema
bit 6	1 se ocorreu erro de paridade em expansão de memória
bit 5	saída o canal 2 da 8253
bit 4	reservado
bits 3 a 0	chaves de configuração. O significado destas chaves varia de modelo para modelo

O registrador de escrita fica no endereço de E/S 61H e tem as seguintes funções:

bit 7	"0" permite a recepção de códigos do teclado; deve ser pulsado em "1" apos a leitura do código
bit 6	"1" na operação normal, mantido em "0" por 20 ms re-inicia o teclado
bit 5	"0" habilita teste de paridade na RAM da placa de sistema.
	"1" desabilita o teste e limpa erro anterior
bit 4	idem ao bit 5, para a expansão de memória
bit 3	reservado (no PC XT da IBM seleciona o conjunto de chaves de configuração a ser lido)
bit 2	não usado
bit 1	controla o alto-falante
bit 0	controla o canal 2 da 8253

Apesar deste registrador ser de escrita, ele pode ser lido para obter a programação anterior.

NMI – Interrupção não mascarável

Uma interrupção não mascarável pode ser gerada por um erro de paridade ou pelo coprocessador numérico 8087. A escrita de 080H no endereço de E/S 0A0H habilita a geração desta interrupção; a escrita de 00H neste mesmo endereço a desabilita.

2 ACESSO AO TECLADO

2.1 CARACTERÍSTICAS DO HARDWARE

O teclado dos micros compatíveis com o PC-IBM se caracteriza por possuir um microprocessador próprio e por enviar um código (de 8 bits) para o processador principal sempre que uma tecla for pressionada (ou mantida pressionada por certo tempo) ou liberada. Este código, denominado pela IBM de "*scan code*" (código de varredura), possui o bit mais significativo igual a "1" se a tecla foi liberada ou "0" se a tecla foi apertada (ou mantida pressionada). Os demais bits indicam qual a tecla detectada (ver apêndice B).

O envio deste código ao 8088 é sinalizado através de uma interrupção de hardware (INT 09H). O tratamento desta interrupção é normalmente efetuado pelo BIOS, que recolhe e trata o código fornecido pelo teclado. Este tratamento consiste em:

- Ler o código da tecla, no port da 8255 de endereço 60H
- Sinalizar a leitura ao teclado, pulsando o bit 7 do port da 8255 de endereço 62H (colocando-o em "1" e logo em seguida retornando-o a "0")
- Atualizar os estados das teclas alteradoras de código (Shift, Control, Alt, Caps Lock, Num Lock e Scroll Lock)
- Suprimir a auto repetição da tecla Ins e as combinações inválidas de teclas
- Tratar combinações especiais (Ctrl Alt Del, Ctrl Break, Ctrl Num Lock e Shift PrtSc)
- Gerar o código apropriado para as demais combinações e o armazenar na fila de teclado

Um tratamento especial realizado na liberação da tecla Alt permite gerar todos os códigos de 1 a 255 através da digitação do seu código decimal no teclado numérico, com a tecla Alt pressionada. Ao se liberar a tecla Alt o código correspondente é gerado.

Os códigos colocados na fila de recepção estão codificados em 16 bits, no que a IBM denomina "*Extended ASCII*" (ASCII expandido), que consiste no código ASCII (*American Standard Code for Information Interchange*) acrescido de códigos especiais (os "*extended codes*" ou códigos expandidos). Os códigos ASCII normais se caracterizam por terem o byte mais significativo igual a zero e os "*extended codes*" por terem o byte menos significativo igual a zero.

2.2 ESTRUTURA DE DADOS

A principal estrutura de dados utilizada é a fila de teclado, normalmente com 15 posições, utilizada para armazenar os códigos gerados em tempo de interrupção até que os eles sejam lidos pelo programa que está sendo executado.

2.3 FUNÇÕES DISPONÍVEIS

O acesso às funções do teclado é feito através de INT 16H, com AH contendo o número da função desejada:

AH = 0 **Lê um código do teclado**

Aguarda existir um código disponível na fila do teclado e o retira da fila, retornando-o em AX.

AH = 1 **Informa se há tecla disponível**

Retorna com o flag Z ativo se não houver tecla disponível. Caso contrário, retorna com o flag Z inativo e com o código da próxima tecla a ser lida em AX (o código NÃO é retirado da fila).

AH = 2 **Informa estado das teclas alteradoras**

Retorna em AL o estado com a seguinte codificação:

Bit	Ativo se
7	Ins travado
6	Caps Lock travado
5	Num Lock travado
4	Scroll Lock travado
3	Alt pressionado
2	Control pressionado
1	Shift esquerdo pressionado
0	Shift direito pressionado

Em todas as funções o conteúdo de AX é alterado; na função chamada por AH = 1 os flags são também alterados.

2.4 EXEMPLO

O trecho abaixo espera um código estar disponível ou a tecla Shift esquerda ser pressionada:

```
ESPERA:
        MOV     AH,2
        INT     16H
        TEST    AL,2
        JNZ     SHIFT_ESQ       ; desvia, se shift esquerdo pressionado

        MOV     AH,1
        INT     16H
        JZ      ESPERA          ; repete, se não há código disponível

        MOV     AH,0            ; tem código disponível
        INT     16H             ; AX = código
```

3 ACESSO AO VÍDEO

3.1 CARACTERÍSTICAS DO HARDWARE

Inicialmente, vamos definir alguns conceitos:

- Vídeo alfanumérico é um vídeo onde a leitura e escrita são feitas em termos de caractere
- Vídeo gráfico é um vídeo onde a leitura e escrita são feitas em termos de pontos
- Atributo é um modificador que determina a apresentação de um caractere (reverso, intenso, sublinhado, etc.)
- Cor de frente é a cor dos pontos acesos de um caractere
- Cor de fundo é a cor dos pontos apagados de um caractere

O BIOS dos micros compatíveis com o PC-IBM suporta dois tipos de placas controladoras de vídeo:

- *Monochrome Display Adapter*

  Capaz de gerar apenas vídeo alfanumérico monocromático de alta definição

- *Color/Graphics Adapter*

  Capaz de gerar vídeo alfanumérico e gráfico, ambos coloridos.

3.1.1 "*MONOCHROME DISPLAY ADAPTER*"

Esta placa possui um único modo de operação, gerando um vídeo de 25 linhas de 80 caracteres. Cada caractere é apresentado na tela dentro de um bloco de 9x14 pontos, ocupando uma região de 7x9 pontos. Isto permite a obtenção de caracteres de alta definição, próprios para a leitura de texto com baixa fadiga visual.

A tela é armazenada em memória contida na própria placa, de endereço inicial 0B0000H. Cada posição da tela corresponde a 2 bytes nesta memória. O primeiro (posição par) determina o caractere a ser apresentado (ver apêndice C). O segundo (posição impar) contem o atributo do caractere:

Bit 7 = 1 → piscante
Bit 3 = 1 → intenso
Bits 6 5 4 2 1 0
1 1 1 0 0 0 → reverso
0 0 0 1 1 1 → normal
0 0 0 0 0 1 → sublinhado
0 0 0 0 0 0 → invisível
Demais → normal

O armazenamento da tela na memória é feito sequencialmente, ou seja, o caractere na linha **l** (0 a 24) e coluna **c** (0 a 79) fica armazenado no endereço

```
0B0000H + 160 * l + c
```

A memória de tela está sempre acessível ao processador, sem ocorrência de interferência no vídeo.

3.1.2 "COLOR/GRAPHICS ADAPTER"

Esta placa possui quatro modos de operação:

- Alfanumérico, 25 linhas de 40 caracteres, 16 cores
- Alfanumérico, 25 linhas de 80 caracteres, 16 cores
- Gráfico, 200 linhas de 320 pontos, 4 cores
- Gráfico, 200 linhas de 640 pontos, monocromático

Em todos os modos a tela é armazenada em uma memória de 16Kbytes, de endereço inicial 0B8000H, contida na própria placa. O BIOS suporta 8 páginas independentes de vídeo no modo alfanumérico de 40 colunas e 4 páginas no modo alfanumérico de 80 colunas.

O armazenamento da tela no modo alfanumérico é semelhante ao já visto na "*Monochrome Adapter*", mudando apenas a codificação do atributo:

- Bit 7 - este bit pode ser programado como atributo piscante (programação normal do BIOS) ou como intensidade da cor de fundo.
- Bits 6,5,4 -selecionam a cor de fundo do caractere, correspondendo respectivamente às cores Vermelho (R - Red), Verde (G - Green) e Azul (B -Blue)
- Bit 3 -intensidade da cor de frente (1 = intenso)
- Bits 2,1,0 - selecionam a cor de frente, correspondendo respectivamente a R,G e B

O caractere correspondente à linha **l** (0 a 25) e coluna **c** (0 a 79) da página **p** (0 a 3) é armazenado no endereço

```
0B8000H + 01000H * p + 160 * l + c
```

No modo 40 colunas (c de 0 a 39) temos

```
0B8000H + 00800H * p + 80 * l + c
```

Notar que em ambos os casos há um arredondamento do tamanho da página: de 4000 para 4096 (80 colunas) e de 2000 para 2048 (40 colunas).

Nos modos gráficos as linhas pares são armazenadas a partir do endereço 0B8000H e as ímpares a partir do endereço 0BA000H.

No modo 200 linhas de 320 pontos ("media resolução") cada byte na memória corresponde a 4 pontos na tela:

```
a0 a1 b0 b1 c0 c1 d0 d1
```

Onde a0 e a1 determinam a cor do primeiro ponto, b0 e b1 do ponto a sua direita e assim por diante. Os dois bits associados a cada ponto permitem selecionar uma dentre quatro cores, que podemos chamar de cores "lógicas". A cor lógica 0 pode ser escolhida livremente dentre 16 cores; para as demais cores existem dois conjuntos possíveis:

Cor lógica	Primeiro conjunto	Segundo conjunto
1	cian (B + G)	verde (G)
2	magenta (B + R)	vermelho (R)
3	branco (R + G + B)	amarelo (R + G)

A escolha do conjunto e da cor 0 é fixa para toda a tela. Notar que este esquema proíbe algumas combinações de cores (por exemplo, não e possível desenhar uma bandeira brasileira - verde, amarelo, azul e branco simultâneos).

Neste modo o ponto na linha **l** (0 a 199) coluna **c** (0 a 319) fica no byte de endereço:

Se **l** par `0B8000H + 80 * l + c / 4`

Se **l** impar `0BA000H + 80 * l + c / 4`

No modo 200 linhas de 640 pontos ("alta resolução") as coisas são mais simples: a cada ponto da tela corresponde um bit que é 1 se o ponto está aceso ou 0 se o ponto está apagado. O bit mais significativo de cada byte corresponde ao ponto mais à esquerda.

O endereço correspondente ao ponto na linha **l** (0 a 199) e coluna **c** (0 a 639) e:

Se **l** par `0B8000H + 80 * l + c / 8`

Se **l** impar `0BA000H + 80 * l + c / 8`

O acesso à memória de vídeo no modo alfanumérico 80 colunas só deve ser feito quando a imagem da tela não estiver sendo refrescada, sob o risco de se gerar perturbações na imagem. Isto pode ser feito lendo-se a porta de entrada de endereço 03DAH; o bit 0 indica o retraço horizontal e o bit 3 o retraço vertical. O retraço vertical ocorre 60 vezes por segundo e sua duração é suficiente para o acesso a cerca de 40 words da memória de vídeo; o retraço horizontal ocorre 1200 vezes por segundo, porém sua duração só permite o acesso a um word da memória de vídeo.

Nos demais modos a memória de vídeo pode ser acessada a qualquer instante, sem afetar a imagem.

3.2 ESTRUTURA DE DADOS

Nas operações de leitura e escrita de caracteres, o BIOS refere- se a um ponteiro, denominado **cursor**, que aponta para a linha e coluna onde será realizada a operação.

Conforme já dito, o BIOS suporta várias páginas independentes de vídeo na placa "*Color/Graphics Adapter*". Para tanto, ele mantém uma tabela contendo a posição do cursor de cada página do vídeo.

Existe ainda o conceito da **página ativa** de vídeo, que é a que está sendo exibida. Na apresentação da página ativa, a posição do cursor é indicada através de um bloco (ou linha) piscante.

O BIOS suporta também a escrita e leitura de caracteres em modo gráfico. Para isso utiliza duas tabelas, uma em EPROM (no endereço 0FFA6EH) e outra em RAM (apontada pelo vetor de interrupção 1FH). A

primeira tabela corresponde aos caracteres de código 00H a 7FH, e a segunda aos de código 80H a 0FFH. Cada tabela contém 8 bytes por caractere, com os bits "ligados" indicando pontos "acesos" do caractere. Por exemplo, a letra "A" (código ASCII 41H) corresponde aos bytes 030H, 078H, 0CCH, 0CCH, 0FCH, 0CCH, 0CCH e 000H, contidos a partir do endereço 0FFC76 (0FFA6EH + 8*41H):

```
00110000
01111000
11001100
11001100
11111100
11001100
11001100
00000000
```

O BIOS inicia o vetor 1FH com 0:0, o que indica para o BIOS que não existe a segunda tabela.

3.3 FUNÇÕES DISPONÍVEIS

O acesso às funções do vídeo é feito através de INT 10H, com AH contendo o número da função desejada. As seguintes convenções são adotadas:

Página de vídeo:	0 "*Monochrome*" ou modo gráfico
	0 a 3 80x25, "*Color/Graphics*"
	0 a 7 40x25, "*Color/Graphics*"
Linha:	0 a 23
Coluna:	0 a 39 40x25 ou gráfico 320x200
	0 a 79 demais modos

O registrador AX é sempre alterado; os registradores BX, CX e DX só são alterados quando usados para retornar valor; o registrador BP é alterado por algumas funções; os demais registradores nunca são alterados.

AH = 0 **Seleciona modo**

Reinicia a interface de vídeo, programando-a para o modo selecionado em AL:

0 = alfa, 40x25, desabilita "burst" de cor

1 = alfa, 40x25, habilita "burst" de cor

2 = alfa, 80x25, desabilita "burst" de cor

3 = alfa, 80x25, habilita "burst" de cor

4 = graf, 320x200, habilita "burst" de cor

5 = graf, 320x200, desabilita "burst" de cor

6 = graf, 640x200

O "burst" de cor é o sinal de seleção das cores quando o monitor for ligado através da saída de vídeo composto. Normalmente utilizam-se monitores ligados na saída RGB, que não são afetados por esta programação.

No caso de uso de placa "*Monochrome*" esta função irá sempre programar o modo 80x25.

AH = 1 **Seleciona formato do cursor**

Esta função recebe em CH a linha inicial do cursor e em CL a linha final. Para a placa "*Color/Graphics*" as linhas vão de 0 a 7; para a "*Monochrome*" de 0 a 12.

Por exemplo, para dar ao cursor o formato de um bloco especificamos linha inicial 0 e linha final 7 ("*Color/Graphics*") ou 12 ("*Monochrome*").

AH = 2 **Posiciona o cursor**

Posiciona o cursor da página especificada em BH na linha e coluna especificadas em DH e DL.

AH = 3 **Lê posição do cursor**

Recebe em BH o número de uma página de vídeo e devolve em DH e DL a linha e coluna do cursor desta página.

AH = 4 **Lê posição do "*light pen*"**

Retorna AH = 0 se não detectou "*light pen*". Caso contrário retorna AH = 1, a linha e coluna em que o "*light pen*" foi detectado (DH e DL) e as coordenadas aproximadas do ponto (BX = coluna gráfica, 0 a 319 ou 639; CH = linha gráfica, 0 a 199). Se não detectou o "*light-pen*", BX, CX e DX tem seu conteúdo indefinido.

AH = 5 **Seleciona página ativa**

Seleciona a página passada em AL como nova página ativa. Esta função só deve ser usada nos modos alfanuméricos da placa "*Color/Graphics*".

AH = 6 **"Rola" página ativa para cima**

Esta função permite rolar para cima uma ou várias linhas de uma região da página ativa. Recebe os seguintes parâmetros:

AL = número de linhas a rolar
se for 0, toda a região é limpa
CH = linha superior da região a rolar
CL = coluna esquerda da região a rolar
DH = linha inferior da região a rolar
DL = coluna direita da região a rolar
BH = atributo a ser usado nas novas linhas

Altera o conteúdo de BP; o cursor não é movido.

Por exemplo, se chamarmos esta função com AL = 2, CH = 0, CL = 0, DH = 24, DL = 9 e BH = 70H, as primeiras 10 colunas das 25 linhas serão roladas duas linhas para cima (i.e., as primeiras 10 colunas da linha 2 irão para a linha 0, as da linha 3 para a linha 1, etc). As

primeiras 10 colunas das duas últimas linhas serão preenchidas com o caractere branco (20H) e o atributo 70H.

AH = 7 **"Rola" página ativa para baixo**

Esta função e análoga a anterior, recebendo os mesmos parâmetros, porem rola a região para baixo.

AH = 8 **Lê caractere e atributo**

Devolve em AL o caractere e em AH o atributo presentes na posição do cursor na página especificada em BH. O atributo só é fornecido nos modos alfanuméricos. No modo gráfico o BIOS procura reconhecer na posição do cursor um dos caracteres contidos nas tabelas de caracteres gráficos. Para isso considera apagados os pontos na cor 0 e acesos os pontos nas demais cores.

Altera BP, o cursor não é movido.

