

### 3 - ACESSO DIRETO AO FDC

Muito embora não seja recomendado, é possível o acesso direto ao disco, sobrepujando ao sistema operacional. Nesse caso, é necessário saber qual tipo de interface está instalada, pois diferentes interfaces usam diferentes meios para o acesso. Nessa seção está descrito somente o acesso ao FDC (floppy disk control), de forma bastante resumida.

O FDC é acessado escrevendo e lendo dados em seus registradores internos. Esses registradores são os seguintes:

- 1 - Registrador de status
- 2 - Registrador de comando
- 3 - Registrador de trilha
- 4 - Registrador de setor
- 5 - Registrador de dados
- 6 - Registrador de drive, lado do disquete e motor do drive
- 7 - Registrador de IRQ, ocupado e requisição de dados

Alguns FDC's têm diferenças entre esses registradores. Isso será descrito detalhadamente mais adiante.

#### 3.1 - COMANDOS DO FDC

Existem 4 categorias de comandos que podem ser executados pelos controladores de disco, conforme descrito abaixo:

Tipo	Comando	b7	b6	b5	b4	b3	b2	b1	b0
I	Restore	0	0	0	0	h	V	r1	r0
I	Seek	0	0	0	1	h	V	r1	r0
I	Step	0	0	1	T	h	V	r1	r0
I	Step-In	0	1	0	T	h	V	r1	r0
I	Step-Out	0	1	1	T	h	V	r1	r0
II	Read Sector	1	0	0	m	S	E	C	0
II	Write Sector	1	0	1	m	S	E	C	a0
III	Read Adress	1	1	0	0	0	E	0	0
III	Read Track	1	1	1	0	0	E	0	0
III	Write Track	1	1	1	1	0	E	0	0
IV	Force Interrupt	1	1	0	1	i3	i2	i1	i0

As flags mostradas na tabela da página anterior estão descritas abaixo.

r1,r0	Taxa de passo do motor (0=6ms, 1=12ms, 2=20ms, 3=30ms)
V	Flag de verificação do número da trilha (0=não, 1=verifica dest)
h	Flag de cabeçote (1=posiciona cabeçote na trilha 0)
T	Flag de atualização de trilha (1=atualiza registrador de trilha)
a0	Marca de endereço de dados (0=FB, 1=F8 [DAM deletada])
C	Flag de comparação de lado (1=habilita comparação de lado)
E	Atraso de 15ms (1=ativa atraso de 15ms)
S	Flag de comparação de lado (0=compara para lado 0, 1=compara para lado 1)
m	Flag de múltiplos registros (0=registro simples, 1=múltiplos registros)
i3-i0	Termina sem interrupção (INTRQ)
i3	1 = interrupção imediata, requer reset
i2	1 = pulso de índice
i1	1 = transição pronto para não pronto
i0	1 = transição não pronto para pronto

## COMANDOS TIPO I

Os comandos tipo I são usados para mover o cabeçote do drive. O passo do motor normalmente é setado para 6 ms ( $r1$  e  $r0 = 0$ ) para drives de disquetes de 3½". Uma verificação opcional da posição do cabeçote pode ser feita setando o bit 2 ( $V=1$ ) da palavra de comando.

Quando  $V=1$ , ao completar a busca, o número da trilha do campo ID do primeiro setor encontrado é lido e comparado com o conteúdo do registrador de trilha. Se os dois forem iguais e o CRC do campo ID estiver correto, será gerada uma INTRQ sem erros. Caso contrário, o bit Seek Error do registrador de status será setado.

Quando  $V=0$ , ao completar uma busca, o número da trilha não será verificado. Esse modo deve ser habilitado para disquetes não formatados. O comando termina quando o último pulso for enviado ao motor de passo. É necessária uma pausa antes de ler ou gravar para que o cabeçote estabilize sobre a trilha.

Quando a busca for completada, uma requisição de interrupção é gerada e o bit "busy" do registrador de status é resetado 0. Quando a CPU ler o registrador de status, o sinal de interrupção é resetado.

