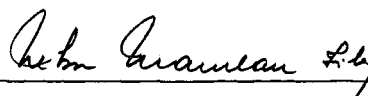


SOFTWARE PARA CONTROLE DE  
DISCOS FLEXÍVEIS

Roberto Manoel Juckowsky Macedo

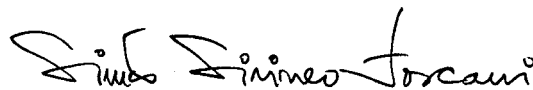
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE  
PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE  
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO  
DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.).

Aprovada por:



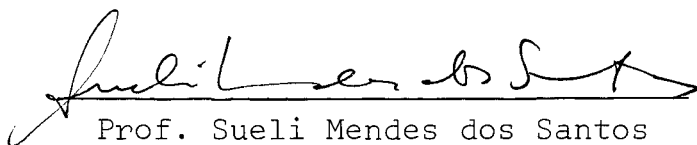
---

Prof. Nelson Maculan Filho  
Presidente



---

Prof. Simão Sirineu Toscani



---

Prof. Sueli Mendes dos Santos

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 1979

MACEDO, ROBERTO MANOEL JUCKOWSKY

Software para Controle de Discos Flexíveis [Rio de Janeiro] 1979.

XII, 126p. 29,7cm (COPPE-UFRJ, M.Sc, Engenharia de Sistemas, 1979)

Tese - Univ. Fed. Rio de Janeiro. Fac. Engenharia

1. Utilização de Discos Flexíveis

I. COPPE/UFRJ

II. Título (série)



## AGRADECIMENTOS

Ao Professor Simão Sirineo Toscani, orientador deste trabalho, pelo apoio técnico e dedicação.

Ao Professor Nelson Maculan, co-orientador deste trabalho, pela colaboração e ajuda junto à COPPE.

Ao Professor Manoel Luiz Leão, Diretor do Centro de Processamento de Dados da Universidade Federal do Rio Grande do Sul, pelo incentivo ao aperfeiçoamento e pelas facilidades que proporcionou para o desenvolvimento deste trabalho.

Ao Professor Daltro José Nunes, Coordenador do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal do Rio Grande do Sul, pelo estímulo.

Ao Professor Ronaldo Cesar Marinho Perciano, Coordenador do Programa de Engenharia de Sistemas da COPPE/UFRJ, pela compreensão.

Ao Professor Altamiro Amadeu Suzim e ao Eng. Ricardo Menna Barreto Felizzola pelas valiosas informações que permitiram um melhor entendimento de "hardware" do Controlador de Discos Flexíveis.

Aos meus familiares pelo apoio moral em todas as horas.

Aos colegas pelo incentivo.

A todos aqueles que contribuíram, direta ou indiretamente, para a realização deste trabalho.

## SINOPSE

Descreve-se a implementação de um Sistema, em um controlador programável, que facilita e otimiza a utilização de discos flexíveis. O controlador, baseado em um microprocessador, possui memória própria e admite a ligação de vários discos flexíveis.

O "software" implementado transforma o controlador em um processador de E/S especializado, o qual pode ser ligado a qualquer tipo de computador que tenha interface de comunicação de 16 bits E/S.

A comunicação entre o computador e o controlador é feita através da troca de mensagens, na relação mestre-escravo: o computador envia ordens para execução de tarefas (como por exemplo abrir e fechar arquivos, ler ou gravar registros físicos e inicializar discos) e recebe respostas ao término das execuções.

Desta maneira, o processador do computador mestre fica liberado do grande número de tarefas que seriam necessárias para garantir uma utilização otimizada de arquivos em vários discos flexíveis.

ABSTRACT

An implementation is described of a system on a programable controller that simplifies and optimizes the utilization of floppy disks. The controller, based on a microprocessor, has a local memory and allows the connection of several floppy disks.

The "software" transforms the controller to a specialized I/O processor, that can be connected to any type of computer having a 16 bits I/O communication interface.