AH = 9 **Escreve caractere e atributo**

Recebe em AL um caractere, e em AH um atributo a ser escrito na página selecionada por BH, o número de vezes especificado em CX.

Nos modos gráficos, AH é a cor dos pontos "acesos" do caractere; os pontos "apagados" são sempre escritos na cor 0. Nestes modos, CX deve ser tal que todos os caracteres fiquem na mesma linha. Por exemplo, se o cursor está na coluna 10 e estamos no modo gráfico 320x200, CX pode ser no máximo 30. Se o bit mais significativo de AH for 1 é feito um "ou exclusivo" nos pontos acesos do caractere da cor já presente na tela com a cor especificada nos demais bits de AH. Por exemplo, se o caractere for um caractere com todos os pontos acesos, especificando AH = 83H iremos "complementar" todas as cores na posição do cursor:

cor 0 → cor 3, cor 1 → cor 2

cor 2 → cor 1, cor 3 → cor 0

Altera BP, o cursor não é movido.

AH = 10 **Escreve caractere**

Recebe em AL um caractere a ser escrito na página selecionada por BH, o número de vezes especificado em CX, sem alterar os atributos.

Nos modos gráficos, esta função é idêntica à anterior, devendo ser fornecida em AH a cor dos pontos acesos.

Altera BP, o cursor não é movido.

AH = 11 **Seleciona cor**

Esta função só tem sentido com a placa "*Color Graphics*".

Nos modos alfanuméricos, esta função permite escolher a cor da "borda" do vídeo e a intensidade das cores de fundo dos caracteres. Para tanto, deve ser chamada com BH igual a 0

e BL contendo nos bits 0 a 3 a cor da borda e no bit 4 a intensidade das cores de fundo (1 = intenso, 0 = normal).

Nos modos gráficos, temos duas sub-funções:

BH = 0 seleciona cor 0

BL contem a cor física (0 a 15)

BH = 1 seleciona conjunto de cores

BL = 0 Verde / Vermelho / Amarelo

BL = 1 Cian / Magneta / Branco

AH = 12 Escreve ponto

Esta função só está disponível nos modos gráficos e permite alterar a cor de um ponto da tela. Recebe em DX o número da linha gráfica (0 a 199), em CX o número da coluna gráfica (0 a 319 ou 639) e em AL a nova cor do ponto. Se o bit mais significativo de AL for 1 a cor do ponto é alterada para o "ou exclusivo" entre a sua cor atual e a cor especificada nos demais bits de AL.

AH = 13 Lê cor de ponto

Esta função devolve em AL a cor do ponto situado na linha gráfica especificada em DX e na coluna gráfica especificada em CX.

AH = 14 Escreve em modo "TTY"

Esta rotina escreve o caractere passado em AL no vídeo, identificando os seguintes códigos especiais:

07H -"*Bell*"

Soa um tom no alto-falante por cerca de 0,5 segundo. Não retorna do BIOS enquanto não acabar o tom.

08H - "*Backspace*"

Recua o cursor uma posição, sem mudar de linha.

0AH -"*Line Feed*"

Desce o cursor uma linha, sem mudar de coluna. Se o cursor estiver na ultima linha a tela é rolada para cima.

0DH -"*Carriage Return*"

Move o cursor para a coluna 0 da mesma linha.

Os demais caracteres terão o seu código escrito na tela (mesmo que seja um código de controle) e o cursor será avançado para a coluna seguinte. Se o cursor já estiver na última coluna de uma linha, ela passará para a coluna inicial da linha seguinte, rolando a linha se necessário.

Nos modos alfanuméricos deve-se especificar em BH a página ativa, devido a erros de programação em varias versões do BIOS da IBM. Nestes modos, as linhas em branco geradas pela rolagem da tela são preenchidas com o atributo da primeira posição da ultima linha da tela. Os caracteres são escritos sem alteração do atributo.

Nos modos gráficos deve-se especificar em BL a cor de escrita dos pontos acesos do caractere e colocar-se zero em BH. As linhas em branco geradas pela rolagem da tela são preenchidas com a cor 0.

Esta função altera BP.

AH = 15 Informa estado do vídeo

Retorna em AL o modo do vídeo, em AH o número de colunas (40 ou 80) e em BH a página ativa. O modo retornado está codificado da mesma forma que na função chamada por AH = 0, exceto que se a placa for a "*Monochrome*" retorna AL = 7.

3.4 EXEMPLO

O trecho abaixo limpa a tela e escreve uma mensagem na linha 12:

```
MENSAGEM DB      'Exemplo de acesso ao vídeo'
         DB      0                ; marca o fim
ESCREVE:
         MOV     AH,15
         INT     10H              ; AH = # de colunas
         PUSH    BX               ; BH = página ativa
         MOV     DL,AH
         DEC     DL               ; coluna final
         MOV     DH,24            ; linha final
         SUB     CX,CX            ; linha e coluna inicial
         MOV     BH,07H           ; atributo normal
         MOV     AX,0600H
         INT     10H              ; limpa tela
         POP     BX
         MOV     DH,12
         SUB     DL,DL
         MOV     AH,2
         INT     10H              ; posiciona cursor
         MOV     SI,OFFSET MENSAGEM
         CLD
MOSTRA:
         LODSB
         OR      AL,AL
         JZ      FIM
         MOV     AH,14
         INT     10H
         JMP     MOSTRA
FIM:
```

4 ACESSO AO DISQUETE

4.1 CARACTERÍSTICAS DO HARDWARE

A interface controladora de disquetes do PC-IBM utiliza um circuito uPD 765 da NEC (ou 8272 da Intel) para controlar até 4 unidades de disquete de 5 1/4 polegadas. A codificação dos dados é sempre feita em densidade dupla (MFM).

As unidades normalmente utilizadas trabalham com 40 trilhas de gravação e permitem operar tanto com discos de face simples como de face dupla. Estas unidades fornecem à interface indicação de disco protegido e da detecção do furo de índice, porém não informam a abertura da porta da unidade nem o tipo de disco presente (face simples ou dupla).

O sistema operacional DOS formata as trilhas com 8 (versão 1) ou 9 (versão 2 e posteriores) setores de 512 bytes, o que fornece as seguintes capacidades:

Disco face	Setores/trilhas	Capacidade
Simples	8	160 Kbytes
	9	180 Kbytes
Dupla	8	320 Kbytes
	9	360 Kbytes

A transferência de dados entre a interface e o processador é feita através do canal 2 de DMA ("*Direct Memory Access*" - acesso direto a memória). Uma característica do circuito de DMA utilizado no PC é o fato dele controlar apenas os 16 bits menos significativos de endereço, ficando os 4 bits mais significativos fixos durante uma transferência. Infelizmente, esta restrição não é contornada pelo BIOS, o que significa que uma transferência não pode cruzar fronteiras físicas de 64 Kbytes. Por exemplo, não é possível transferir diretamente, via BIOS, 4 setores (4 x 512 bytes = 2 Kbytes) a partir do endereço de memória 2FE00H, pois, na passagem do primeiro para o segundo setor, os 4 bits mais significativos mudam de 02H para 03H.

4.2 ESTRUTURA DE DADOS

A principal estrutura de dados é uma tabela de parâmetros para o controlador, apontada pelo vetor 1EH. O conteúdo desta tabela é o seguinte:

Byte	Bits	Conteúdo
0	7 a 4	"*step rate time*"
		2 a 32 ms em passos de 2 ms
	3 a 0	"*head unload time*"
		0 a 480 ms em passos de 32 ms
1	7 a 1	"*head load time*"
		4 a 512 ms em passos de 4 ms
	0	1 = usa DMA
		0 = não usa DMA
2	7 a 0	tempo de espera apos um acesso antes de desligar o motor da unidade. Especificado em unidades de 55 ms

3	7 a 0	bytes por setor (N): 0 = DTL 1 = 256 2 = 512 3 = 1024
4	7 a 0	setores por trilha
5	7 a 0	tamanho do "gap" de escrita
6	7 a 0	número de bytes por setor qdo N = 0 deve ser 0FFH quando N não for 0
7	7 a 0	tamanho do "gap" de formatação
8	7 a 0	byte de preenchimento na formatação
9	7 a 0	"*head settle time*", em ms
10	7 a 0	tempo de partida do motor, em oitavos de segundo

Embora esta tabela não seja normalmente manipulada pelo programador Assembler, cabe notar que o sistema DOS altera o vetor de forma a utilizar valores diferentes do padrão iniciado pelo BIOS:

byte	BIOS	DOS 1.1	DOS 2.0
0	0CFH	0DFH	0DFH
1	002H	002H	002H
2	025H	025H	025H
3	002H	002H	002H
4	008H	008H	009H
5	02AH	02AH	02AH
6	0FFH	0FFH	0FFH
7	050H	050H	050H
8	0F6H	0F6H	0F6H
9	019H	00BH	00FH
10	004H	002H	002H

4.3 FUNÇÕES DISPONÍVEIS

O acesso às funções de disquete é feito através de INT 13H, com AH contendo o número da função desejada.

Nestas funções as seguintes convenções são adotadas:

Unidade a acessar:	0 (A:) a 3 (D:)
Face a acessar:	0 ou 1
Trilha a acessar:	0 a 39
Setor a acessar:	1 a 9

Em todas as funções, o flag CY indica no retorno se ocorreu um erro (CY = 1) ou se a operação foi bem sucedida (CY = 0) e AH fornece o resultado da operação:

00H - operação bem sucedida

01H - função invalida

02H - "*addres mark*" não encontrado

03H - tentativa de escrita em disco protegido

04H - setor não encontrado

08H - falha no controlador de DMA

09H - buffer ultrapassou fronteira física de 64K

O BIOS verifica se o endereço inicial fornecido está no mesmo segmento físico de 64K que o endereço final (que é o endereço inicial mais o número de setores multiplicado por 512 bytes).

10H - setor com erro

20H - falha no controlador de disco

40H - trilha não encontrada

80H - unidade sem disco ("*time-out*")

Nas funções de leitura e verificação o BIOS não espera o tempo de partida do motor. Normalmente isto não ocasiona erro, o que permite um ganho substancial de tempo no acesso. Deve-se, entretanto, prever o caso em que a primeira leitura não é bem sucedida. Isto é feito realizando-se pelo menos três tentativas de leitura antes de considerar o erro como definitivo (as tentativas adicionais não necessitam ser feitas em caso de "*time-out*").

Nas funções de leitura, escrita e verificação, AL retorna o número de setores efetivamente acessados, exceto se ocorreu erro de unidade sem disco.

Em todas as funções apenas o registrador AX e os *flags* são alterados.

AH = 0 **Reinicia a interface**

Esta função reinicia a interface de disquete, ligando a pendência de recalibração (movimentação da cabeça de leitura/escrita para a trilha 0) em todas as unidades. Deve ser chamada sempre que ocorrer um erro em acesso ao disco. Conforme será detalhado no próximo capítulo, é conveniente chamar esta função com DL = 0 para garantir que a interface de disco rígido não será iniciada também.

AH = 1 **Informa resultado da ultima operação**

Retorna em AL o resultado da última operação realizada.

AH = 2 **Lê setores**

AL = número de setores a serem lidos, devem estar na mesma face e na mesma trilha

ES:BX = endereço da área onde serão colocados os dados

CH = trilha

CL = setor inicial

DH = face

DL = unidade

AH = 3 **Escreve setores**

AL = número de setores a serem escritos, devem estar na mesma face e na mesma trilha

ES:BX = endereço da área de onde serão obtidos os dados

CH = trilha

CL = setor inicial

DH = face

DL = unidade

AH = 4 Verifica setores

Esta função efetua a leitura dos dados sem transferi-los da interface para a memória. Desta forma, verifica-se se o setor pode ser acessado sem erro.

AL = número de setores a serem verificados, devem estar na mesma face e na mesma trilha

ES:BX = embora nenhuma transferência seja realizada, o BIOS testa (erroneamente) a presença de fronteira de 64K na região de transferência. Por este motivo deve-se colocar um valor válido para leitura nestes registradores

CH = trilha

CL = setor inicial

DH = face

DL = unidade

AH = 5 Formata trilha

AL = número de setores por trilha

ES:BX = endereço de uma área que contem os identificadores para os setores da trilha. Para cada setor esta área deve conter 4 bytes: o número da trilha (C), a face (H), o número do setor (R) e a indicação do número de bytes por setor (N). O BIOS testa a presença de fronteira de DMA considerando 512 bytes por setor, o que obriga cuidado na seleção deste endereço.

CH = trilha

CL = setor inicial

DH = face

DL = unidade

4.4 EXEMPLO

O trecho abaixo copia o conteúdo do disco na unidade A: para o disco na unidade B:, assumindo que o disco está formatado em 9 setores/trilha, dupla face e que o buffer utilizado não contem fronteira de 64K:

```
COPIA:
        MOV     CH,0            ; trilha
COPTRK:
        MOV     DH,0            ; face
COPHD:
        MOV     SI,3            ; # de tentativas
LE:
        MOV     DL,0
        MOV     CL,1
        MOV     BX,OFFSET BUFFER
        MOV     AH,2
        MOV     AL,9
        INT     13H             ; tenta ler a trilha da unidade A:
        JNC     LEU
        CMP     AH,80H          ; não conseguiu:
        JE      ERRO            ; desiste se "time-out"
        MOV     AH,0
        INT     13H             ; reinicia interface
        DEC     SI
        JNZ     LE              ; repete no máximo 3 vezes
        JMP     ERRO
LEU:
        MOV     DL,1
        MOV     AH,3
        MOV     AL,9
        INT     13H             ; escreve na unidade B:
        JC      ERRO
        INC     DH
        CMP     DH,2            ; repete para as 2 faces
        JB      COPHD           ; da trilha atual
        INC     CH
        CMP     CH,40
        JB      COPTRK          ; repete para as 40 trilhas

BUFFER  DB      9 * 512 DUP (?)
```

5 ACESSO AO DISCO RÍGIDO

5.1 CARACTERÍSTICAS DO HARDWARE

A interface controladora de discos rígidos do PC-IBM controla até 2 unidades de disco rígido (*winchester*).

As unidades normalmente utilizadas são de 5 1/4 polegadas, com 4 faces e 306 trilhas formatadas em 17 setores de 512 bytes, o que dá uma capacidade total de 4 x 306 x 17 x 512 bytes, ou seja, aproximadamente 10 Mbytes.

A transferência de dados entre a interface e o processador é feita através do canal 3 de DMA ("*Direct Memory Access*" - acesso direto à memória). Uma característica do circuito de DMA utilizado no PC é o fato dele controlar apenas os 16 bits menos significativos de endereço, ficando os 4 bits mais significativos fixos durante uma transferência. Infelizmente, esta restrição não é contornada pelo BIOS, o que significa que uma transferência não pode cruzar fronteiras físicas de 64 Kbytes. Por exemplo, não é possível transferir diretamente, via BIOS, 4 setores (4 x 512 bytes = 2 Kbytes) a partir do endereço de memória 2FE00H, pois na passagem do primeiro para o segundo setor os 4 bits mais significativos mudam de 02H para 03H.

Os setores são gravados em disco com o uso de ECC ("*Error Checking and Correction*") - detecção e correção de erros. Esta técnica permite a detecção de grande porcentagem dos erros de leitura/escrita e correção da maioria deles.

As unidades de disco rígido são normalmente testadas rigorosamente na fabricação, sendo levantadas quais as trilhas potencialmente defeituosas. A partir desta informação é feita a formatação física da unidade, com estas trilhas suspeitas sendo formatadas como trilhas "ruins" e as demais como "boas". A "formatação" realizada pelo sistema operacional consiste apenas na verificação das trilhas.

5.2 ESTRUTURA DE DADOS

As rotinas de BIOS que efetuam o tratamento da interface de disco rígido estão na própria interface. Durante a iniciação, o BIOS da placa de sistema reconhece a presença da interface e ativa a sua rotina de iniciação (como será detalhado no item 9.5). Esta rotina de iniciação intercepta os vetores de interrupção 13H (acesso a disco) e 19H (leitura do carregador do sistema operacional). O vetor de interrupção 40H é utilizado para salvar o endereço originalmente contido em INT 13H e passar adiante os comandos endereçados às unidades de disquete.

O BIOS utiliza também uma tabela de parâmetros para o controlador, apontada pelo vetor 41H. O conteúdo desta tabela é o seguinte:

Bytes	Conteúdo
0 e 1	Número de trilhas
2	Número de faces
3 e 4	Trilha inicial para redução da corrente de gravação
5 e 6	Trilha inicial para pré-compensação na gravação
7	Tamanho do ECC
8	Byte de controle
	bit 7 - se 1, desabilita repetição de acesso em caso de erro

bit 6 - se 1, desabilita repetição de acesso em caso de erro de ECC

9 Tempo máximo de espera em operações normais

10 Tempo máximo de espera em formatação

11 Tempo máximo em iniciação

12 a 15 Reservados

5.3 FUNÇÕES DISPONÍVEIS

O acesso às funções de disco rígido é feito através de INT 13H, com AH contendo o número da função desejada e DL o número da unidade com o bit mais significativo ligado. Caso o bit mais significativo de DL seja "0" é efetuado um acesso em unidade de disquete.