### Comando Restore (busca trilha 0)

Esse comando posiciona o cabeçote do drive na trilha 0. O registrador de trilha será levado a 0 e uma interrupção será gerada quando a

trilha 0 for atingida.

### **Comando Seek**

Esse comando posiciona o cabeçote na trilha indicada pelo registrador de dados. O FDC atualizará o registrador de trilha e enviará pulsos ao motor de passo até que o cabeçote atinja a posição desejada. Uma interrupção será gerada ao final do comando.

### **Comandos Step-In, Step-Out e Step**

Esses comandos enviam um pulso ao motor de passo. O comando Step-Out movimenta o cabeçote em direção à trilha 0, Step-Out em direção à última trilha e Step movimenta para a mesma direção do comando anterior. O registrador de trilha será atualizado somente se o bit "T" estiver setado na palavra de comando. Uma interrupção será gerada ao final do comando.

## **COMANDOS TIPO II**

Os comandos tipo II são usados para ler e escrever setores no disco. Antes de executar um comando tipo II, o registrador de setor deve ser carregado com o setor desejado. Ao receber um comando tipo II, o bit "busy" do registrador de status é setado. Se o campo ID do setor com a trilha e setor corretos não for encontrado, a flag "setor não encontrado" do registrador de status será setada e uma interrupção será gerada.

A flag m indica múltiplos setores. Se for 0, será acessado um único setor; se for 1, múltiplos setores são acessados. Nesse caso, o registrador de setor vai sendo atualizado e uma verificação de endereço pode ocorrer a cada setor lido. O FDC vai acessando os setores em ordem ascendente até que o registrador de setor exceda o número de setores da trilha ou até que uma interrupção forçada seja solicitada (comando Force Interrupt).

A flag C é usada para habilitar a comparação de lado do disco. Se for 0, não haverá comparação. Se for 1, o bit LSB do campo ID do disco é lido e comparado com o conteúdo da flag S.

### **Comando Read Sector**

Ao receber esse comando, o cabeçote é posicionado, o bit "busy" do registrador de status é setado, e quando o campo ID é encontrado e contiver a trilha correta, setor correto, lado correto e CRC correto, o campo de dados é disponibilizado à CPU. Uma DRQ é gerada sempre que o registrador de dados contiver um dado válido. Nesse caso, a CPU

deve ler o dado imediatamente. O bit "lost data" do registrador de status será setado se a CPU não leu o dado em tempo, mas a leitura continuará até o fim do setor ser atingido. Ao final da operação de leitura, o tipo "data address mark" encontrado no campo de dados será gravado no registrador de status (bit 5).

### **Comando Write Sector**

Ao receber esse comando, o cabeçote é posicionado, o bit "busy" do registrador de status é setado, e quando o campo ID é encontrado e contiver a trilha correta, setor correto, lado correto e CRC correto, uma DRQ é gerada. O FDC conta 22 bytes (em dupla densidade) do CRC e a saída "write gate" é ativada se a DRQ for respondida. Se a DRQ não for respondida, o comando é encerrado e o bit "lost data" do registrador de status é setado. Se a DRQ for respondida, 12 bytes 00H (em dupla densidade) serão escritos no disco. Então a DAM (data address mark) é determinada pelo campo a0 do comando. Após isso, o FDC escreverá o campo de dados e gerará DRQ's para a CPU. Se a DRQ não for respondida em tempo para escrita contínua, o bit "lost data" do registrador de status será setado e um byte 00H será escrito no disco. O comando continuará até que o último byte do setor seja atingido. Após o último byte de dados ser escrito, um CRC de dois bytes é computado internamente e escrito no disco, seguido por um byte FFH.

### **COMANDOS TIPO III**

Os comandos tipo III são usados para acessar os headers das trilhas e setores do disco.