The communication between the computer and the controller has been made through the exchange of messages, on a master-slave relationship: the computer sends orders for the execution of tasks (as for instance open and close of files, input or output of physical records and volume initialization) and receives messages of task completion.

Therefore, the processor of the master computer is freed of a number of jobs that would be necessary to guarantee an optimized use of files in several floppy-disks.

ÍNDICE

I.	INTRODUÇÃO.....	1
II.	O CONTROLADOR INTELIGENTE DE DISCOS FLEXÍVEIS.....	3
II.1	O Microprocessador.....	4
II.2	Interrupções.....	5
II.2.1	Interrupções Mascaráveis.....	5
II.2.2	Interrupções Não Mascaráveis.....	10
II.3	Interfaces de Discos Flexíveis.....	10
II.3.1	Registrador de Comando/Estado.....	12
II.3.2	Registrador de Posição.....	16
II.4	O Formatador.....	16
II.4.1	Registrador de Comando/Estado.....	18
II.4.2	Registrador Contador de Palavras.....	23
II.4.3	Registrador de Dado.....	23
II.4.4	Registrador de Comunicação.....	25
II.4.5	Comunicação adicional entre o <u>Mi</u> croprocessador e o Formatador.....	25
III.	CARACTERÍSTICAS DO SISTEMA.....	28
III.1	Comunicação entre Controlador e Computa- dor Mestre.....	29
III.2	Textos de Comando.....	33
III.2.1	Tarefa abre arquivo.....	34
III.2.2	Tarefa faz Entrada ou Saída.....	36
III.2.3	Tarefa fecha arquivo.....	37
III.2.4	Tarefa remove processo.....	38
III.2.5	Tarefa carrega módulo.....	40
III.2.6	Tarefa inicializa disco.....	42

III.3	Texto de Resposta.....	42
III.4	Escolha do Formato para o disco.....	44
III.4.1	Formato IBM.....	44
III.4.1.1	Campo "DADOS" em setores de trilha zero.....	47
III.4.1.2	Campo "DADOS" em setores das trilhas 01 até 76.....	56
III.4.2	O formato escolhido.....	56
III.4.3	Comentários.....	60
III.5	Gerenciamento de arquivos.....	61
III.5.1	Definição das estruturas.....	61
III.5.1.1	Cópia dos diretórios.....	62
III.5.1.2	Descritores dos arquivos em uso.....	63
III.5.1.3	Vetores de ocupação de trilhas.....	64
III.5.1.4	Vetores de ocupação de descritores.....	64
III.5.2	Funcionamento.....	65
III.5.2.1	Abertura de um arquivo.....	66
III.5.2.2	Fechamento de um arquivo...	68
III.6	Resumo das limitações.....	70
IV.	O SISTEMA IMPLEMENTADO.....	72
IV.1	Recepção e enfileiramento de mensagens.....	72
IV.2	Análise do texto e geração de estrutura para otimização do acesso aos discos.....	77
IV.2.1	Análise do texto.....	77
IV.2.2	Geração de estrutura para otimiza- ção do acesso aos discos.....	79



IV.3	Corpo do Programa do Controlador.....	88
IV.4	Seleção da tarefa que otimiza o acesso aos discos .....	92
IV.5	Execução das tarefas .....	99
IV.5.1	Tarefa monta cópia do diretório do disco .....	101
IV.5.2	Tarefa inicializa disco .....	103
IV.5.3	Tarefa abre arquivo .....	108
IV.5.4	Tarefa faz entrada ou saída .....	111
IV.5.5	Tarefa fecha arquivo .....	116
IV.5.6	Tarefa remove processos .....	116
IV.5.7	Tarefa carrega módulo .....	118
IV.6	Transmissão das respostas ao computador mes tre .....	118
IV.7	Procedimento executado quando um disco é des ligado .....	119
V.	CONCLUSÕES .....	123
	BIBLIOGRAFIA .....	126