Nestas funções as seguintes convenções são adotadas:

unidade a acessar: 80H ou 81H

face a acessar: 0 a 3

trilha a acessar: 0 a 305

setor a acessar: 1 a 17

O número da trilha e o número do setor são codificados conjuntamente em um word:

```
trilha:       t9 t8 t7 t6 t5 t4 t3 t2 t1 t0
setor:        s7 s6 s5 s4 s3 s2 s1 s0
codificação:  t7 t6 t5 t4 t3 t2 t1 t0 (+ significativo)
              t9 t8 s5 s4 s3 s2 s1 s0 (- significativo)
```

Em todas as funções o *flag* CY indica no retorno se ocorreu um erro (CY = 1) ou se a operação foi bem sucedida (CY = 0), e AH fornece o resultado da operação:

00H - operação bem sucedida

01H - função invalida

02H - "*address mark*" não encontrado

04H - setor não encontrado

05H - falha em iniciação

07H - falha na programação dos parâmetros

09H - buffer ultrapassou fronteira física de 64K

O BIOS verifica se o endereço inicial fornecido está no mesmo segmento físico de 64K que o endereço final (que é o endereço inicial mais o número de setores multiplicado por 512 bytes).

0BH - tentativa de ler ou escrever em trilha formatada como ruim

10H - setor com erro

11H - setor com erro, corrigido através do ECC

20H - falha no controlador de disco

40H - trilha não encontrada

80H - ultrapassou tempo máximo de espera

B8H - erro indefinido

FFH - falha em leitura de status do controlador

Nas funções de leitura, escrita e verificação AL retorna o número de setores efetivamente acessados. Em todas as funções, exceto obtenção dos parâmetros, apenas o registrador AX e os *flags* são alterados.

AH = 0 Reinicia as interfaces de disco

Esta função reinicia as interfaces de disquete e de disco rígido. Para reiniciar somente a interface de disco rígido usar a função AH = 13.

AH = 1 Informa resultado da ultima operação

Retorna em AL o resultado da ultima operação realizada.

AH = 2 Lê setores

AL = número de setores a serem lidos, pode estar implícita mudança de face e trilha.

ES:BX = endereço da área onde serão colocados os dados

CX = trilha e setor iniciais

DH = face inicial

DL = unidade

AH = 3 Escreve setores

AL = número de setores a serem escritos, pode estar implícita mudança de face e trilha.

ES:BX = endereço da área de onde serão obtidos os dados

CX = trilha e setor iniciais

DH = face inicial

DL = unidade

AH = 4 Verifica setores

Esta função efetua a leitura dos dados sem transferi-los da interface para a memória. Desta forma, verifica-se se o setor pode ser acessado sem erro.

AL = número de setores a serem verificados, pode estar implícita mudança de face e trilha.

ES:BX = não utilizados nesta função, podem conter qualquer valor.

CX = trilha e setor iniciais

DH = face inicial

DL = unidade

AH = 5 **Formata trilha como "boa"**

AL = fator de entrelaçamento ("*interleave*"), normalmente 6

ES:BX = não utilizados nesta função, podem conter qualquer valor.

CX = trilha a formatar, setor deve ser 1

DH = face

DL = unidade

AH = 6 **Formata trilha como "ruim"**

AL = fator de entrelaçamento ("*interleave*"), normalmente 6

ES:BX = não utilizados nesta função, podem conter qualquer valor.

CX = trilha a formatar, setor deve ser 1

DH = face

DL = unidade

AH = 7 **Formata unidade a partir de uma trilha**

Esta função formata como "boa" uma trilha e todas as que a seguem, até o fim da unidade.

AL = fator de entrelaçamento ("*interleave*"), normalmente 6

ES:BX = não utilizados nesta função, podem conter qualquer valor.

CX = trilha inicial a formatar, setor deve ser 1

DH = face inicial a formatar

DL = unidade

AH = 8 **Informa parâmetros da unidade**

Retorna os parâmetros da unidade especificada em DL:

DL = número de unidades de disco rígido

DH = valor máximo para face

CX = valores máximos para trilha e setor

AH = 9 **Inicia parâmetros das unidades**

Esta função passa para a interface os parâmetros das unidades, apontados pelo vetor de INT 41H.

AH = 10 **Leitura "longa"**

Esta função é semelhante à chamada por AH = 2, recebendo os mesmos parâmetros, porém transfere 516 bytes por setor, sendo os 512 primeiros os dados e os 4 últimos referentes ao ECC.

AH = 11 Escrita "longa"

Esta função é semelhante à chamada por AH = 3, recebendo os mesmos parâmetros, porém transfere 516 bytes por setor, sendo os 512 primeiros os dados e os 4 últimos referentes ao ECC.

AH = 12 Posiciona cabeça em uma trilha

Posiciona a cabeça de leitura/escrita da unidade especificada em DL na trilha especificada em CX (setor deve ser 1).

AH = 13 Reinicia interface de disco rígido

Esta função serve como alternativa para a função chamada por AH = 0, reiniciando apenas a interface de disco rígido sem acessar a interface de disquetes.

AH = 14 Lê "*buffer*" da interface

A interface de disco rígido dispõe de uma memória local (um "*buffer*") com capacidade para um setor. Esta função lê para este "buffer" o setor especificado através dos seguintes parâmetros:

ES:BX = não utilizados nesta função, podem conter qualquer valor.

CX = trilha e setor

DH = face

DL = unidade

AH = 15 Escreve "*buffer*" da interface

Esta função grava o "*buffer*" da interface no setor especificado através dos seguintes parâmetros:

ES:BX = não utilizados nesta função, podem conter qualquer valor.

CX = trilha e setor

DH = face

DL = unidade

AH = 16 Testa se unidade pronta

Informa se a unidade em DL está pronta (CY = 0 e AH = 0) ou se há um posicionamento da cabeça em andamento (CY =1 e AH = 40H).

AH = 17 Recalibra unidade

Posiciona a cabeça de leitura/escrita na trilha 0.

5.4 EXEMPLO

O trecho abaixo lê os 3 últimos setores da face 2 da trilha 110H, os 17 setores da face 3 da mesma trilha e os 4 primeiros setores da face 0 da trilha 111H da primeira unidade de disco rígido. Assume que o *buffer* não contem fronteira de 64K.

```
        MOV     DH,2              ; face inicial
        MOV     DL,80H            ; unidade
        MOV     CX,110H           ; trilha inicial
        ROR     CH,1              ; move 2 bits + significativos
        ROR     CH,1              ; da trilha
        OR      CH,0FH            ; setor inicial
        XCHG    CH,CL             ; coloca nos regs corretos
        MOV     BX,OFFSET BUFFER
        MOV     AH,2
        MOV     AL,24             ; total de 24 setores
        INT     13H               ; le os setores
        JC      ERRO
```

6 ACESSO À IMPRESSORA

6.1 CARACTERÍSTICAS DO HARDWARE

A placa de interface com impressora utilizada no PC-IBM é uma placa de interface paralela, destinada a conexão de impressoras com interface padrão *Centronics*.

6.2 ESTRUTURA DE DADOS

O BIOS prevê até 3 interfaces, identificadas por um número de 0 a 2. Utiliza uma tabela de 3 words, localizada no endereço 00408H, para obter o endereço da interface, considerando o endereço 0000H como indicação da inexistência da interface.

As versões mais recentes do BIOS utilizam uma segunda tabela, no endereço 00478H para definir o tempo máximo de espera por interface pronta.

6.3 FUNÇÕES DISPONÍVEIS

O acesso às funções de impressora é feito através de INT 17H, com AH contendo o número da função desejada, e DX o número da interface (0 a 2); o conteúdo de AH é sempre alterado.

AH = 0 **imprime o caractere passado em AL**

Se a interface não ficar pronta dentro de um tempo máximo, retorna com AH = 1. Caso contrário, retorna em AH o estado da interface (ver abaixo)

AH = 1 **inicia a interface de impressora**

Retorna em AH o estado da interface (ver abaixo).

AH = 2 **informa estado da interface**

Retorna em AH o estado com a seguinte codificação:

bit	ativo se
7	interface pronta
6	recebeu sinal "acknowledge" da impressora
5	recebeu indicação de fim de papel
4	interface selecionada
3	recebeu sinal de erro

os bits 2, 1 e 0 não são usados.

6.4 EXEMPLO

A rotina abaixo envia para a impressora conectada à interface 1 a cadeia de caracteres apontada por SI e finalizada por um caractere '#':

```
IMPRIME  PROC    NEAR
         CLD                     ;para LODSB incrementar SI
         MOV     DX,1            ;número da interface
IMPCH:
         LODSB                   ;pega próximo caractere
         CMP     AL,'#'
         JE      FIM             ;retorna se chegou ao fim
         SUB     AH,AH
         INT     17H             ;imprime o caractere
         JMP     IMPCH           ;repete
FIM:
         RET
IMPRIME  ENDP
```

7 ACESSO ÀS INTERFACES SERIAIS

7.1 CARACTERÍSTICAS DO HARDWARE

As interfaces seriais tratadas pelo BIOS são baseadas na UART ("*Universal Assyncronous Receiver/Transmiter*" - Receptor/Transmissor Assíncrono Universal) 8250, permitindo a comunicação serial assíncrona com as seguintes características:

- 5, 6, 7 ou 8 bits de dados (o BIOS não suporta o uso de 5 ou 6 bits de dados)
- Paridade par, impar ou ausente
- 1, 1 1/2 ou 2 "stop bits" (o BIOS não suporta 1 1/2 "stop bit")
- Velocidade de comunicação selecionada por software, sendo a velocidade de recepção igual à de transmissão. O circuito utilizado pela IBM permite gerar as taxas de 50, 75, 110, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200 e 9600 bps com excelente precisão (o BIOS não suporta as taxas de 50, 75, 1800, 3600 e 7200).

A 8250 dispõe ainda de um sistema completo de interrupções, suportado pelo hardware, porem não utilizado pelo BIOS.

7.2 ESTRUTURA DE DADOS

O BIOS prevê até 4 interfaces, identificadas por um número de 0 a 3. Utiliza uma tabela de 4 words, localizada no endereço 00400H, para obter o endereço da interface, considerando o endereço 0000H como indicação da inexistência da interface.

As versões mais recentes do BIOS utilizam uma segunda tabela, no endereço 0047CH, para definir o tempo máximo de espera por interface pronta.

7.3 FUNÇÕES DISPONÍVEIS

O acesso às funções de tratamento das interfaces seriais é feito através de INT 14H, com AH contendo o número da função desejada, e DX o número da interface (0 a 3); o conteúdo de AH e sempre alterado.

AH = 0 **Inicia a interface serial**

Recebe em AL os parâmetros desejados:

bits 7,6,5 -	velocidade	bits 4,3 -	paridade
0 0 0	110	0 0	sem
0 0 1	150	0 1	impar
0 1 0	300	1 0	sem
0 1 1	600	1 1	par
1 0 0	1200		
1 0 1	2400		
1 1 0	4800		

1 1 1	9600		
bit 2 -	"*stop bits*"	bits 1,0 -	tamanho
0	1	1 0	7 bits
1	2	1 1	8 bits

Retorna em AX o estado da interface (ver abaixo)

AH = 1 **Transmite o caractere passado em AL**

Se a interface não ficar pronta dentro de um tempo máximo, retorna com o bit mais significativo de AH = 1. Os demais bits de AH correspondem ao estado da interface (ver abaixo). O conteúdo de AL é preservado.

AH = 2 **Recebe um caractere pela interface**

Se a interface não indicar a recepção de um caractere dentro de um tempo máximo, retorna com o bit mais significativo de AH = 1. Os bits 4, 3, 2 e 1 de AH correspondem ao estado da interface (ver abaixo). O caractere recebido é devolvido em AL.

AH =3 **Informa estado da interface**

Retorna em AH o estado da comunicação:

Bit	Ativo se
7	não conseguiu transmitir ou receber no tempo máximo
6	nenhum caractere em transmissão
5	pronto para aceitar caractere a transmitir
4	recebeu sinal de BREAK
3	recebeu caractere com erro de formato
2	recebeu caractere com erro de paridade
1	perdeu caractere na recepção, por ter chegado outro antes dele ter sido lido
0	recebeu dado

e em AL o estado dos sinais de modem:

Bit	Ativo se
7	sinal CD (detecção de portadora) presente
6	sinal RI ("*ring indicator*" - detecção de chamada) presente
5	sinal DSR ("*data set ready*" - modem pronto) presente
4	sinal CTS ("*clear to send*" - modem pronto para transmitir) presente
3	houve variação no sinal CD desde a última leitura de status
2	o sinal RI ficou ausente desde a última leitura de status
1	houve variação no sinal DSR desde a última leitura de status
0	houve variação no sinal CTS desde a última leitura de status

7.4 EXEMPLO

A rotina abaixo envia para a interface serial número 2 (COM3) a cadeia de caracteres apontada por SI e finalizada por um caractere '#':

```
ENVIA   PROC    NEAR

        CLD                     ;para LODSB incrementar SI
        MOV     DX,2            ;número da interface
TXCH:
        LODSB                   ;pega próximo caractere
        CMP     AL,'#'
        JE      FIM             ;retorna se chegou ao fim
        MOV     AH,1
        INT     14H             ;envia o caractere
        JMP     TXCH            ;repete
FIM:
        RET

ENVIA   ENDP
```

8 ACESSO AO RELÓGIO

8.1 CARACTERÍSTICAS DO HARDWARE

O hardware do PC para medição de tempo é bastante rudimentar, consistindo em um temporizador ("timer") que interrompe periodicamente o processador. O BIOS programa este timer para interromper 1193180/65536 vezes por segundo (aproximadamente 18.2 vezes) o que corresponde a um intervalo de 55 ms entre interrupções. Esta interrupção é utilizada para incrementar um contador, que é zerado quando a contagem atinge o correspondente a 24 horas.

Entrando em maiores detalhes, o PC dispõe de um circuito temporizador/ contador 8253 da Intel (ou equivalente), cujo canal 1 recebe uma frequência de 1193180 Hz e tem sua saída conectada ao sistema de interrupções (gerando a interrupção 8). A programação da taxa de interrupção e feita da seguinte forma:

1. Seleciona-se o canal 1, escrevendo-se o valor 36H no port de controle da 8253 (endereço 043H).
2. Escreve-se o byte menos significativo do contador no canal 1 da 8253 (endereço 041H)
3. Escreve-se o byte mais significativo do contador no canal 1 da 8253 (endereço 041H)

A taxa de interrupção será de 1193180/contador interrupções por segundo; uma contagem de zero é interpretada como 65536 e é o valor usado normalmente pelo BIOS.

8.2 ESTRUTURA DE DADOS

O tratamento do relógio do PC-IBM envolve duas interrupções, uma de hardware e outra de software:

1. O hardware gera um INT 8, que é normalmente tratado pelo BIOS;
2. No decorrer do tratamento do INT 8, o BIOS executa INT 1CH, de forma a fornecer uma base de tempo para programas aplicativos. Na sua iniciação, o BIOS aponta esta interrupção para uma instrução IRET.

O programa que reprogramar o timer deve tomar os seguintes cuidados:

1. Assumir o tratamento de INT 8;
2. Continuar chamando o tratamento original de INT 8 na taxa anterior, de forma a não interferir nas temporizações do BIOS;
3. Restaurar o vetor de INT 8 ao final de sua execução.

O programa que assumir o tratamento de INT 1CH deve continuar chamando o tratamento original e restaurar o vetor ao final de sua execução.

No capítulo 10, quando tratamos do acesso ao alto-falante, e apresentado um exemplo prático de reprogramação do timer.

8.3 FUNÇÕES DISPONÍVEIS

O acesso às funções de relógio é feito através de INT 1AH, com AH contendo o número da função desejada:

AH = 0 **Lê o contador de tempo**

Retorna em CX o word mais significativo e em DX o word menos significativo do contador de interrupções de relógio ("*tics*"). Retorna com AL diferente de zero se o contador foi zerado ("mudou de dia") desde a última leitura.

AH = 1 **Inicia o contador de tempo**

Recebe em CX o word mais significativo e em DX o word menos significativo do novo valor ("hora atual") do contador de interrupções.

Somente são alterados os registradores utilizados para retornar resultados.

8.4 EXEMPLO

A rotina apresentada a seguir devolve a hora atual no formato hora (CH), minuto (CL) e segundo (AL), assumindo que o contador de tempo foi acertado com a hora correta (o que é feito, por exemplo, pelo comando TIME do DOS):

```
TICS_HORA EQU 65520               ; 18.2 * 60 * 60
TICS_MIN  EQU 1092                ; 18.2 * 60
TICS_SEG  EQU 18

LE_HORA   PROC    NEAR

          MOV     AH,0
          INT     1AH
          MOV     AX,DX
          MOV     DX,CX           ; DXAX = tics
          MOV     BX,TICS_HORA
          DIV     BX
          MOV     CH,AL           ; CH = hora
          MOV     AX,DX
          SUB     DX,DX           ; DXAX = tics restantes
          MOV     BX,TICS_MIN
          DIV     BX
          MOV     CL,AL           ; CL = minutos
          MOV     AX,DX           ; AX = tics restantes
          MOV     BL,TICS_SEG
          DIV     BL              ; AL = segundos
          RET                     ; AH = tics restantes

LE_HORA   ENDP
```

Lembrar que o 8088 só têm dois tipos de divisão: divisão de DXAX por um word (valor de 1 a 65535) com quociente em AX e resto em DX, e divisão de AX por um byte (valor de 1 a 255) com quociente em AL e resto em AH.

9 OUTRAS FUNÇÕES DO BIOS

9.1 INFORMAÇÃO DA MEMÓRIA DISPONÍVEL

Esta função permite obter o tamanho da memória RAM do microcomputador. É chamada através de INT 12H, não tem parâmetros, e devolve em AX o tamanho da memória em Kbytes.