#### **Comando Read Address**

Ao receber esse comando, o cabeçote é posicionado, o bit "busy" do registrador de status é setado. O próximo campo ID encontrado é lido e os seis bytes de dados do campo ID são montados e transferidos para o registrador de dados. Uma DRQ é gerada por cada byte lido. Os seis bytes ID são:

- |                        |                      |
|------------------------|----------------------|
| 1 - Endereço de trilha | 4 - Tamanho do setor |
| 2 - Número do lado     | 5 - CRC1             |
| 3 - Endereço de setor  | 6 - CRC2             |

Embora os bytes CRC sejam transferidos para a CPU, o FDC checa a validade dos mesmos e o bit "CRC error" do registrador de status será setado se houver erro de CRC. O endereço de trilha do campo ID é escrito no registrador de setor para possibilitar uma comparação pelo usuário, se for desejável. Ao final do comando, uma interrupção é gerada e o bit "busy" do registrador de status é resetado.

### **Comando Read Track**

Ao receber esse comando, o cabeçote é posicionado e o bit "busy" do registrador de status é setado. A leitura é iniciada imediatamente ao primeiro pulso de indexação encontrado e continua até o pulso de indexação seguinte. Todos os gaps, headers e bytes de dados são montados e transferidos para o registrador de dados. Uma DRQ é gerada para cada byte transferido. A acumulação de bytes é sincronizada para cada marca de endereço encontrada. Uma interrupção é gerada quando o comando for completado. O ID da marca de endereço, campo ID, ID dos bytes CRC, DAM, dados e bytes de dados do CRC para cada setor devem estar corretos. Os bytes gap podem ser lidos incorretamente durante a pausa na escrita por causa da sincronização.

### **Comando Write Track (formatação de trilha)**

Ao receber esse comando, o cabeçote é posicionado, o bit "busy" do registrador de status é setado. A escrita é iniciada imediatamente ao primeiro pulso de indexação encontrado e continua até o próximo pulso de indexação, quando então a interrupção é ativada. A requisição de dados é ativada imediatamente ao receber o comando, mas a escrita não será iniciada antes do primeiro byte de dados ser escrito no registrador de dados. Se este não for carregado na temporização do pulso de indexação, a operação é terminada com dispositivo não ocupado, o bit "lost data" do registrador de status é setado e a interrupção é ativada. Se um byte não estiver presente no registrador de dados quando necessário, será assumido um byte 00H. Essa seqüência é repetida de uma marca de indexação a outra.

Normalmente, qualquer padrão de dados que for carregado no registrador de dados é escrito no disco com um ciclo padrão normal. Entretanto, se o FDC detectar um padrão de dados de F5H até FEH no registrador de dados, será interpretado como marca de endereço, sem geração de ciclos ou CRC. O gerador CRC é inicializado quando um byte F5H está para ser transferido (em MFM). Um byte F7H gerará dois bytes CRC. Como consequência, os bytes F5H a FEH não podem fazer parte dos gaps, campos de dados ou campos ID. Na formatação das trilhas, os setores podem conter 128, 256, 512 ou 1024 bytes.

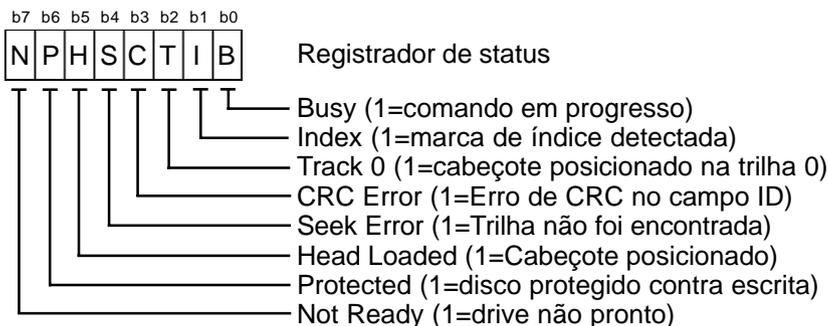
### **COMANDO TIPO IV**

O comando tipo IV (force interrupt) é geralmente usado para encerrar o acesso a múltiplos setores. Esse comando pode ser carregado no registrador de comando a qualquer tempo. Se houver um comando em execução (bit "busy" = 1), o comando será encerrado e o bit "busy" do registrador de status será resetado.