ÍNDICE DAS FIGURAS

Figura 1 - Mapa de memória para os vetores de interrupção.....	6
Figura 2 - Mapa do vetor de interrupções Mascaráveis.....	7
Figura 3 - Detalhamento do mapa do vetor de interrupções mascaráveis.....	9
Figura 4 - Registradores de controle das unidades de disco flexível.....	11
Figura 5 - Registradores de Comando/Estado dos interfaces dos discos flexíveis.....	13
Figura 6 - Ligação Formatador x Interfaces.....	17
Figura 7 - Registrador de Comando/Estado do formatador...	19
Figura 8 - Estados do formatador.....	21
Figura 9 - Estados e Atividades de formatador.....	24
Figura 10 - Autômato Transmissão.....	31
Figura 11 - Autômato Recepção.....	32
Figura 12 - Lay-out de trilha: (Formato IBM).....	45
Figura 13 - Campo de "Dados" geral.....	48
Figura 14 - Campo de "Dados" do setor 5 da trilha zero....	49
Figura 15 - Campo de "Dados" do setor 7 da trilha zero....	51
Figura 16 - Seqüência de registros nos setores de uma trilha.....	52
Figura 17 - Campo de "Dados" dos setores 08-26 da trilha zero .....	54
Figura 18 - Setor 05 (Formato Controlador).....	59
Figura 19 - Fila de buffers para recepção de mensagens vindas do computador mestre.....	74

Figura 20 - Rotina de inserção de uma mensagem na "fila de mensagens recebidas".....	75
Figura 21 - Fila de mensagens recebidas.....	76
Figura 22 - Rotina de remoção de uma mensagem da "fila de mensagens recebidas".....	78
Figura 23 - Rotina de inserção de uma mensagem na "fila de mensagens a transmitir".....	78
Figura 24 - Rotina de inserção de uma mensagem na "fila de mensagens analisadas".....	80
Figura 25 - Estrutura para otimização de acesso aos discos.....	83
Figura 26 - Rotina para geração de uma entrada na fila para otimização de acesso aos discos.....	85
Figura 27 - Situação da inserção na estrutura para otimização de acesso aos discos.....	87
Figura 28 - Uma situação para as estruturas de discos definidos na seção IV.2.....	89
Figura 29 - Fluxograma básico do corpo do controlador.....	90
Figura 30 - Rotina de busca da tarefa a executar.....	97
Figura 31 - Rotina para remoção da tarefa de estrutura para otimização de acesso.....	100
Figura 32 - Rotina para remoção de mensagens da "fila de mensagens analisadas".....	104
Figura 33 - Rotina de remoção de uma mensagem da "fila de mensagens a transmitir".....	120
Figura 34 - Rotina de liberação da área de mensagens.....	121

ÍNDICE DE TABELAS

Tabela 1 - Erros da tarefa "Inicializa disco".....	107
Tabela 2 - Erros da tarefa "Abre arquivo".....	109
Tabela 3 - Erros, na fase pré-operativa, de tarefa "Faz entrada ou saída".....	114
Tabela 4 - Erros, na fase pós-operativa, da tarefa "Faz entrada ou saída".....	115
Tabela 5 - Erros da tarefa "Fecha arquivo".....	117

## I. INTRODUÇÃO

Os objetivos do trabalho aqui descrito foram projetar e implementar um programa de controle que permitisse otimizar o funcionamento do Controlador Inteligente de Discos Flexíveis, desenvolvido no Curso de Pós-Graduação em Ciência da Computação da Universidade Federal do Rio Grande do Sul.

O "software" implementado transforma o Controlador de Discos Flexíveis em um processador (canal) de E/S inteligente que libera o computador, ao qual estiver ligado, da maioria das tarefas relacionadas com a utilização de arquivos em discos flexíveis. O Controlador passa a ser visto, pelo computador, como um processador especializado em operações de E/S sobre discos flexíveis. Este processador de E/S oferece, ao computador, facilidades para a utilização dos discos flexíveis como memória de massa, tomando para si as tarefas relacionadas com o gerenciamento dos arquivos e com a otimização dos acessos para operações de leitura/gravação nos discos.

Este "software", mantendo informações sobre os discos ligados ao controlador e sobre os arquivos em uso pelo computador, permite a utilização simultânea de vários arquivos e se encarrega dos detalhes relacionados com operações de E/S ou atualização de diretórios.