9.2 INFORMAÇÃO DOS DISPOSITIVOS DISPONÍVEIS

Esta função permite obter a configuração de dispositivos e interfaces do microcomputador. É chamada através de INT 11H, não tem parâmetros, e devolve o resultado em AX, conforme descrito abaixo:

Bits	Significado
15 e 14	Número de interfaces paralelas de impressora (0 a 3)
13	Indefinido
12	"1" se tem interface para "*joystick*"
11 a 9	Número de interfaces seriais (0 a 4)
8	Não usado
7 e 6	Número de unidades de disquete menos um, desde que o bit 0 esteja em "1"
5 e 4	Modo inicial do vídeo: 00 - invalido 01 - 40x25, placa "*Color/Graphics*" 10 - 80x25, placa "*Color/Graphics*" 11 - 80x25, placa "*Monochrome*"
3 e 2	Indefinidos
1	"1" se processador numérico 8087 configurado
0	"1" se interface de disquete configurada

No modelo original do PC-IBM as informações contidas nos bits 7 a 0 eram provenientes de chaves de configuração ("*dip-switchs*")

9.3 IMPRESSÃO DE TELA

Esta rotina é normalmente chamada pelo tratamento da interrupção de teclado, através de INT 5, e realiza a impressão do conteúdo da tela na impressora conectada à primeira interface paralela.

A leitura da tela é feita através de INT 10H e a impressão através de INT 17H. Existem alguns efeitos colaterais na rotina original que devem ser considerados:

1. Sempre pula de linha antes de começar a imprimir a tela e ao final de cada tela (ou seja, duas impressões consecutivas não saem contíguas);

2. Nos modos gráficos, imprime um branco no lugar dos caracteres não identificados; exceto neste caso, os códigos da tela são sempre jogados diretamente para a impressora, o que pode causar o envio de códigos de controle e códigos não ASCII (maiores de 07FH);

3. Nos modos gráficos, o registrador BP é alterado; se a função foi chamada pela interrupção de teclado, isto pode causar um mau funcionamento do programa aplicativo que está sendo executado;

4. Quando esta função é chamada pela interrupção de teclado, se o vídeo estava apagado (como ocorre durante uma operação de "rolagem") ele permanece apagado durante toda a impressão.

9.4 LEITURA DO CARREGADOR DO SISTEMA OPERACIONAL

Esta função lê o setor 1 da face 0 da trilha 0 (que normalmente contem o carregador do sistema operacional) para o endereço 07C00H e executa um desvio para 0:7C00H.

Na codificação de um carregador, deve-se considerar que DS, ES e CS conterão 0, e que a pilha (SS:SP) poderá estar em qualquer região da memória.

Esta função pode ser chamada através de INT 19H, sem parâmetros. Deve-se tomar o cuidado de posicionar a pilha fora da região de leitura.

Se existir uma interface de disco fixo, o vetor de INT 19H aponta para uma rotina que efetua a leitura da primeira unidade winchester (setor 1 da face 0 da trilha 0), caso não consiga ler do disquete. Para que um carregador escrito em winchester seja considerado válido ele deve conter 0AAH e 055H nos seus dois últimos bytes.

9.5 TESTES INICIAIS

Toda vez que o PC é ligado ou reiniciado (por exemplo, através de CTRL ALT DEL) é executada a rotina de iniciação e teste do BIOS, que segue os seguintes passos:

- Testa o processador 8088
- Testa a memória que contem o BIOS
- Testa o "*timer*" 8253
- Testa o controlador de DMA 8237
- Testa a memória RAM
- Inicia o controlador de interrupções 8259
- Inicia o vetor de interrupções
- Lê as chaves de configuração
- Testa a interface de vídeo
- Testa o teclado
- Determina a presença de gabinete de expansão
- Procura "expansões" do BIOS em placas de expansão
- Inicia a interface de disquete
- Verifica a presença de interfaces seriais e paralelas
- Desvia para a leitura do carregador do sistema

Para evitar que o teste de memória RAM seja realizado a cada reiniciação, o BIOS só o executa quando a variável no endereço 0472H não contém 1234H. Este valor é colocado pelo BIOS quando detecta a combinação CTRL ALT DEL.

Para facilitar a integração de placas de expansão, o BIOS permite a presença de "expansões" do BIOS. Estas expansões são procuradas entre os endereços 0C8000H e 0F4000H, de 2 em 2 Kbytes, e se caracterizam por terem a seguinte estrutura:

Byte 0: contém 55H

Byte 1: contém AAH

Byte 2: contém o tamanho do módulo, em blocos de 512 bytes

Byte 3: rotina de iniciação do módulo. O BIOS executa uma chamada intersegmento ("*call far*") para esta posição. Esta rotina pode interceptar vetores de interrupção, iniciar a placa de expansão, etc. Deve considerar o conteúdo dos registradores como indeterminado e não alterá-lo.

O BIOS efetua a soma de todos os bytes da expansão, módulo 256, considerando a expansão válida somente se esta soma for 0.

Para ativar a iniciação do BIOS deve-se executar um desvio longo ("*jump far*") para o endereço 0FFFFH:0.

10 UMA ROTINA PARA ACESSO AO ALTO-FALANTE

10.1 INTRODUÇÃO

Um dos pontos mais omissos do BIOS é o tratamento do alto-falante. A única função disponível é a geração de um tom de aproximadamente 1KHz, por cerca de 0,5 segundo, obtido pela "escrita" do código 07H através de INT 10H com AH = 14. Ainda por cima, o controle do processador permanece com o BIOS durante toda a duração do tom, "travando" momentaneamente o processamento.

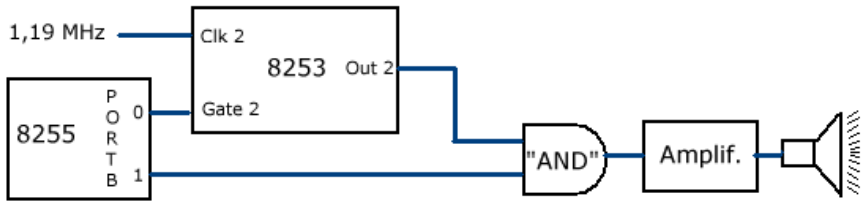
Neste capítulo será apresentada e discutida uma rotina com as seguintes características:

- Permite a geração de tons de freqüência a partir de 37Hz e duração de 10ms a quase 4 minutos.
- O controle do alto-falante é feito através da interrupção de tempo real, permitindo a execução de outros processamentos em paralelo.
- Permite o enfileiramento de tons para execução, dispensando o aguardo do término do tom atual.

10.2 CARACTERÍSTICAS DO HARDWARE

O coração do tratamento do alto-falante é o circuito temporizador/contador 8253 da Intel (ou equivalente), que recebe uma freqüência de 1193180 Hz e tem seu canal 1 conectado ao sistema de interrupções (gerando a interrupção 8), e o seu canal 2 conectado ao circuito do alto-falante.

O controle do alto-falante envolve ainda dois bits do port B da 8255: o bit 0 habilita o canal 2 do timer e o bit 1 permite a passagem para o alto-falante do tom gerado pelo timer (ver figura abaixo).



Normalmente, geram-se apenas tons puros a partir do timer; neste caso segue-se o seguinte esquema:

1. Seleciona-se o canal 2, escrevendo-se o valor B6H no port de controle da 8253 (endereço 043H).
2. Escreve-se o byte menos significativo do contador no canal 2 da 8253 (endereço 042H).
3. Escreve-se o byte mais significativo do contador no canal 2 da 8253 (endereço 042H).
4. Permite-se a passagem do tom ao alto-falante, programando-se para "1" os bits 0 e 1 do port B da 8255 (endereço 061H).

A frequência do tom será 2386360/contador Hz.

10.3 DESCRIÇÃO DA ROTINA

A rotina é composta de quatro partes:

1. Iniciação

   Nesta parte, é assumido o tratamento da interrupção de tempo real e a sua taxa é aumentada 16 vezes. Desta forma, obtém-se uma base de tempo de 3,4 ms, suficientemente precisa para a aplicação. Os ponteiros da fila são também iniciados.

2. Colocação de tons na fila

   Os valores fornecidos pelo usuário, em Hz e centésimos de segundos, são convertidos para contagens do timer. Aguarda-se então a existência de um lugar na fila e colocam-se nela os valores obtidos.

3. Tratamento da interrupção de tempo real

   A cada interrupção verifica-se se existe nota em execução e, em caso afirmativo, se chegou-se ao fim de sua duração. Se o alto-falante está (ou ficou) livre, uma nova nota é retirada da fila, sendo reprogramado o canal 2 do timer. A cada 16 interrupções chama-se o tratamento original de INT 8, de forma a não afetar as interrupções do BIOS.

4. Finalização

   Esta rotina restaura o vetor 8 e a programação do canal 1 do timer, encerrando a execução de tons no alto-falante.

10.4 LISTAGEM DA ROTINA

```
;
;        Rotina de Acesso ao Alto Falante
;
TIM_0    EQU     40H              ; Contador do canal 0
TIM_2    EQU     42H              ; Contador do canal 2
TIM_CTRL EQU     43H              ; Controle do timer
INT_CTRL EQU     20H              ; Controle da 8259
EOI      EQU     20H              ; Comando EOI p/ 8259
PORT_B   EQU     61H              ; Port B da 8255
SOM      EQU     3                ; Máscara para os bits de controle
                                  ; do alto-falante
TAM_FILA EQU     30               ; Tamanho da fila de notas

VETOR    SEGMENT AT 0
         ORG     8*4
INT8_OFF DW      ?                ; Vetor da interrupção do timer
INT8_SEG DW      ?
VETOR    ENDS

MUSICA   SEGMENT
         ASSUME  CS:MUSICA, DS:MUSICA

PRIMEIRO DW      ?                ; Ponteiro para o primeiro da fila
```

```
ULTIMO   DW      ?                ; Ponteiro para depois do último da fila
                                  ; i.e., onde colocar o próximo

INICIO   LABEL   BYTE
         DW      (TAM_FILA+1)*2 DUP (?)      ; Fila de notas
FIM      LABEL   BYTE

ENTR_FILA STRUCT                  ; Estrutura de uma entrada na fila de notas
FREQ     DW      ?
TEMPO    DW      ?
ENTR_FILA ENDS

DURACAO  DW      0                ; Duração da nota atual
CONT_INT DB      0                ; Contador de interrupções

INT_BIOS LABEL   DWORD
OFF_BIOS DW      ?                ; Endereço da rotina original de interrupção
SEG_BIOS DW      ?

;
; PREP_TIMER     Rotina para preparar a execução de notas
;                Assume a interrupção do timer e o reprograma
;
PREP_TIMER PROC  NEAR
         MOV     AX,OFFSET INICIO  ; Inicia a fila
         MOV     PRIMEIRO,AX
         MOV     ULTIMO,AX
         MOV     DX,DS
         SUB     AX,AX
         MOV     DS,AX
         ASSUME  DS:VETOR
         MOV     BX,OFFSET INT_TIMER
         MOV     CX,CS
         CLI
         XCHG    INT8_OFF,BX       ; Assume interrupções do timer
         XCHG    INT8_SEG,CX
         MOV     AL,36H
         OUT     TIM_CTRL,AL       ; Reprograma o timer, 16 vezes
         MOV     AL,0              ; mais rápido
         OUT     TIM_0,AL
         MOV     AL,10H
         OUT     TIM_0,AL
         MOV     DS,DX
         ASSUME  DS:MUSICA
         MOV     OFF_BIOS,BX       ; Salva endereço anterior
         MOV     SEG_BIOS,CX
         STI
         RET

PREP_TIMER ENDP
```

```
;
; POE_NOTA      Rotina que coloca uma nota na fila
;               <E>  BX – freqüência da nota (37 a 32767 Hz)
;                         0 indica pausa (silêncio)
;                    CX – duração da nota (1 a 65535 1/100 segundo)
;
POE_NOTA PROC   NEAR

         OR     BX,BX            ; Pausa?
         JZ     PN_1             ; Não:
         MOV    DX,36            ;   contagem = 2386360 / freq
         MOV    AX,27064
         ADD    AX,BX
         DEC    AX               ;   (arredondamento)
         DIV    BX
         MOV    BX,AX
PN_1:
         MOV    AX,1000          ; contagem = duração * 1000 / 343
         MUL    CX
         ADD    AX,342           ; (arredondamento)
         ADC    DX,0
         MOV    CX,343
         DIV    CX
         MOV    CX,AX
         MOV    DI,ULTIMO
         MOV    [DI].TEMPO,CX    ; Coloca no fim da fila
         MOV    [DI].FREQ,BX
         ADD    DI,4             ; Avança ponteiro
         CMP    DI,OFFSET FIM
         JB     PN_2             ; Se chegou ao fim,
         MOV    DI,OFFSET INICIO ;   circula para o início
PN_2:
         CMP    DI,PRIMEIRO
         JE     PN_2             ; Espera ter espaço na fila
         MOV    ULTIMO,DI        ; Atualiza o ponteiro
         RET
POE_NOTA ENDP

;
; INT_TIMER     Tratamento da interrupção do timer
;
INT_TIMER  PROC  NEAR

         PUSH   AX
         PUSH   BX
         PUSH   DS               ; Salva registradores
         MOV    AX,CS
         MOV    DS,AX
         CMP    DURACAO,0        ; Tem nota em execução?
         JE     IT_1             ; Sim:
         DEC    DURACAO          ;   decrementa duração da nota
         JNZ    IT_4             ;   acabou?
```

```
        IN      AL,PORT_B           ;   sim:
        AND     AL,NOT SOM
        OUT     PORT_B,AL           ;         desliga alto-falante
IT_1:                               ; Alto-falante está livre:
        MOV     BX,PRIMEIRO
        CMP     BX,ULTIMO           ;   tem nota na fila?
        JE      IT_4                ;   Sim:
        MOV     AX,[BX].TEMPO
        MOV     DURACAO,AX          ;         guarda duração
        MOV     AX,[BX].FREQ
        OR      AX,AX               ;         pausa?
        JZ      IT_2                ;         Não:
        OUT     TIM_2,AL            ;               reprograma o timer
        MOV     AL,AH
        OUT     TIM_2,AL
        IN      AL,PORT_B
        OR      AL,SOM
        OUT     PORT_B,AL           ;               liga alto-falante
IT_2:
        ADD     BX,4                ;         avança ponteiro p/ primeiro
        CMP     BX,OFFSET FIM       ;         se chegou ao fim
        JB      IT_3
        MOV     BX,OFFSET INICIO    ;               circula para o início
IT_3:
        MOV     PRIMEIRO,BX         ;         atualiza o ponteiro
IT_4:
        INC     CONT_INT            ; Incrementa o contador de ints
        TEST    CONT_INT,0FH        ; Contador é múltiplo de 16?
        JNZ     IT_5                ; Sim:
        POP     DS
        POP     BX
        POP     AX                  ;   restaura registradores
        JMP     CS:INT_BIOS         ;   desvia para a rotina original
IT_5:                               ; Não:
        MOV     AL,EOI
        OUT     INT_CTRL,AL         ;   sinaliza fim do tratamento
        POP     DS
        POP     BX
        POP     AX                  ;   restaura registradores
        IRET                        ;   retoma execução interrompida

INT_TIMER ENDP

;
; DEV_TIMER     Finaliza a execução das notas
;
DEV_TIMER PROC  NEAR
        MOV     AX,ULTIMO
DEV_1:
        CMP     AX,PRIMEIRO
        JNE     DEV_1               ; Espera a fila esvaziar
DEV_2:
```

```
        CMP     DURACAO,0
        JNE     DEV_2             ; Espera acabar a última nota
        MOV     DX,DS             ; Salva DS
        SUB     AX,AX
        MOV     DS,AX
        ASSUME  DS:VETOR
        MOV     BX,OFF_BIOS
        MOV     AX,SEG_BIOS
        CLI
        MOV     INT8_OFF,BX       ; Restaura vetor original
        MOV     INT8_SEG,AX
        MOV     AL,36H
        OUT     TIMER_CTRL,AL     ; Volta à programação original
        MOV     AL,0
        OUT     TIM_0,AL
        MOV     AL,0
        OUT     TIM_0,AL
        STI
        MOV     DS,DX             ; Restaura DS
        RET

DEV_TIMER ENDP

MUSICA   ENDS

        END
```

11 PROGRAMAS EXEMPLO

11.1 COMO ENTRAR COM ESTES PROGRAMAS EM UM PC

Os dois programas apresentados neste capítulo ilustram o uso das várias funções apresentadas ao longo deste livro, consistindo em programas completos, listados diretamente na impressora.

Para usar estes programas, você vai precisar de um PC (com pelo menos 128 Kbytes, uma unidade de disquete e interface de vídeo compatível com a "*Color Graphics Adapter*"), de um editor de programas, de um *Assembler*, do *Linker* da Microsoft (ou equivalente) e do utilitário EXE2BIN (ou equivalente).