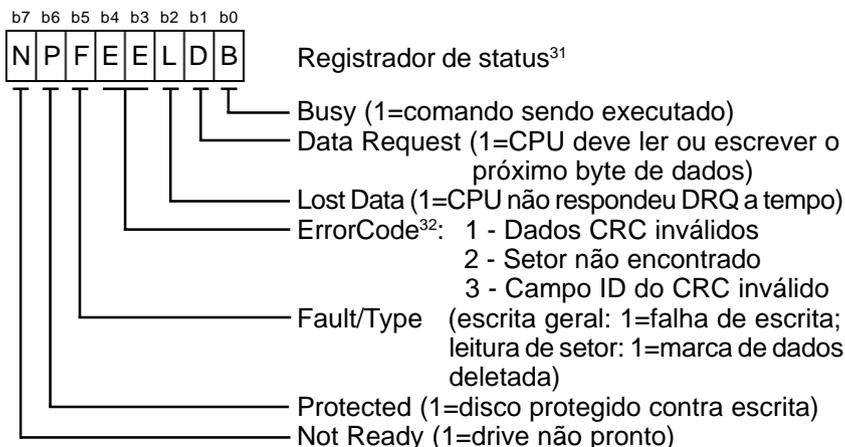
## 3.2 - O REGISTRADOR DE STATUS

Ao receber qualquer comando, à exceção do comando Force Interrupt, o bit “busy” é setado e os outros bits do registrador de status são atualizados ou limpos para o novo comando. O usuário tem a opção de ler o registrador de status através do programa ou usar a linha DRQ juntamente com DMA ou interrupção. Quando o registrador de dados é lido, o bit DRQ no registrador de status e a linha DRQ são automaticamente resetadas. Uma escrita no registrador de dados também causa o mesmo efeito. O bit “busy” deve ser sempre monitorado para que o usuário saiba quando um comando está sendo executado. Ao usar a INTRQ, a checagem do bit “busy” não é recomendada porque a leitura desse bit resetará a linha INTRQ.

### STATUS PARA COMANDOS TIPO I



### STATUS PARA COMANDOS TIPO II e III



**Nota 31:** Os bits 1~6 são resetados quando atualizados.

**Nota 32:** Códigos não válidos para escrita ou leitura de trilhas.

## STATUS PARA COMANDOS TIPO IV

Se um comando “Force Interrupt” for recebido durante a execução de outro comando, o bit 0 (busy) do registrador de status é resetado e os bits restantes permanecem inalterados. Entretanto, se um comando “Force Interrupt” for recebido quando nenhum outro comando estiver em execução, o bit 0 (busy) do registrador de status é resetado e os outros bits serão atualizados ou limpos. Nesse caso, o registrador de status se comporta como nos comandos tipo I.

### 3.3 - FUNÇÕES ADICIONAIS

O FDC não possui internamente seleção para número de drive, lado, densidade de disco e controle liga/desliga motor dos drives. Essas funções devem ser adicionadas por circuitos externos, que devem ser controlados separadamente.

Como o FDC tem somente um registrador de trilha que deve ser usado para todos os drives, a posição da trilha deve ser salva na memória e o registrador de trilha deve ser atualizado a cada troca de drive.

### 3.4 - FORMATAÇÃO

Para que o disco possa ser utilizado, é necessário um processo chamado *formatação*. Na formatação, o disco é dividido logicamente em trilhas e setores. A tabela abaixo mostra o padrão de dados e sua interpretação pelo FDC no sistema MFM.

00~F4	Escreve 00 até F4
F5	Escreve A1, preset CRC
F6	Escreve C2
F7	Gera 2 bytes CRC
F8~FF	Escreve F8 até FF

#### EXEMPLO DE FORMATAÇÃO

O exemplo da página seguinte mostra a seqüência de dados que deve ser enviada para o comando “Write Track” para formatar um disco com 256 bytes por setor (o MSX usa setores de 512 bytes por padrão). Os valores à esquerda são contadores de repetição para escrita (em decimal) para os valores da direita.

Primeiro, o cabeçalho de trilha (Track Header) deve ser escrito, seguido pelo ID de setor e campos de dados dos setores (para cada setor). Finalmente, bytes 4EH devem ser escritos até o comando ser completado.