Desta maneira, o trabalho do computador se resume ao encaminhamento de ordens, ao controlador, para a execução de tarefas de alto nível sobre os discos flexíveis.

Existirá, assim, uma relação mestre-escravo entre o computador e o controlador.

## II. O CONTROLADOR INTELIGENTE DE DISCOS FLEXÍVEIS

Neste capítulo é apresentado um resumo das características de "hardware" do Controlador Inteligente de Discos Flexíveis. Maiores detalhes podem ser obtidos em SUZIM<sup>1</sup>.

O Controlador Inteligente de Discos Flexíveis (CIDF) está baseado em um microprocessador Motorola (MC68000).

Cada unidade de disco flexível é ligada a ele através de um interface, cujos sinais de controle são transmitidos ao microprocessador através de interrupções.

Os sinais de dados são processados por um interface especializado chamado Formatador, compartilhado por todas as unidades e chaveado, no interface correspondente, pelo microprocessador.

A unidade que opera com o disco flexível é bastante simples. O disco em si é feito de "mylar" revestido de óxido e está contido em um envelope de papel para proteção. Uma ranhura no envelope permite o acesso ao disco que tem, aproximadamente, 20cm de diâmetro com um furo de 4cm para centralização.

Com a unidade em operação, o disco está continuamente girando, dentro do envelope, a 360 RPM o que proporciona um tempo de latência média da ordem de 83ms. A cabeça de leitura/gravação tem um movimento radial, tracionada por um motor passo-a-passo. Cada passo do motor determina uma trilha do

disco (existindo 77 trilhas numeradas de 0 a 76) com tempo de passagem, de uma trilha para outra, de 6ms. Um sensor detecta a trilha zero (a mais externa) e, a partir daí, o controle da trilha deverá ser efetuado por "software".

A capacidade total de armazenamento de um disco flexível é de, aproximadamente, 400KB. Levando em conta que são necessários dados de controle e sincronização, para a utilização do disco, há uma redução da capacidade total, em função do formato escolhido. No formato IBM, por exemplo, podem ser armazenados somente 256KB.

O método de gravação utilizado é o PE-MARK. Neste método a gravação de um bit corresponde a uma reversão da corrente na cabeça gravadora, quando deve ser gravado um "zero", e duas reversões, quando a informação a ser gravada é "um".

Na recepção, a unidade de disco devolve o sinal da mesma forma como o recebeu, ou seja, um pulso para cada reversão de fluxo no disco.

## II.1

### 0 Microprocessador

O microprocessador MC6800 possui um repertório de 72 instruções e tem 7 modos de endereçamento. As instruções podem ser de um, dois ou três bytes e requerem, para suas execuções, de 2 a 12 ciclos de máquina.

Possui 3 registradores de 16 bits ("Program Counter", "Stack Pointer" e "Index Register") e 3 registradores de 8 bits (dois Acumuladores e um "Condition Register").



Existem duas linhas de interrupção: uma mascarável e outra não mascarável.

Para possibilitar a sincronização, durante uma leitura ou gravação de dados no disco, foi implementado um circuito que, associado com um temporizador programável, permite ao microprocessador colocar-se no estado de PARADO (HALT) e sair do mesmo, por programa.

## II.2 Interrupções

O microprocessador MC6800 reserva as interrupções não mascaráveis (de uso exclusivo das rotinas de leitura e gravação em disco) ao formatador e as interrupções mascaráveis ao "OU" de todos os outros pedidos de interrupção.

Quando há uma interrupção, o microprocessador desvia para o endereço que está armazenado nas posições de memória mostradas na figura (1).

### II.2.1 Interrupções Mascaráveis

A identificação da origem de uma interrupção é efetuada, na rotina de atendimento, pela análise de um vetor de 128 bits, correspondentes aos endereços de memória de  $0010_{16}$  a  $001F_{16}$ , conforme figura (2).