Para obter o programa "Quadro de Avisos", siga os passos abaixo:

1. Entre com o programa-fonte (QUADRO.ASM), usando o editor de programas.
2. Assemble o programa-fonte, criando o objeto relocável (QUADRO.OBJ). Por exemplo, se você usar o MASM da Microsoft ou IBM:

   ```
   MASM QUADRO;
   ```

3. Gere o objeto executável (QUADRO.EXE), a partir do relocável:

   ```
   LINK QUADRO;
   ```

Para o segundo programa ("Duplicador de Disco"), deve-se também seguir estes três passo para obter os arquivos DUPL.ASM, DUPL.OBJ, DUPL.EXE. No último passo, o LINK indicará um "erro" ("*No stack segment*"), que deve ser ignorado. O programa DUPL.EXE não deve ser executado! Deve-se utilizar o utilitário EXE2BIN para gerar o objeto final DUPL.COM:

```
EXE2BIN DUPL DUPL.COM
```

11.2 QUADRO DE AVISOS

Este exemplo consiste, na realidade, em 3 programas:

1. Um programa executável sob DOS, que formata um disquete e grava nele os demais programas.
2. Um programa carregador que, acionado pelo BIOS, carrega o terceiro programa.
3. O programa "Quadro de Avisos" em si, que mostra continuamente um texto no vídeo. O texto é mostrado no modo gráfico média resolução, com caracteres de 16 pontos por 16 linhas (20 caracteres por linha), sendo "rolado" suavemente uma linha gráfica por vez.

Embora o terceiro programa caiba em um setor (o que permitiria dispensar o carregador), os outros dois foram escritos de foram a se adaptarem ao seu tamanho. Fica assim fácil ampliar o exemplo, acrescentando efeitos sonoros, aceitando comandos do teclado, etc.

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

                            ;
                            ; Quadro de Avisos
                            ;

= 0010                      VIDEO       EQU     10H             ; Int. p/ funcoes de video
= B800                      SEG_VIDEO   EQU     0B800H          ; Segto da memoria do video
= F000                      SEG_BIOS    EQU     0F000H          ; Segmento do BIOS
= FA6E                      TABCAR      EQU     0FA6EH          ; Tabela de carac. do BIOS
= 000F                      FRENTE      EQU     00001111B       ; Cor de frente do texto
= 0001                      FUNDO       EQU     1               ; Cor de fundo  do texto

= 0016                      TECLADO     EQU     16H             ; Int. funcoes de teclado
= 0013                      DISK_IO     EQU     13H             ; Int. funcoes de disco
= 0019                      REBOOT      EQU     19H             ; Int. p/ recarregar S.O.


                            ;===============================================================
                            ;
                            ; Segto que contem a pilha a ser usada pelo programa de gravacao
                            ;
                            ;===============================================================

0000                        PILHA       SEGMENT STACK
0000  0100 [                            DW      256 DUP (?)
            ????
                ]

0200                        PILHA       ENDS


                            ;===============================================================
                            ;
                            ; Segmento que contem o programa de exibicao do quadro de avisos,
                            ; que e' carregado e executado pelo BOOT.
                            ;
                            ;===============================================================

0000                        QUADRO      SEGMENT
                                        ASSUME  CS:QUADRO

0000                        INI_QUAD:
0000  E9 0088 R                         JMP     MOSQUAD


0003  2D 2D 2D 2D 2D 2D     TEXTO       DB      '--------------------'  ; Texto a exibir
      2D 2D 2D 2D 2D 2D
      2D 2D 2D 2D 2D 2D
      2D 2D
0017  50 43 20 41 73 73                 DB      'PC Assembler  Vol II'
      65 6D 62 6C 65 72
      20 20 56 6F 6C 20
      49 49
002B  20 20 20 44 61 6E                 DB      '   Daniel Quadros   '
      69 65 6C 20 51 75
      61 64 72 6F 73 20
      20 20
003F  2D 2D 2D 2D 2D 2D                 DB      '--------------------'
      2D 2D 2D 2D 2D 2D
      2D 2D 2D 2D 2D 2D
      2D 2D
0053  20 20 20 20 20 20                 DB      '                    '
      20 20 20 20 20 20
      20 20 20 20 20 20
      20 20
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

0067  50 72 6F 67 72 61              DB      'Programa Exemplo #1:'
      6D 61 20 45 78 65
      6D 70 6C 6F 20 23
      31 3A
007B  20 20 51 75 61 64              DB      '  Quadro de Avisos  '
      72 6F 20 64 65 20
      41 76 69 73 6F 73
      20 20
008F  20 20 20 20 20 20              DB      '                    '
      20 20 20 20 20 20
      20 20 20 20 20 20
      20 20
00A3  20 20 20 20 20 20              DB      '                    '
      20 20 20 20 20 20
      20 20 20 20 20 20
      20 20
00B7  FF                             DB      0FFH                ; marca de fim do texto


                                     ;
                                     ;  Rotina principal, mostra o texto no quadro de avisos
                                     ;

00B8                                 MOSQUAD:
00B8  BB 0151 R                              MOV     BX,OFFSET FIM_QUAD
00BB  81 C3 0200                             ADD     BX,512
00BF  8C C8                                  MOV     AX,CS               ; faz SS = CS, para
00C1  8E D0                                  MOV     SS,AX               ;   facilitar uso de BP
00C3  8B E3                                  MOV     SP,BX               ; pilha depois do programa
00C5  B8 0004                                MOV     AX,4                ; seleciona modo grafico
00C8  CD 10                                  INT     VIDEO               ;   320x200
00CA  B7 00                                  MOV     BH,0
00CC  B3 01                                  MOV     BL,FUNDO
00CE  B4 0B                                  MOV     AH,11
00D0  CD 10                                  INT     VIDEO               ; seleciona cor de fundo

00D2  B8 B800                                MOV     AX,SEG_VIDEO
00D5  8E C0                                  MOV     ES,AX
00D7  FC                                     CLD
00D8                                 QUAD1:
00D8  BD 0003 R                              MOV     BP,OFFSET TEXTO     ; ponteiro p/ texto
00DB                                 QUAD2:
00DB  2B DB                                  SUB     BX,BX               ; linha do caracter
00DD                                 QUAD3:
00DD  E8 00FA R                              CALL    ROLLUP              ; sobe tela 1 linha
00E0  E8 0115 R                              CALL    MOSLIN              ; mostra linha
00E3  E8 00FA R                              CALL    ROLLUP              ; sobre outra linha
00E6  E8 0115 R                              CALL    MOSLIN              ; mostra linha novamente
00E9  43                                     INC     BX
00EA  83 FB 08                               CMP     BX,8
00ED  72 EE                                  JB      QUAD3               ; repete para as 8 linhas
00EF  83 C5 14                               ADD     BP,20               ; avanca linha de texto
00F2  80 7E 00 FF                            CMP     BYTE PTR [BP],0FFH  ; final do texto ?
00F6  75 E3                                  JNE     QUAD2               ; nao: mostra nova linha
00F8  EB DE                                  JMP     QUAD1               ; sim: volta ao comeco


                                     ;
                                     ;  Sobe a tela de uma linha grafica
                                     ;

00FA                                 ROLLUP      PROC    NEAR

00FA  B8 B800                                MOV     AX,SEG_VIDEO
00FD  8E D8                                  MOV     DS,AX               ; DS aponta p/ video
00FF  BE 2000                                MOV     SI,2000H            ; linha 1
0102  BF 0000                                MOV     DI,0                ; linha 0
0105  BA 00C7                                MOV     DX,199              ; # de linhas a mover
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

0108                        RUP:
0108  B9 0028                        MOV     CX,40
0108  F3/ A5                  REP     MOVSW                   ; move linha
010D  87 F7                           XCHG    SI,DI           ; vamos copiar a linha
010F  83 EF 50                        SUB     DI,80           ;   seguinte para cima
0112  4A                              DEC     DX              ;   da anterior
0113  75 F3                           JNZ     RUP             ; repete p/ todas

0115                        ROLLUP    ENDP

                            ;
                            ;  Mostra uma linha grafica
                            ;
                            ;  (E)  BP - ponteiro para linha de texto
                            ;       BX - linha grafica a mostrar (0 a 7)
                            ;

0115                        MOSLIN    PROC    NEAR

0115  55                              PUSH    BP
0116  B8 F000                        MOV     AX,SEG_BIOS     ; acerta DS para acessar
0119  8E D8                           MOV     DS,AX           ;   tabela do BIOS
011B  BF 3EF0                         MOV     DI,2000H + 99*80 ; inicio da ultima linha
011E  B9 0014                         MOV     CX,20           ; numero de colunas
0121                        ML1:
0121  8A 46 00                        MOV     AL,[BP]         ; pega proximo caracter
0124  45                              INC     BP
0125  32 E4                           XOR     AH,AH
0127  8B F0                           MOV     SI,AX
0129  D1 E6                           SAL     SI,1
012B  D1 E6                           SAL     SI,1
012D  D1 E6                           SAL     SI,1
012F  B6 08                           MOV     DH,8
0131  8A 90 FA6E                     MOV     DL,TABCAR [BX+SI] ; pega imagem na tabela
0135                        ML2:
0135  D0 E0                           SAL     AL,1
0137  D0 E0                           SAL     AL,1
0139  D0 E0                           SAL     AL,1
013B  D0 E0                           SAL     AL,1            ; desloca 2 pontos
013D  D0 E2                           SAL     DL,1            ;   anteriores e coloca
013F  73 02                           JNC     ML3             ;   2 novos pontos
0141  0C 0F                           OR      AL,FRENTE
0143                        ML3:
0143  F6 C6 01                        TEST    DH,1
0146  74 01                           JZ      ML4             ; completou byte ?
0148  AA                              STOSB                   ; Sim: atualiza video
0149                        ML4:
0149  FE CE                           DEC     DH              ; repete p/ os 8 pares
014B  75 E8                           JNZ     ML2             ;   de pontos
014D  E2 D2                           LOOP    ML1             ; repete p/ os 20 caracs
014F  5D                              POP     BP
0150  C3                              RET

0151                        MOSLIN    ENDP

0151                        FIM_QUAD  LABEL   BYTE            ; p/ calcular # setores
= 0001                      TAM_QUADRO EQU    (FIM_QUAD - INI_QUAD + 511) / 512

0151                        QUADRO    ENDS
```

```
---- Objeto ----------    ------- Programa Fonte ----------------------------------------
                          ;
                          ; Definicao do endereco absoluto em que sera carregado
                          ; e executado o programa acima.
                          ;
0000                      QB          SEGMENT AT 50H

                                      ASSUME  CS:QB
0000                                  ORG     0

0000                      QENTR       LABEL   FAR

0000                      QB          ENDS
                          ;===============================================================
                          ;
                          ; Segmento que contem o BOOT, carregado e executado pelo BIOS.
                          ; Lembrar que a execucao e' iniciada em 0:7C00H
                          ;
                          ;===============================================================

0000                      BOOT        SEGMENT
                                      ASSUME  CS:BOOT, DS:BOOT


0000  B8 07C0                         MOV     AX,7C0H           ; DS - segmento de
0003  8E D8                           MOV     DS,AX             ;      execucao do BOOT
0005  B8 ---- R                       MOV     AX,QB             ; ES - segmento de
0008  8E C0                           MOV     ES,AX             ;      execucao de QUADRO

000A  BE 0055 R                       MOV     SI,OFFSET MEN_BOOT
000D  E8 009C R                       CALL    MOSMEN_BOOT       ; Avisa "Carregando"

0010  BE 0001                         MOV     SI,TAM_QUADRO     ; SI - # de setores a ler
0013  BA 0000                         MOV     DX,0              ; DH - face, DL - unidade
0016  B9 0001                         MOV     CX,1              ; CH - trilha, CL - setor
0019  BB 0000                         MOV     BX,0              ; ES:BX - buffer
001C                      LE_SET:
001C  BF 0003                         MOV     DI,3              ; DI - # de tentativas
001F  FE C1                           INC     CL                ; Avanca setor
0021  80 F9 08                        CMP     CL,8              ; Fim da face?
0024  76 0D                           JBE     RETRY             ; Sim:
0026  B1 01                           MOV     CL,1              ;   Volta ao setor 1
0028  FE C6                           INC     DH                ;   Avanca face
002A  80 FE 01                        CMP     DH,1              ;   Ja' leu as duas?
002D  76 04                           JBE     RETRY             ;   Sim:
002F  B6 00                           MOV     DH,0              ;     Volta a face 0
0031  FE C5                           INC     CH                ;     Avanca trilha
0033                      RETRY:
0033  B8 0201                         MOV     AX,0201H
0036  CD 13                           INT     DISK_IO           ; Tenta ler o setor
0038  73 0F                           JNC     LEU_SET           ; Conseguiu?
003A  4F                              DEC     DI                ; Nao:
003B  75 F6                           JNZ     RETRY             ;   Tenta ate' 3 vezes

003D  BE 0076 R                       MOV     SI,OFFSET ERR_BOOT
0040  E8 009C R                       CALL    MOSMEN_BOOT       ;   Da' erro
0043  B4 00                           MOV     AH,0
0045  CD 16                           INT     TECLADO           ;   Espera uma tecla
0047  CD 19                           INT     REBOOT            ;   Recarrega
0049                      LEU_SET:
0049  81 C3 0200                      ADD     BX,512            ; Avanca endereco
004D  4E                              DEC     SI
004E  75 CC                           JNZ     LE_SET            ; Repete ate' terminar
0050  EA 0000 ---- R                  JMP     QENTR             ; Desvia para o programa
                                                                ;   carregado
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

0055  0D 0A                 MEN_BOOT    DB      0Dh, 0Ah
0057  43 61 72 72 65 67                 DB      'Carregando QUADRO de AVISOS...
      61 6E 64 6F 20 51
      55 41 44 52 4F 20
      64 65 20 41 56 49
      53 4F 53 2E 2E 2E
0075  00                              DB      0

0076  0D 0A                 ERR_BOOT    DB      0Dh, 0Ah
0078  45 72 72 6F 20 6E               DB      'Erro na carga - digite uma tecla...'
      61 20 63 61 72 67
      61 20 2D 20 64 69
      67 69 74 65 20 75
      6D 61 20 74 65 63
      6C 61 2E 2E 2E
009B  00                              DB      0

                          ;
                          ; Mostra mensagem apontada por DS:SI e finalizada por 0.
                          ;

009C                      MOSMEN_BOOT PROC    NEAR

009C  FC                              CLD
009D                      MMB_1:
009D  AC                              LODSB                   ; Pega proximo caracter
009E  0A C0                           OR      AL,AL           ; Marca de fim?
00A0  74 08                           JZ      MMB_2           ; Sim: retorna
00A2  B7 00                           MOV     BH,0
00A4  B4 0E                           MOV     AH,14
00A6  CD 10                           INT     VIDEO           ; Escreve na pagina 0
00A8  EB F3                           JMP     MMB_1           ; Repete
00AA                      MMB_2:
00AA  C3                              RET

00AB                      MOSMEN_BOOT ENDP

00AB                      BOOT        ENDS

                          ;=============================================================
                          ;
                          ; Segmento que contem o programa de gravacao de BOOT e QUADRO,
                          ; executado pelo DOS
                          ;
                          ;=============================================================


0000                      FORMATA     SEGMENT
                                      ASSUME  CS:FORMATA, DS:FORMATA

= 0002                    NSET        EQU     TAM_QUADRO + 1
= 0001                    NTRK        EQU     (NSET +7) / 8


0000                      DOS         LABEL   DWORD           ; Retorno ao DOS
0000  0000                            DW      0
0002  ????                SEG_PSP     DW      ?