Track Header (cabeçalho de trilha)		
80	x	4EH
12	x	00H
03	x	F6H (escreve C2)
01	x	FCH (marca de índice)
50	x	4EH
Campo ID do setor		
12	x	00H
03	x	F5H (escreve A1, preset CRC)
01	x	FEH (marca ID de endereço)
01	x	número da trilha
01	x	número do lado
01	x	número do setor
01	x	01 (tamanho do setor - 256 bytes)
01	x	F7H (escreve 2 bytes CRC)
22	x	4EH
Campo de dados do setor		
12	x	00H
03	x	F5H (escreve A1, preset CRC)
01	x	FBH (marca da dados de endereço)
256	x	dados do setor
1	x	F7H (escreve 2 bytes CRC)
54	x	4EH
Fim de trilha (preencher bytes não usados)		
...	x	4EH

### 3.5 - ENDEREÇOS DE ACESSO AO FDC

Nessa seção estão descritos diversos endereços de acesso para interfaces de drive, baseadas tanto em memória como em I/O. Para interfaces acessadas por memória, o slot onde a mesma estiver instalada deve estar habilitado. Para interfaces acessadas por I/O, não é necessário esse cuidado, pois elas são acessadas diretamente por portas de I/O. Esse tipo de acesso foi utilizado somente em interfaces brasileiras. O acesso padrão para o MSX é por memória.

#### ENDEREÇOS PARA ACESSO POR MEMÓRIA (Padrão)

7FF8H	R	Registrador de status
7FF8H	W	Registrador de comando
7FF9H	R/W	Registrador de trilha
7FFAH	R/W	Registrador de setor
7FFBH	R/W	Registrador de dados
7FFCH	R?/W	Lado (bit 0) [Motor aqui?]
7FFDH	R?/W	Drive (bit 0) [Motor aqui?]
7FFEH	-	Não usado
7FFFH	R	Requisição de dados (bit 7) e busy (bit 6)

**Nota:** O MSXDOS/BarbarianLoader seleciona a memória nos endereços 8000H~BFFFH; nesse caso, devem ser usados os endereços BFFxH ao invés de 7FFxH.

### ENDEREÇOS PARA ACESSO POR MEMÓRIA (Alternativo)

Esse mapeamento é utilizado somente pelo modelo SV738 (X'Press) da SpectraVideo, pelo BDOS da Technohead e pelo BDOS arábico. Nesse último caso, os endereços usados são 7F80H~7F87H, e nos dois primeiros casos são 7FB8H~7FBFH.

7FB8H/7F80H	R	Registrador de status
7FB8H/7F80H	W	Registrador de comando
7FB9H/7F81H	R/W	Registrador de trilha
7FBAH/7F82H	R/W	Registrador de setor
7FBBH/7F83H	R/W	Registrador de dados
7FBCH/7F84H	R	bit 7 = IRQ/Não ocupado
		bit 6 = Requisição de dados
7FBCH/7F84H	W	bits 0/1 = Seleciona drive
		bit 2 = lado
		bit 3 = lado

Os endereços 7FBDH~7FBFH e 7F85H~7F87H não são usados.

### ENDEREÇOS PARA ACESSO POR PORTAS DE I/O

D0H	R	Registrador de status
D0H	W	Registrador de comando
D1H	R/W	Registrador de trilha
D2H	R/W	Registrador de setor
D3H	R/W	Registrador de dados
D4H	W	Drive (bit 1), Lado (bit 4), Motor (bit ??)
D4H	R	IRQ/Não ocupado (bit 7), Requisição de dados (bit 6)

Os endereços de D5H a D7H não são usados. Esse tipo de acesso é usado por todas as interfaces brasileiras, exceto pela ACVS/CIEL que usa o acesso por memória padrão.

A leitura pela porta D4H somente é suportada pela versão 3.0 ou superior. Para versões anteriores são usados os bits 0 e 1 do registrador de status, que têm o mesmo significado. A versão 2.7 e superiores usam acesso misto, por portas de I/O e por memória padrão.