0004  43 6F 6C 6F 71 75   PEDE_DISCO  DB      'Coloque o disco a formatar na unidade A: '
      65 20 6F 20 64 69
      73 63 6F 20 61 20
      66 6F 72 6D 61 74
      61 72 20 6E 61 20
      75 6E 69 64 61 64
      65 20 41 3A 20
002D  65 20 64 69 67 69               DB      'e digite uma tecla.'
      74 65 20 75 6D 61
      20 74 65 63 6C 61
      2E
0040  00                              DB      0
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

0041  0D 0A               ERR_FMT     DB      0Dh, 0Ah
0043  45 72 72 6F 20 6E               DB      'Erro na formatacao !'
      61 20 66 6F 72 6D
      61 74 61 63 61 6F
      20 21
0057  00                              DB      0

0058  0D 0A               MEN_FIM     DB      0Dh, 0Ah
005A  44 69 73 63 6F 20               DB      'Disco formatado.'
      66 6F 72 6D 61 74
      61 64 6F 2E
006A  00                              DB      0

                         IDENT       STRUC                   ; Identificador para
0000  ??                 TRK         DB      ?               ;   formatacao
0001  ??                 FACE        DB      ?
0002  ??                 SET         DB      ?
0003  ??                 N           DB      ?
0004                     IDENT       ENDS

0068  00 00 01 02        BUF_FMT     IDENT   (0, 0, 1, 2)    ; Buffer para formatacao
006F  00 00 02 02                    IDENT   (0, 0, 2, 2)
0073  00 00 03 02                    IDENT   (0, 0, 3, 2)
0077  00 00 04 02                    IDENT   (0, 0, 4, 2)
007B  00 00 05 02                    IDENT   (0, 0, 5, 2)
007F  00 00 06 02                    IDENT   (0, 0, 6, 2)
0083  00 00 07 02                    IDENT   (0, 0, 7, 2)
0087  00 00 08 02                    IDENT   (0, 0, 8, 2)

0088                     INICIO:
0088  8C C8                          MOV     AX,CS
008D  8E D8                          MOV     DS,AX           ; Salva endereco de
008F  8C 06 0002 R                   MOV     SEG_PSP,ES      ;   retorno ao DOS
0093  8E C0                          MOV     ES,AX
0095  BE 0004 R                      MOV     SI,OFFSET PEDE_DISCO
0098  E8 0137 R                      CALL    MOSMEN_FMT      ; Pede colocacao do disco
0098                     INI_1:
0098  B4 01                          MOV     AH,1
009D  CD 16                          INT     TECLADO
009F  74 06                          JZ      INI_2
00A1  B4 00                          MOV     AH,0
00A3  CD 16                          INT     TECLADO
00A5  EB F4                          JMP     INI_1           ; Limpa a fila do teclado
00A7                     INI_2:
00A7  B4 00                          MOV     AH,0
00A9  CD 16                          INT     TECLADO         ; Espera digitar uma tecla

00AB  BE 0001                        MOV     SI,NTRK         ; SI - # de trilhas
00AE  BA 0000                        MOV     DX,0            ; DH - face, DL - unidade
00B1  B9 0001                        MOV     CX,1            ; CH - trilha, CL - setor
00B4  BB 0068 R                      MOV     BX,OFFSET BUF_FMT ; ES:BX - buffer
00B7                     FMT_1:
00B7  B8 0008                        MOV     AX,8            ; 8 setores por trilha
00BA  53                             PUSH    BX
00BB                     FMT_2:
00BB  88 2F                          MOV     [BX].TRK,CH     ; Coloca trilha
00BD  88 77 01                       MOV     [BX].FACE,DH    ;  e setor no buffer
00C0  83 C3 04                       ADD     BX,SIZE IDENT
00C3  48                             DEC     AX
00C4  75 F5                          JNZ     FMT_2
00C6  5B                             POP     BX
00C7  B8 0508                        MOV     AX,0508H
00CA  E8 0146 R                      CALL    DISCO           ; Formata a trilha
00CD  72 10                          JC      ERRO_FMT
00CF  4E                             DEC     SI              ; Formatou todas?
00D0  74 17                          JZ      GRV             ; Nao:
00D2  FE C6                          INC     DH              ;   Avanca face
00D4  80 FE 01                       CMP     DH,1            ;   Ja' formatou as duas?
00D7  76 DE                          JBE     FMT_1           ;   Sim:
```

```
---- Objeto ----------    ------- Programa Fonte ----------------------------------------

00D9  B6 00                       MOV     DH,0                ;     Volta a face 0
00DB  FE C5                       INC     CH                  ;     Avanca trilha
00DD  EB D8                       JMP     FMT_1
00DF                        ERRO_FMT:
00DF  BE 0041 R                   MOV     SI,OFFSET ERR_FMT
00E2  E8 0137 R                   CALL    MOSMEN_FMT          ; Avisa erro
00E5  FF 2E 0000 R                JMP     DOS                 ; Termina o programa
00E9                        GRV:
00E9  B8 ---- R                   MOV     AX,BOOT
00EC  8E C0                       MOV     ES,AX
00EE  BB 0000                     MOV     BX,0                ; ES:BX - setor de boot
00F1  BA 0000                     MOV     DX,0                ; face 0, unidade A:
00F4  B9 0001                     MOV     CX,1                ; trilha 0, setor 1
00F7  B8 0301                     MOV     AX,0301H
00FA  E8 0146 R                   CALL    DISCO               ; Grava BOOT
00FD  72 E0                       JC      ERRO_FMT
00FF  BE 0001                     MOV     SI,TAM_QUADRO       ; SI - N de setores
0102  B8 ---- R                   MOV     AX,QUADRO
0105  8E C0                       MOV     ES,AX               ; ES:BX - programa a ser
0107  BB 0000                     MOV     BX,0                ;         gravado

010A                        GRV_1:
010A  FE C1                       INC     CL                  ; Avanca setor
010C  80 F9 08                    CMP     CL,8                ; Fim da face?
010F  76 0D                       JBE     GRV_2               ; Sim:
0111  B1 01                       MOV     CL,1                ;   Volta ao setor 1
0113  FE C6                       INC     DH                  ;   Avanca face
0115  80 FE 01                    CMP     DH,1                ;   Ja' gravou nas duas ?
0118  76 04                       JBE     GRV_2               ;   Sim:
011A  B6 00                       MOV     DH,0                ;     Volta a face 0
011C  FE C5                       INC     CH                  ;     Avanca trilha

011E                        GRV_2:
011E  B8 0301                     MOV     AX,0301H
0121  E8 0146 R                   CALL    DISCO               ; Grava o setor
0124  72 B9                       JC      ERRO_FMT

0126  81 C3 0200                  ADD     BX,512              ; Avanca endereco
012A  4E                          DEC     SI
012B  75 DD                       JNZ     GRV_1               ; Repete ate' gravar tudo

012D  BE 0058 R                   MOV     SI,OFFSET MEN_FIM
0130  E8 0137 R                   CALL    MOSMEN_FMT          ; Avisa termino normal
0133  FF 2E 0000 R                JMP     DOS                 ; Volta ao DOS

                            ;
                            ; Mostra mensagem apontada por DS:SI e finalizada por 0.
                            ;

0137                        MOSMEN_FMT  PROC    NEAR


0137  FC                          CLD
0138                        MMF_1:
0138  AC                          LODSB                       ; Pega proximo caracter
0139  0A C0                       OR      AL,AL               ; Marca de fim?
013B  74 08                       JZ      MMF_2               ; Sim: retorna
013D  B7 00                       MOV     BH,0
013F  B4 0E                       MOV     AH,14
0141  CD 10                       INT     VIDEO               ; Escreve na pagina 0
0143  EB F3                       JMP     MMF_1               ; Repete
0145                        MMF_2:
0145  C3                          RET


0146                        MOSMEN_FMT  ENDP
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

                           ;
                           ; Executa acesso de escrita em disco, tratando erro de fronteira
                           ;
                           ; (E)   AX, CX, DX e ES:BX - parametros p/ INT 13H
                           ; (S)   CY e AH - status
                           ;
                           ; Altera:  Flags, AX, DI e BP
                           ;

0146                       DISCO       PROC    NEAR

0146  8B E8                            MOV     BP,AX           ; Salva op e # de setores
0148  CD 13                            INT     DISK_IO         ; Chama BIOS
014A  73 30                            JNC     DSC_1           ; Se nao deu erro retorna
014C  80 FC 09                         CMP     AH,9            ; Erro de fronteira ?
014F  F9                               STC                     ; (mantem CY = 1)
0150  75 2A                            JNE     DSC_1           ; Retorna se nao foi

0152  53                               PUSH    BX
0153  56                               PUSH    SI
0154  1E                               PUSH    DS
0155  06                               PUSH    ES
0156  51                               PUSH    CX              ; Salva registradores
0157  8C C7                            MOV     DI,ES
0159  B8 ---- R                        MOV     AX,ULTSEG       ; Vamos usar um buffer no
015C  05 001F                          ADD     AX,1FH          ;   fim do programa,
015F  25 FFE0                          AND     AX,0FFE0H       ;   calculado de forma
0162  8E DF                            MOV     DS,DI           ;   a nao ter fronteira
0164  8B F3                            MOV     SI,BX           ; DS:SI - buffer original
0166  8E C0                            MOV     ES,AX
0168  BF 0000                          MOV     DI,0            ; ES:DI - buffer auxiliar
016B  8B DF                            MOV     BX,DI           ; ES:BX - idem
016D  B9 0200                          MOV     CX,200H
0170  FC                               CLD
0171  F3/ A4                     REP   MOVSB                   ; Copia os dados
0173  59                               POP     CX
0174  8B C5                            MOV     AX,BP
0176  CD 13                            INT     DISK_IO         ; Re-executa a operacao
0178  07                               POP     ES
0179  1F                               POP     DS
017A  5E                               POP     SI
017B  5B                               POP     BX              ; Recupera registradores
017C                       DSC_1:
017C  C3                               RET

017D                       DISCO       ENDP

017D                       FORMATA     ENDS

0000                       ULTSEG      SEGMENT                 ; Define ultimo endereco
0000                       ULTSEG      ENDS                    ;   usado

                                       END     INICIO
```

11.3 DUPLICADOR DE DISCO

Este programa realiza a cópia de disco setor a setor, permitindo a duplicação de discos de 8 ou 9 setores por trilha, face simples ou dupla, sendo capaz de formatar o disco destino.

Para garantir um melhor desempenho, ele utiliza a técnica de entrelaçamento de setores ("*interleaving*"). Esta técnica consiste em ler os setores em uma ordem tal que, lido e processado um deles, a cabeça da unidade esteja posicionada próxima ao setor seguinte. Desta foram, minimiza-se o tempo de procura do setor na trilha ("*search*"). O programa utiliza também toda a memória livre disponível no PC, de forma a minimizar a espera pelo tempo de partida do motor.

Durante a duplicação é mantido atualizado um quadro no vídeo, indicando a operação em curso. Este quadro possui uma posição para cada setor do disco, e nele são colocadas as letras "L", "F" e "E" durante as operações de Leitura, Formatação e Escrita. No início de cada operação, o caractere é escrito em reverso; ao final bem sucedido da operação, ele é escrito em vídeo normal. Ao final da cópia, os setores copiados corretamente estarão indicados por ".", e os demais apresentarão, em vídeo reverso, a letra correspondente à operação em que ocorreu erro.

O programador interessado não terá dificuldade em acrescentar outras características, como a verificação após a escrita, cópia com uma única unidade de disco, etc.

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

                           ; Duplicador de Disco
                           ;

0000                       DUPL    SEGMENT
                                   ASSUME  CS:DUPL, DS:DUPL

0100                               ORG     100H

0100                       INICIO:
0100  E9 0548 R                     JMP     COMECO

= 0010                     VIDEO   EQU     10H     ; Int. de SW para funcoes de video
= 0016                     TECLADO EQU     16H     ;                               teclado
= 0013                     DISCO   EQU     13H     ;                               disco
= 0012                     TAM_MEM EQU     12H     ;                   obter tamanho da memoria
= 0017                     NORMAL  EQU     17H     ; Atributo normal
= 0071                     REVERSO EQU     71H     ;          reverso
= 001F                     INTENSO EQU     1FH     ;          intenso


0103  FF 03 00 00          TELA    DB      0FFH, 3, 0 ,0
0107  20 20 20 20 20 20           DB      '      ', 0DAH, 0FFH, 0, 40, 0C4H, 08FH, 0FFH, 2
      DA FF 00 28 C4 8F
      FF 02
0115  20 20 20 20 20 20           DB      '      '
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

0118  B3 30 FF 00 0F 20          DB      0B3H, '0', 0FFH, 0, 15, ' ', '1'
      31
0122  FF 00 0F 20 32 FF          DB      0FFH, 0, 15, ' ', '2', 0FFH
0128  00 07 20 B3 FF 02          DB      0, 7, ' ', 0B3H, 0FFH, 2
012E  20 20 20 20 20 20          DB      '      ', 0B3H, '0123456789ABCDEF0'
      B3 30 31 32 33 34
      35 36 37 38 39 41
      42 43 44 45 46 30
0146  31 32 33 34 35 36          DB      '123456789ABCDEF01234567'
      37 38 39 41 42 43
      44 45 46 30 31 32
      33 34 35 36 37
015D  B3 FF 02                   DB      0B3H, 0FFH, 2
0160  DA FF 00 05 C4 C5          DB      0DAH, 0FFH, 0, 5, 0C4H, 0C5H
0166  FF 00 28 C4 B4 FF          DB      0FFH, 0, 40, 0C4H, 0B4H, 0FFH, 2
      02
016D  B3 20 20 20 31 20          DB      0B3H, '   1 '
0173  B3 FF 00 28 20 B3          DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
017B  B3 20 46 20 32 20          DB      0B3H, ' F 2 '
0181  B3 FF 00 28 20 B3          DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
0189  B3 20 41 20 33 20          DB      0B3H, ' A 3 '
018F  B3 FF 00 28 20 B3          DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
0197  B3 20 43 20 34 20          DB      0B3H, ' C 4 ', 0B3H, 0FFH, 0, 40, ' ', 0B3H
      B3 FF 00 28 20 B3
01A3  20 20 20 FF 01 71          DB      '   ', 0FFH, 1, REVERSO, 'DUPLICADOR DE DISCO'
      44 55 50 4C 49 43
      41 44 4F 52 20 44
      45 20 44 49 53 43
      4F
01BC  FF 01 17 FF 02             DB      0FFH, 1, NORMAL, 0FFH, 2
01C1  B3 20 45 20 35 20          DB      0B3H, ' E 5 '
01C7  B3 FF 00 28 20 B3          DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
01CF  B3 20 20 20 36 20          DB      0B3H, '   6 ', 0B3H, 0FFH, 0, 40, ' ', 0B3H
      B3 FF 00 28 20 B3
01DB  20 20 20 53 65 74          DB      '   Setores por trilha. ', 0FFH, 2
      6F 72 65 73 20 70
      6F 72 20 74 72 69
      6C 68 61 3A 20 FF
      02
01F4  B3 20 30 20 37 20          DB      0B3H, ' 0 7 ', 0B3H, 0FFH, 0, 40, ' ', 0B3H
      B3 FF 00 28 20 B3
0200  20 20 20 46 61 63          DB      '   Face(s).....: ', 0FFH, 2
      65 28 73 29 2E 2E
      2E 2E 2E 2E 2E 2E
      2E 2E 2E 3A 20 FF
      02
0219  B3 20 20 20 38 20          DB      0B3H, '   8 '
021F  B3 FF 00 28 20 B3          DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
0227  B3 20 20 20 39 20          DB      0B3H, '   9 ', 0B3H, 0FFH, 0, 40, ' ', 0B3H
      B3 FF 00 28 20 B3
0233  20 20 20 46 6F 72          DB      '   Formata?.....: ',0FFH, 2
      6D 61 74 61 3F 2E
      2E 2E 2E 2E 2E 2E
      2E 2E 2E 3A 20 FF
      02
024C  C3 FF 00 05 C4 C5          DB      0C3H, 0FFH, 0, 5, 0C4H, 0C5H
0252  FF 00 2B C4 B4 FF          DB      0FFH, 0, 40, 0C4H, 0B4H, 0FFH, 2
      02
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

0259  B3 20 20 20 31 20           DB      0B3H, '   1 '
025F  B3 FF 00 28 20 B3           DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
0267  B3 20 46 20 32 20           DB      0B3H, ' F 2 '
026D  B3 FF 00 28 20 B3           DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
0275  B3 20 41 20 33 20           DB      0B3H, ' A 3 '
027B  B3 FF 00 28 20 B3           DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
0283  B3 20 43 20 34 20           DB      0B3H, ' C 4 '
0289  B3 FF 00 28 20 B3           DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
0291  B3 20 45 20 35 20           DB      0B3H, ' E 5 '
0297  B3 FF 00 28 20 B3           DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
029F  B3 20 20 20 36 20           DB      0B3H, '   6 '
02A5  B3 FF 00 28 20 B3           DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
02AD  B3 20 31 20 37 20           DB      0B3H, ' 1 7 '
02B3  B3 FF 00 28 20 B3           DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
02BB  B3 20 20 20 38 20           DB      0B3H, '   8 '
02C1  B3 FF 00 28 20 B3           DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
02C9  B3 20 20 20 39 20           DB      0B3H, '   9 '
02CF  B3 FF 00 28 20 B3           DB      0B3H, 0FFH, 0, 40, ' ', 0B3H, 0FFH, 2
      FF 02
02D7  C0 FF 00 05 C4 C1           DB      0C0H, 0FFH, 0, 5, 0C4H, 0C1H
02DD  FF 00 28 C4 D9 FF           DB      0FFH, 0, 40, 0C4H, 0D9H, 0FFH, 2
      02
02E4  00                          DB      0

                                  ; Tabela para mapeamento de setores

02E5  01 04 07 02 05 08           TABSET  DB      1, 4, 7, 2, 5, 8, 3, 6, 9
      03 06 09


                                  IDENT   STRUC           ; Identificador para formatacao

0000  ??                          TRK     DB      ?
0001  ??                          FACE    DB      ?
0002  ??                          SET     DB      ?
0003  ??                          N       DB      ?

0004                              IDENT   ENDS

02EE  ??                          NSET    DB      ?       ; numero de setores por trilha
02EF  ??                          NFACES  DB      ?       ; numero de faces
02F0  ??                          FMT     DB      ?       ; se 1, formatar disco destino
02F1  ??                          TRILHA  DB      ?       ; trilha atual
02F2  ????                        SEG_INI DW      ?       ; segto inicial para buffers de disco
02F4  ????                        NSET_M  DW      ?       ; numero de setores que cabem na memoria
02F6  ????                        NTRK    DW      ?       ; numero de trilhas que cabem na memoria
02F8  ????                        BUF_FMT DW      ?       ; segto inicial do buffer de formatacao
02FA  ??                          PAG_VID DB      ?       ; pagina ativa do video

02FB  FF 03 14 34 44 65           PERG    DB      0FFH, 3, 20, 52, 'Deseja outra copia? '
      73 65 6A 61 20 6F
      75 74 72 61 20 63
      6F 70 69 61 3F 20
0313  00                          DB      0
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

0314  FF 03 0F 34 43 6F      PEDE    DB      0FFH, 3, 15, 52, 'Coloque os discos:'
      6C 6F 71 75 65 20
      6F 73 20 64 69 73
      63 6F 73 3A
032A  FF 03 10 38 41 20              DB      0FFH, 3, 16, 56, 'A - Origem'
      2D 20 4F 72 69 67

      65 6D
0338  FF 03 11 38 42 20              DB      0FFH, 3, 17, 56, 'B - Destino'
      2D 20 44 65 73 74
      69 6E 6F
0347  00                             DB      0

0348  0100 [                        DW      256 DUP (?)     ; Area para a pilha
            ????
                  ]

0548                         PILHA   LABEL   WORD

                             ; Iniciacao e Programa Principal

0548                         COMECO:
0548  BC 0548 R                      MOV     SP,OFFSET PILHA     ; Pilha original
0548  B8 0891 R                      MOV     AX,OFFSET FIM_PGM   ;    sera' sobreposta
054E  05 000F                        ADD     AX,15
0551  B1 04                          MOV     CL,4
0553  D3 E8                          SHR     AX,CL               ; AX = primeiro paragrafo
0555  8C D8                          MOV     BX,DS               ;      livre apos o program
0557  03 C3                          ADD     AX,BX
0559  05 001F                        ADD     AX,1FH
055C  25 FFE0                        AND     AX,0FFE0H           ; Alinha em fronteira
055F  A3 02F8 R                      MOV     BUF_FMT,AX
0562  05 0020                        ADD     AX,20H              ; Reserva inicio
0565  A3 02F2 R                      MOV     SEG_INI,AX          ;   p/ formatacao

0568  CD 12                          INT     TAM_MEM             ; AX = tamanho da memoria
056A  B1 06                          MOV     CL,6                ;      em Kbytes
056C  D3 E0                          SHL     AX,CL
056E  2B 06 02F2 R                   SUB     AX,SEG_INI          ; AX = # de paragrafos livres
0572  B1 05                          MOV     CL,5
0574  D3 E8                          SHR     AX,CL               ; AX = # de setores que cabem
0576  3D 02D0                        CMP     AX,720              ;      na memoria
0579  72 03                          JB      INI_1               ; Vamos usar no maximo 720
057B  B8 02D0                        MOV     AX,720              ;      (9 * 40 * 2)
057E                         INI_1:
057E  A3 02F4 R                      MOV     NSET_M,AX

0581  B4 0F                          MOV     AH,15
0583  CD 10                          INT     VIDEO
0585  88 3E 02FA R                   MOV     PAG_VID,BH          ; Salva pagina ativa de video
0589                         NOV_COP:
0589  E8 0821 R                      CALL    MOS_TEL             ; Mostra tela inicial
058C  E8 05A7 R                      CALL    LE_PAR              ; Obtem parametros da copia
058F  72 08                          JC      FIM_COP             ; Fim se digitou ESC
0591  E8 0632 R                      CALL    COPY                ; Copia o disquete
0594  E8 07B7 R                      CALL    OUTRO               ; Deseja outra copia ?
0597  74 F0                          JZ      NOV_COP             ; Sim' repete

0599                         FIM_COP:
0599  B6 17                          MOV     DH,23               ; Nao:
059B  B2 00                          MOV     DL,0
059D  8A 3E 02FA R                   MOV     BH,PAG_VID
05A1  B4 02                          MOV     AH,2                ;   Coloca cursor no
05A3  CD 10                          INT     VIDEO               ;     fim da tela
05A5  CD 20                          INT     20H                 ;   Termina o programa
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

                            ;
                            ; Le parametros da Copia:  Setores por Trilha
                            ;                          Numero de Faces
                            ;                          Formatar antes de gravar
                            ; (S) CY = 1 se digitou ESC
                            ;

05A7                        LE_PAR  PROC    NEAR

05A7  8A 3E 02FA R                  MOV     BH,PAG_VID
05AB  B6 09                         MOV     DH,9
05AD  B2 47                         MOV     DL,71
05AF  B4 02                         MOV     AH,2
05B1  CD 10                         INT     VIDEO               ; Posiciona cursor
05B3                        LEP_1:  .
05B3  B4 00                         MOV     AH,0
05B5  CD 16                         INT     TECLADO             ; Le setores/trilha
05B7  3C 1B                         CMP     AL,1BH
05B9  74 75                         JE      LEP_7               ; Aborta se ESC
05BB  B4 08                         MOV     AH,8
05BD  3C 38                         CMP     AL,'8'
05BF  74 06                         JE      LEP_2
05C1  FE C4                         INC     AH
05C3  3C 39                         CMP     AL,'9'
05C5  75 EC                         JNE     LEP_1
05C7                        LEP_2:
05C7  88 26 02EE R                  MOV     NSET,AH
05CB  B9 0001                       MOV     CX,1
05CE  B3 1F                         MOV     BL,INTENSO
05D0  B4 09                         MOV     AH,9
05D2  CD 10                         INT     VIDEO               ; Mostra escolha no video
05D4  B6 0A                         MOV     DH,10
05D6  B2 47                         MOV     DL,71
05D8  B4 02                         MOV     AH,2
05DA  CD 10                         INT     VIDEO               ; Posiciona cursor
05DC                        LEP_3:
05DC  B4 00                         MOV     AH,0
05DE  CD 16                         INT     TECLADO             ; Le numero de faces
05E0  3C 1B                         CMP     AL,1BH
05E2  74 4C                         JE      LEP_7               ; Aborta se ESC
05E4  B4 01                         MOV     AH,1
05E6  3C 31                         CMP     AL,'1'
05E8  74 06                         JE      LEP_4
05EA  FE C4                         INC     AH
05EC  3C 32                         CMP     AL,'2'
05EE  75 EC                         JNE     LEP_3
05F0                        LEP_4:
05F0  88 26 02EF R                  MOV     NFACES,AH
05F4  B9 0001                       MOV     CX,1
05F7  B3 1F                         MOV     BL,INTENSO
05F9  B4 09                         MOV     AH,9
05FB  CD 10                         INT     VIDEO               ; Mostra escolha no video
05FD  B6 0C                         MOV     DH,12
05FF  B2 47                         MOV     DL,71
0601  B4 02                         MOV     AH,2
0603  CD 10                         INT     VIDEO               ; Posiciona cursor
0605                        LEP_5:
0605  B4 00                         MOV     AH,0
0607  CD 16                         INT     TECLADO             ; Le opcao de formatacao
0609  3C 1B                         CMP     AL,1BH
060B  74 23                         JE      LEP_7               ; Aborta se ESC
060D  B4 01                         MOV     AH,1
060F  3C 53                         CMP     AL,'S'
0611  74 0E                         JE      LEP_6
0613  3C 73                         CMP     AL,'s'
0615  74 0A                         JE      LEP_6
0617  FE CC                         DEC     AH
0619  3C 4E                         CMP     AL,'N'
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

0618  74 04                         JE      LEP_6
061D  3C 6E                         CMP     AL,'n'
061F  75 E4                         JNE     LEP_5
0621                          LEP_6:
0621  88 26 02F0 R                  MOV     FMT,AH
0625  B9 0001                       MOV     CX,1
0628  B3 1F                         MOV     BL,INTENSO
062A  B4 09                         MOV     AH,9
062C  CD 10                         INT     VIDEO           ; Mostra escolha no video
062E  F8                            CLC
062F  C3                            RET
0630                          LEP_7:
0630  F9                           STC                     ; Indica tecla ESC
0631  C3                            RET

0632                          LE_PAR  ENDP

                              ;
                              ;  Controla a copia do disco
                              ;

0632                          COPY    PROC    NEAR

0632  8A 1E 02EE R                  MOV     BL,NSET
0636  80 3E 02EF R 01               CMP     NFACES,1
063B  74 02                         JE      COP_1           ; BL = setores por cilindro
063D  D0 E3                         SHL     BL,1
063F                          COP_1:
063F  A1 02F4 R                     MOV     AX,NSET_M
0642  F6 F3                         DIV     BL              ; AL = numero de cilindros
0644  3C 28                         CMP     AL,40           ;      que cabem na memoria
0646  72 02                         JB      COP_2
0648  B0 28                         MOV     AL,40           ;      (no maximo 40)
064A                          COP_2:
064A  2A E4                         SUB     AH,AH
064C  A3 02F6 R                     MOV     NTRK,AX
064F  E8 0665 R                     CALL    PEDE_DISCO      ; Pede a colocacao dos discos
0652  C6 06 02F1 R 00               MOV     TRILHA,0
0657                          COP_3:
0657  E8 067C R                     CALL    READ            ; Le cilindros para memoria
065A  E8 06DD R                     CALL    WRITE           ; Grava/Formata os cilindros
065D  80 3E 02F1 R 28               CMP     TRILHA,40
0662  72 F3                         JB      COP_3           ; Repete para os 40 cilindros
0664  C3                            RET

0665                          COPY    ENDP

                              ;
                              ; Pede a colocacao dos discos e aguarda a digitacao de uma tecla
                              ;

0665                          PEDE_DISCO      PROC    NEAR

0665  BE 0314 R                      MOV     SI,OFFSET PEDE
0668  E8 0835 R                      CALL    MOSTRA          ; Primeira linha da mensagem
0668                          PD_1:
066B  B4 01                         MOV     AH,1
066D  CD 16                         INT     TECLADO
066F  74 06                         JZ      PD_2
0671  B4 00                         MOV     AH,0
0673  CD 16                         INT     TECLADO
0675  EB F4                         JMP     PD_1            ; Esvazia fila do teclado
0677                          PD_2:
0677  B4 00                         MOV     AH,0
0679  CD 16                         INT     TECLADO         ; Espera digitar uma tecla
067B  C3                            RET
```

```
---- Objeto ----------    ------- Programa Fonte ----------------------------------------

                          ;
                          ; Le NTRK trilhas a partir de TRILHA
                          ;
067C                      READ    PROC    NEAR

067C  8E 06 02F2 R                MOV     ES,SEG_INI          ; ES:0 - buffer
0680  8B 3E 02F6 R                MOV     DI,NTRK             ; DI - # maximo de trilhas
0684  BA 0000                     MOV     DX,0                ; DH - face, DL - unidade A:
0687  8A 2E 02F1 R                MOV     CH,TRILHA           ; CH - trilha
068B  2B DB                       SUB     BX,BX               ; BX - setor "logico"
068D                      RD_1:
068D  8A 8F 02E5 R                MOV     CL,TABSET [BX]      ; CL - setor "fisico"
0691  53                          PUSH    BX
0692  B0 4C                       MOV     AL,'L'
0694  B4 71                       MOV     AH,REVERSO
0696  E8 07C8 R                   CALL    MARCA               ; Marca leitura no quadro
0699  BE 0003                     MOV     SI,3                ; SI - contador de tentativas
069C                      RD_2:
069C  2B DB                       SUB     BX,BX
069E  B8 0201                     MOV     AX,0201H
06A1  CD 13                       INT     DISCO               ; Tenta ler o setor
06A3  73 09                       JNC     RD_3                ; Se nao conseguiu
06A5  B4 00                       MOV     AH,0
06A7  CD 13                       INT     DISCO               ;   Reinicia interface
06A9  4E                          DEC     SI                  ;   Repete ate' 3 vezes
06AA  75 F0                       JNZ     RD_2
06AC  EB 07                       JMP     SHORT RD_4
06AE                      RD_3:
06AE  B0 4C                       MOV     AL,'L'              ; Conseguiu:
06B0  B4 17                       MOV     AH,NORMAL           ;   Marca no quadro
06B2  E8 07C8 R                   CALL    MARCA
06B5                      RD_4:
06B5  8C C0                       MOV     AX,ES
06B7  05 0020                     ADD     AX,20H
06BA  8E C0                       MOV     ES,AX               ; Avanca endereco do buffer
06BC  5B                          POP     BX
06BD  43                          INC     BX                  ; Avanca setor "logico
06BE  3A 1E 02EE R                CMP     BL,NSET             ; Fim da trilha ?
06C2  72 C9                       JB      RD_1                ; Sim:
06C4  2B DB                       SUB     BX,BX               ;   Volta ao primeiro setor
06C6  80 3E 02EF R 01             CMP     NFACES,1            ;   Dupla face ?
06CB  74 05                       JE      RD_5                ;   Sim:
06CD  80 F6 01                    XOR     DH,1                ;     Muda de face
06D0  75 BB                       JNZ     RD_1                ;     Se face 1 vai ler
06D2                      RD_5:
06D2  FE C5                       INC     CH                  ;   Avanca trilha
06D4  80 FD 28                    CMP     CH,40
06D7  74 03                       JE      RD_6                ;   Fim se chegou a trilha 39
06D9  4F                          DEC     DI                  ;     ou encheu a memoria
06DA  75 B1                       JNZ     RD_1
06DC                      RD_6:
06DC  C3                          RET

06DD                      READ    ENDP

                          ;
                          ; Escreve/Formata NTRK trilhas a partir de TRILHA
                          ;

06DD                      WRITE   PROC    NEAR

06DD  8E 06 02F2 R                MOV     ES,SEG_INI          ; ES:0 - buffer
06E1  8B 3E 02F6 R                MOV     DI,NTRK             ; DI - # maximo de trilhas
06E5  BA 0001                     MOV     DX,1                ; DH - face, DL - unidade B:
06E8  8A 2E 02F1 R                MOV     CH,TRILHA           ; CH - trilha
```

```
---- Objeto ------------  ------- Programa Fonte ----------------------------------------
06EC                      WR_0:
06EC  80 3E 02F0 R 00             CMP     FMT,0
06F1  74 03                       JE      WR_1              ; Se requisitou,
06F3  E8 0752 R                   CALL    FORMAT            ;    Formata trilha
06F6                      WR_1:
06F6  2B DB                       SUB     BX,BX             ; BX = setor "logico"
06F8                      WR_2:
06F8  8A 8F 02E5 R                MOV     CL,TABSET [BX]    ; CL = setor "fisico"
06FC  53                          PUSH    BX
06FD  E8 07F7 R                   CALL    LE_MARCA
0700  80 FC 71                    CMP     AH,REVERSO        ; Nao escreve se deu erro na
0703  74 23                       JE      WR_5              ;   leitura ou formatacao
0705  B0 45                       MOV     AL,'E'
0707  B4 71                       MOV     AH,REVERSO
0709  E8 07C8 R                   CALL    MARCA             ; Marca escrita no quadro
070C  BE 0003                     MOV     SI,3              ; SI - contador de tentativas
070F                      WR_3:
070F  2B D8                       SUB     BX,BX
0711  B8 0301                     MOV     AX,0301H
0714  CD 13                       INT     DISCO             ; Tenta gravar o setor
0716  73 09                       JNC     WR_4              ; Se nao conseguiu
0718  B4 00                       MOV     AH,0
071A  CD 13                       INT     DISCO             ;   Reinicia interface
071C  4E                          DEC     SI                ;   Repete ate' 3 vezes
071D  75 F0                       JNZ     WR_3
071F  EB 07                       JMP     SHORT WR_5
0721                      WR_4:
0721  B0 2E                       MOV     AL,'.'            ; Conseguiu:
0723  B4 17                       MOV     AH,NORMAL         ;   Marca no quadro
0725  E8 07C8 R                   CALL    MARCA
0728                      WR_5:
0728  8C C0                       MOV     AX,ES
072A  05 0020                     ADD     AX,20H
072D  8E C0                       MOV     ES,AX             ; Avanca endereco do buffer
072F  5B                          POP     BX
0730  43                          INC     BX                ; Avanca setor "logico"
0731  3A 1E 02EE R                CMP     BL,NSET           ; Fim da trilha ?
0735  72 C1                       JB      WR_2              ; Sim:
0737  80 3E 02EF R 01             CMP     NFACES,1          ;   Dupla face ?
073C  74 05                       JE      WR_6              ;   Sim:
073E  80 F6 01                    XOR     DH,1              ;     Muda de face
0741  75 A9                       JNZ     WR_0              ;     Se face 1 vai gravar
0743                      WR_6:
0743  FE C5                       INC     CH                ;   Avanca trilha
0745  80 FD 28                    CMP     CH,40
0748  74 03                       JE      WR_7              ;   Fim se chegou a trilha 39
074A  4F                          DEC     DI                ;     ou encheu a memoria
074B  75 9F                       JNZ     WR_0
074D                      WR_7:
074D  88 2E 02F1 R                MOV     TRILHA,CH         ; Atualiza trilha
0751  C3                          RET
0752                      WRITE   ENDP
                          ;
                          ; Formata a face DH da trilha CH do disco na unidade DL
                          ; Altera: Flags, AX, BX, CL, SI
                          ;
0752                      FORMAT  PROC    NEAR

0752  B1 01                       MOV     CL,1
0754                      FMT_1:
0754  E8 07F7 R                   CALL    LE_MARCA
0757  80 FC 71                    CMP     AH,REVERSO        ; Nao formata se deu erro
075A  74 5A                       JE      FMT_5             ;   na leitura
075C  FE C1                       INC     CL
075E  3A 0E 02EE R                CMP     CL,NSET
0762  76 F0                       JBE     FMT_1
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

0764  B1 01                        MOV     CL,1
0766                        FMT_2:
0766  B0 46                        MOV     AL,'F'
0768  B4 71                        MOV     AH,REVERSO
076A  E8 07C8 R                     CALL    MARCA               ; Marca formatacao no quadro
076D  FE C1                         INC     CL
076F  3A 0E 02EE R                  CMP     CL,NSET
0773  76 F1                         JBE     FMT_2
0775  8C C6                         MOV     SI,ES
0777  8E 06 02F8 R                  MOV     ES,BUF_FMT          ; ES:BX - buffer de
077B  2B DB                         SUB     BX,BX               ;         formatacao
077D  B8 0001                       MOV     AX,1
0780                        FMT_3:
0780  26: 88 2F                     MOV     ES:[BX].TRK,CH      ; Coloca trilha,
0783  26: 88 77 01                  MOV     ES:[BX].FACE,DH     ;  face,
0787  26: 88 47 02                  MOV     ES:[BX].SET,AL      ;  setor
078B  26: C6 47 03 02               MOV     ES:[BX].N,2         ;  e tamanho.
0790  83 C3 04                      ADD     BX,SIZE IDENT
0793  40                            INC     AX
0794  3A 06 02EE R                  CMP     AL,NSET
0798  76 E6                         JBE     FMT_3

079A  2B DB                         SUB     BX,BX
079C  B8 0508                       MOV     AX,0508H            ; Formata a trilha
079F  CD 13                         INT     DISCO
07A1  8E C6                         MOV     ES,SI
07A3  72 11                         JC      FMT_5

07A5  B1 01                         MOV     CL,1
07A7                        FMT_4:
07A7  B0 46                         MOV     AL,'F'
07A9  B4 17                         MOV     AH,NORMAL
07AB  E8 07C8 R                     CALL    MARCA               ; Atualiza o quadro
07AE  FE C1                         INC     CL
07B0  3A 0E 02EE R                  CMP     CL,NSET
07B4  76 F1                         JBE     FMT_4
07B6                        FMT_5:
07B6  C3                            RET

07B7                        FORMAT  ENDP
                            ;
                            ; Pergunta se deseja copiar outro disco
                            ; (S) Z = 1 se respondeu SIM
                            ;         0 se respondeu NAO
                            ;
07B7                        OUTRO   PROC    NEAR

07B7  BE 02F8 R                     MOV     SI,OFFSET PERG
07BA  E8 0835 R                     CALL    MOSTRA
07BD  B4 00                         MOV     AH,0
07BF  CD 16                         INT     TECLADO
07C1  3C 53                         CMP     AL,'S'
07C3  74 02                         JZ      OUT_1
07C5  3C 73                         CMP     AL,'s'
07C7                        OUT_1:
07C7  C3                            RET

07C8                        OUTRO   ENDP
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

                            ;
                            ; Coloca o caracter em AL, com atributo em AH na posicao do quadro
                            ; correspondente a trilha CH, face DH e setor CL
                            ;
                            ; Altera: Flags, AX, BX

07C8                        MARCA   PROC    NEAR

07C8  52                            PUSH    DX
07C9  51                            PUSH    CX
07CA  50                            PUSH    AX                  ; Salva regs

07CB  8A D5                         MOV     DL,CH
07CD  80 C2 07                      ADD     DL,7                ; Coluna = Trilha + 7
07D0  8A C6                         MOV     AL,DH
07D2  8A F1                         MOV     DH,CL
07D4  80 C6 03                      ADD     DH,3                ; Linha = Setor + 3
07D7  0A C0                         OR      AL,AL

07D9  74 03                         JZ      MRC_1               ; Se segunda face
07DB  80 C6 0A                      ADD     DH,10               ;   Linha = Linha + 10
07DE                        MRC_1:
07DE  8A 3E 02FA R                  MOV     BH,PAG_VID
07E2  B4 02                         MOV     AH,2
07E4  CD 10                         INT     VIDEO               ; Posiciona o cursor
07E6  58                            POP     AX
07E7  B9 0001                       MOV     CX,1
07EA  8A 3E 02FA R                  MOV     BH,PAG_VID
07EE  8A DC                         MOV     BL,AH
07F0  B4 09                         MOV     AH,9
07F2  CD 10                         INT     VIDEO               ; Escreve o caracter
07F4  59                            POP     CX
07F5  5A                            POP     DX                  ; Recupera regs
07F6  C3                            RET

07F7                        MARCA   ENDP
                            ;
                            ; Informa em AL e AH o caracter e atributo na posicao do quadro
                            ; correspondente a trilha CH, face DH e setor CL
                            ;
                            ; Altera: Flags, AX, BX

07F7                        LE_MARCA        PROC    NEAR

07F7  52                            PUSH    DX
07F8  51                            PUSH    CX
07F9  50                            PUSH    AX                  ; Salva regs

07FA  8A D5                         MOV     DL,CH
07FC  80 C2 07                      ADD     DL,7                ; Coluna = Trilha + 7
07FF  8A C6                         MOV     AL,DH
0801  8A F1                         MOV     DH,CL
0803  80 C6 03                      ADD     DH,3                ; Linha = Setor + 3
0806  0A C0                         OR      AL,AL
0808  74 03                         JZ      LMRC_1              ; Se segunda face
080A  80 C6 0A                      ADD     DH,10               ;   Linha = Linha + 10
080D                        LMRC_1:
080D  8A 3E 02FA R                  MOV     BH,PAG_VID
0811  B4 02                         MOV     AH,2
0813  CD 10                         INT     VIDEO               ; Posiciona o cursor
0815  58                            POP     AX

0816  8A 3E 02FA R                  MOV     BH,PAG_VID
081A  B4 08                         MOV     AH,8
081C  CD 10                         INT     VIDEO               ; Le caracter/atributo
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

081E  59                            POP     CX
081F  5A                            POP     DX                  ; Recupera regs
0820  C3                            RET

0821                        LE_MARCA        ENDP

                            ;
                            ; Mostra a tela inicial
                            ;
0821                        MOS_TEL PROC    NEAR

0821  BA 184F                       MOV     DX,184FH
0824  B9 0000                       MOV     CX,0
0827  B7 17                         MOV     BH,NORMAL
0829  B8 0600                       MOV     AX,0600H
082C  CD 10                         INT     VIDEO               ; limpa a tela
082E  BE 0103 R                     MOV     SI,OFFSET TELA
0831  E8 0835 R                     CALL    MOSTRA              ; monta tela inicial
0834  C3                           RET
0835                        MOS_TEL ENDP

                            ;
                            ; MOSTRA - Escreve na tela, conforme tabela apontada por DS:SI
                            ;
                            ;       A tabela pode conter os seguintes codigos:
                            ;       00h     - indica fim da tabela
                            ;       0ffh 0  - posicoes seguintes contem numero de repeticoes
                            ;                 e o caracter a ser repetido
                            ;       0ffh 1  - posicao seguinte contem o novo atributo
                            ;       0ffh 2  - muda de linha
                            ;       0ffh 3  - posicoes seguintes contem nova linha e coluna
                            ;                 para o cursor
                            ;       demais  - escreve na tela caracter correspondente
                            ;
0835                        MOSTRA  PROC    NEAR

0835  8A 3E 02FA R                  MOV     BH,PAG_VID          ; BH - Pagina Ativa
0839  B4 03                         MOV     AH,3
083B  CD 10                         INT     VIDEO               ; DH - Linha, DL - Coluna
083D  B3 17                         MOV     BL,NORMAL           ; BL - Atributo
083F  FC                            CLD
0840                        MOS_0:
0840  80 FA 4F                      CMP     DL,79               ; Acerta coluna para o caso
0843  76 07                         JBE     MOS_1               ;   de mudanca de linha
0845  80 EA 50                      SUB     DL,80
0848  FE C6                         INC     DH
084A  EB F4                         JMP     MOS_0
084C                        MOS_1:
084C  B4 02                         MOV     AH,2
084E  CD 10                         INT     VIDEO               ; Posiciona o cursor
0850  AC                            LODSB                       ; Pega proximo item da tabela
0851  0A C0                         OR      AL,AL
0853  74 3B                         JZ      MOS_6
0855  3C FF                         CMP     AL,0FFH
0857  74 0B                         JE      MOS_2
0859  B9 0001                       MOV     CX,1                ; Caracter comum
085C  B4 09                         MOV     AH,9
085E  CD 10                         INT     VIDEO
0860  FE C2                         INC     DL
0862  EB DC                         JMP     MOS_0
0864                        MOS_2:
0864  AC                            LODSB                       ; Codigo especial
0865  3C 01                         CMP     AL,1
0867  74 10                         JZ      MOS_3
0869  73 13                         JNC     MOS_4
```

```
---- Objeto ------------    ------- Programa Fonte ----------------------------------------

0868  AC                          LODSB                   ; Escrita com repeticao
086C  8A C8                       MOV     CL,AL
086E  02 D0                       ADD     DL,AL
0870  AC                          LODSB
0871  2A ED                       SUB     CH,CH
0873  B4 09                       MOV     AH,9
0875  CD 10                       INT     VIDEO
0877  EB C7                       JMP     MOS_0
0879                        MOS_3:
0879  AC                          LODSB                   ; Altera atributo
087A  8A D8                       MOV     BL,AL
087C  EB CE                       JMP     MOS_1
087E                        MOS_4:
087E  3C 03                       CMP     AL,3
0880  74 06                       JE      MOS_5
0882  2A D2                       SUB     DL,DL           ; Passa p/ linha seguinte
0884  FE C6                       INC     DH
0886  EB C4                       JMP     MOS_1
0888                        MOS_5:
0888  AC                          LODSB
0889  8A F0                       MOV     DH,AL
088B  AC                          LODSB
088C  8A D0                       MOV     DL,AL           ; Altera linha e coluna
088E  EB BC                       JMP     MOS_1
0890                        MOS_6:
0890  C3                          RET

0891                        MOSTRA  ENDP


0891                        FIM_PGM LABEL   BYTE            ; Ultimo endereco usado


0891                        DUPL    ENDS

                                    END     INICIO
```

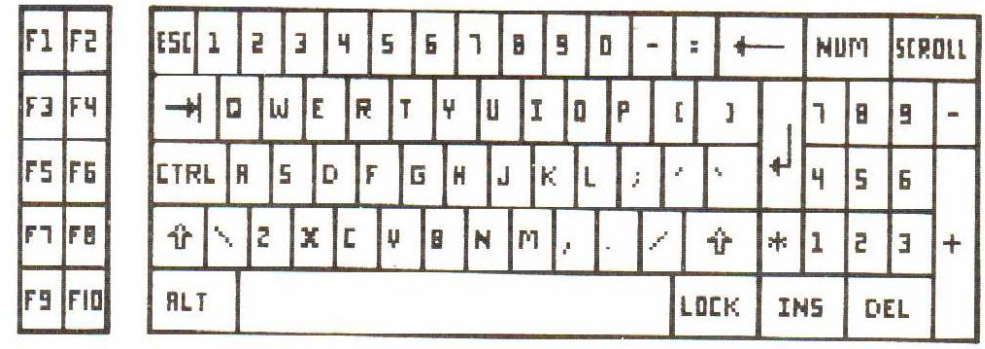
APÊNDICE A – MAPA DA MEMÓRIA

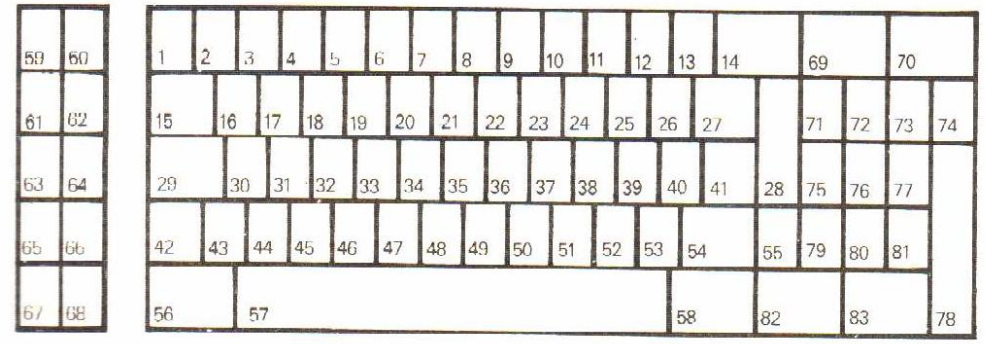
00000H – 9FFFFH	Memória Ram	
	00000H – 003FFH	Vetores de Interrupção
	00400H – 004FFH	Variáveis do BIOS
A0000H – BFFFFH	Reservado para interfaces de vídeo	
	B0000H – B3FFFH	Memória de vídeo da placa "*Monochrome*"
	B0000H – B3FFFH	Memória de vídeo da placa "*Color Graphics*"
C0000H – EFFFFH	Reservado para memória ROM em placa de expansão	
	C8000H – C8FFFH	Memória ROM com o BIOS da placa controladora de winchester
F0000H – FFFFFH	Reservado para memória ROM na placa de sistema	
	F6000H – FDFFFH	ROM BASIC
	FE000H –FF8FFFH	ROM BIOS

Obs.: O mapa fornecido acima é o sugerido pela IBM, no manual de referência técnica do PC-XT. Alguns fabricantes de placa de expansão e micros compatíveis tomam algumas liberdades em relação a este mapa; em particular é muito popular a colocação de RAM na região de A0000H a AFFFFH.

APÊNDICE B – CÓDIGOS DO TECLADO

As figuras abaixo indicam os "*scan codes*" (códigos de varredura) gerados pelo teclado padrão do PC-IBM.





A tabela a seguir indica os códigos obtidos do BIOS para as diversas combinações de teclas.

Scan code	Normal		Shift		Control		Alt
1	ESC	001B	ESC	001B	ESC	001B	–
2	1	0031	!	002		–	7800
3	2	0032	@	0040	NUL	0300	7900
4	3	0033	#	0023		–	7A00
5	4	0034	$	0024		–	7B00
6	5	0035	%	0025		–	7C00
7	6	0036	^	005E	RS	001E	7D00
8	7	0037	&	0026		–	7E00
9	8	0038	*	002A		–	7F00
10	9	0039	(	0028		–	8000
11	0	0030	)	0029		–	8100
12	-	002D	_	005F	US	001F	8200
13	=	003D	+	002B		–	8300
14	BS	0008	BS	0008	DEL	007F	–
15	HT	0009	HT	0009		–	–
16	q	0071	Q	0051	DC1	0011	1000
17	w	0077	W	0057	ETB	0017	1100
18	e	0065	E	0045	ENQ	0005	1200
19	r	0072	R	0052	DC2	0012	1300
20	t	0074	T	0054	DC4	0014	1400
21	y	0079	Y	0059	EM	0019	1500

Scan code	Normal		Shift		Control		Alt
22	u	0075	U	0055	NAK	0015	1600
23	i	0069	I	0049	HT	0009	1700
24	o	006F	O	004F	SI	000F	1800
25	p	0070	P	0050	DLE	0010	1900
26	[	005B	{	007B	ESC	001B	-
27	]	005D	}	007D	GS	001D	-
28	CR	000D	CR	000D	LF	001A	-
29	-		-		-		-
30	a	0061	A	0041	SOH	0001	1E00
31	s	0073	S	0053	DC3	0013	1F00
32	d	0064	D	0044	EOT	0004	2000
33	f	0066	F	0046	ACK	0006	2100
34	g	0067	G	0047	BEL	0007	2200
35	h	0068	H	0048	BS	0008	2300
36	j	006A	J	004A	LF	000A	2400
37	k	006B	K	004B	VT	000B	2500
38	l	006C	L	004C	FF	000C	2600
39	;	003B	:	003A	-		-
40	‘	0027	“	0022	-		-
41	`	0060	~	007E	-		-
42	-		-		-		-
43	\	005C	|	007C	FS	001C	-
44	z	007A	Z	005A	SUB	001A	2C00
45	x	0078	X	0058	CAN	0018	2D00
46	c	0063	C	0043	ETX	0003	2E00
47	v	0076	V	0056	SYN	0016	2F00
48	b	0062	B	0042	STX	0002	3000
49	n	006E	N	004E	SO	000E	3100
50	m	006D	M	004D	CR	000D	3200
51	,	002C	<	003C	-		-
52	.	002E	>	003E	-		-
53	/	002F	?	003F	-		-
54	-		-		-		-
55	*	002A	(Obs 1)		7200		-
56	-		-		-		-
57	espaço	0020	espaço	0020	espaço	0020	0020
58	-		-		-		-
59	3B00		5400		5E00		6800
60	3C00		5500		5F00		6900
61	3D00		5600		6000		6A00
62	3E00		5700		6100		6B00
63	3F00		5800		6200		6C00
64	4000		5900		6300		6D00
65	4100		5A00		6400		6E00
66	4200		5B00		6500		6F00
67	4300		5C00		6600		7000
68	4400		5D00		6700		7100
69	-		-		(Obs 2)		-
70	-		-		(Obs 3)		-

Obs 1: "Shift PrtScr" ("*hardcopy*") dispara a impressão da tela.

Obs 2: "Control Num Lock" ("*break*") limpa a fila do teclado, gera INT 1CH e coloca o código 0000 na fila.

Obs 3: "Control Scroll Lock" ("*pause*") suspende o processamento, aguardando a digitação de uma tecla qualquer.

As teclas abaixo são as teclas do bloco numérico; a tecla Num Lock troca os seus significados nos modos Normal e Shift. A única combinação válida com Alt é "Control Alt Del", que reinicia o PC.

"scan code"	Normal	Shift	Control
71	4700	7 0037	7700
72	4800	8 0038	-
73	4900	9 0039	8400
74	- 002D	-	-
75	4B00	4 0034	7300
76	-	5 0035	-
77	4D00	6 0036	7400
78	+ 002B	-	-
79	4F00	1 0031	7500
80	5000	2 0032	-
81	5100	3 0033	7600
82	5200	0 0030	-
83	2300	. 002E	-

APÊNDICE C – SÍMBOLOS DO VÍDEO