Estes endereços só podem ser lidos e cada

POS.DE MEMÓRIA	EVENTO
FFFE e FFFF	Inicialização (RESET)
AFFC e AFFD	Interrupção não mascarável
FFFA e FFFB	Interrupção causada por programa
FFF8 e FFF9	Interrupção mascarável

Figura 1 - Mapa de memória para os vetores de interrupção

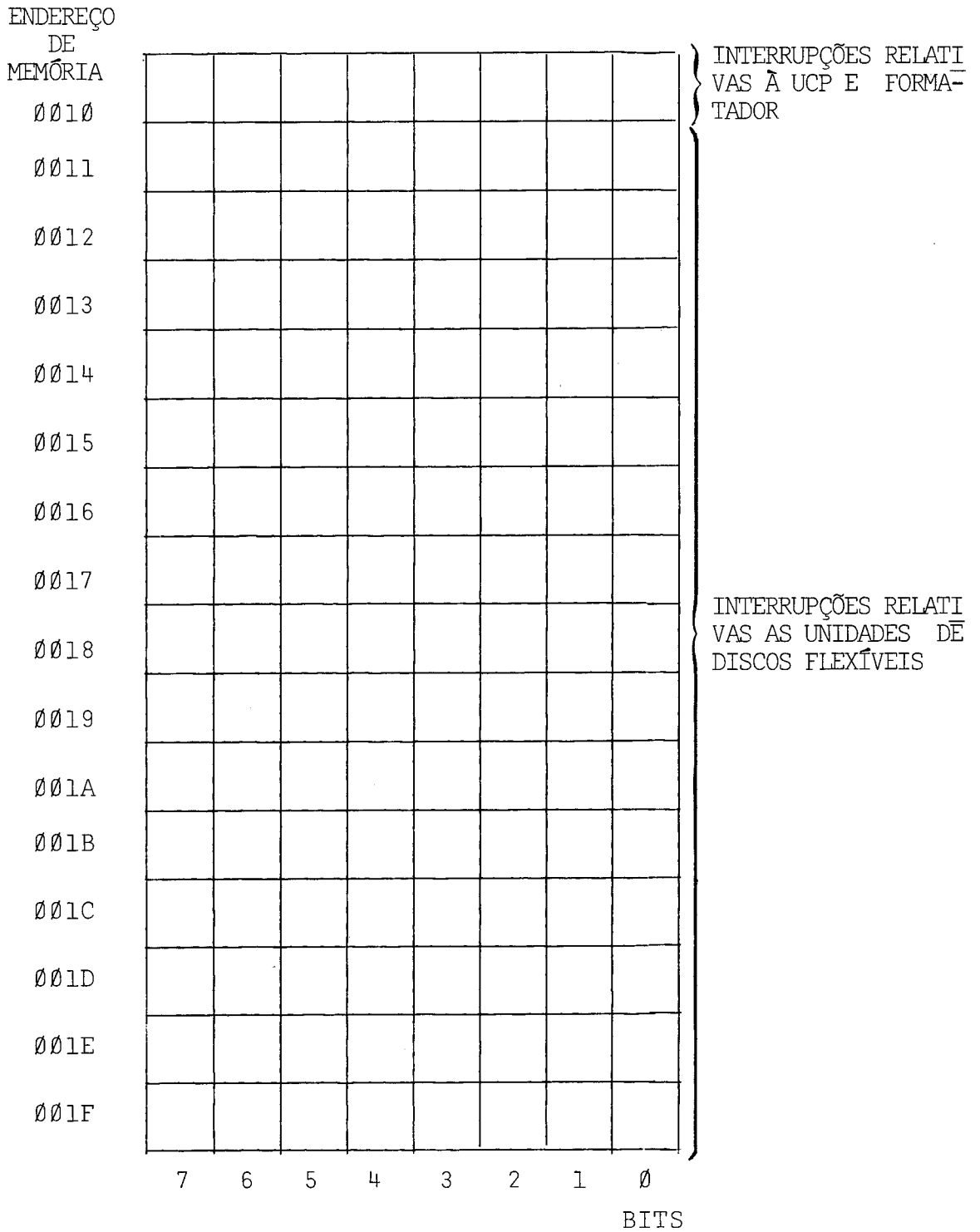
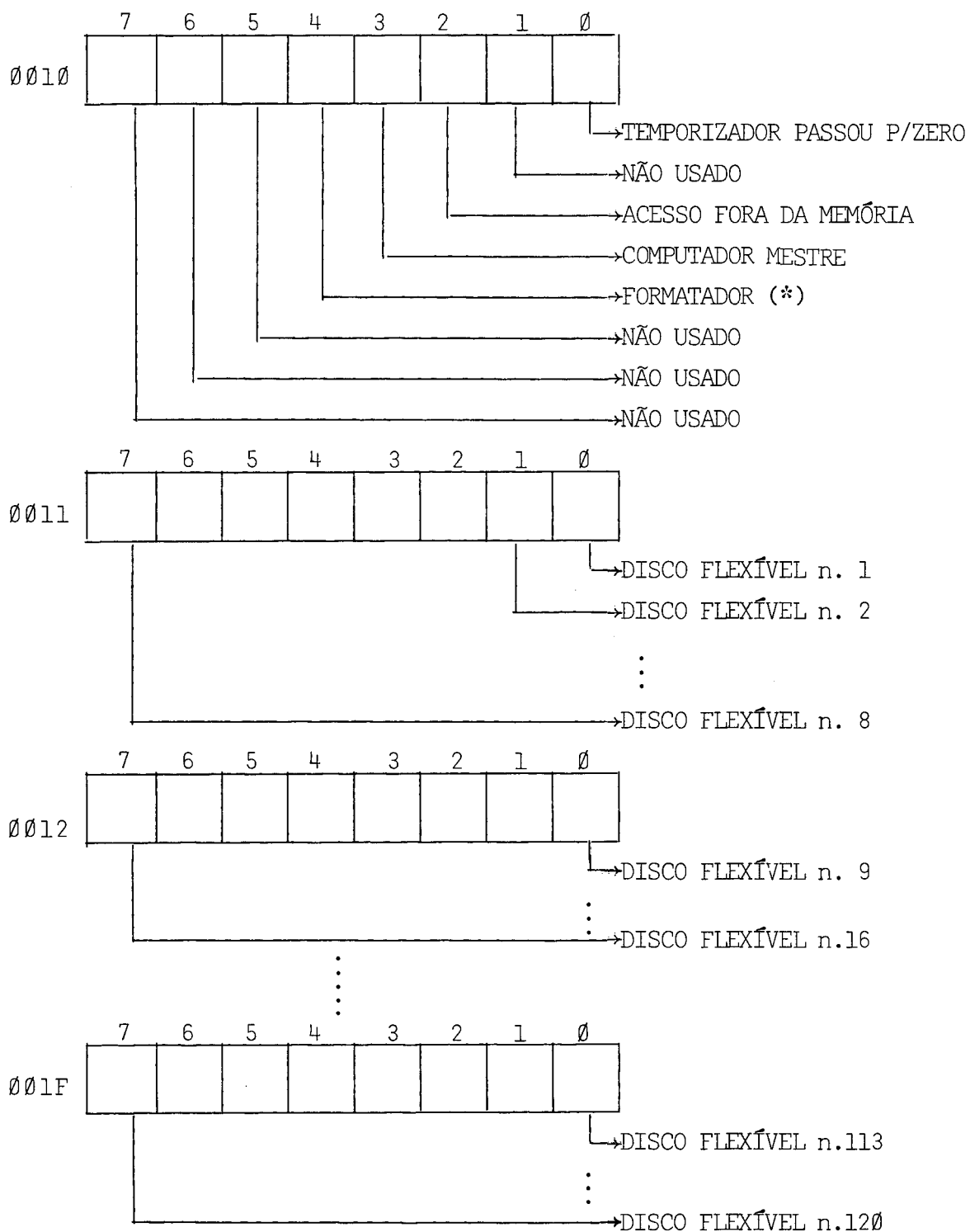


Figura 2 - Mapa do Vetor de Interrupções Mascaráveis

bit identifica, univocamente, uma fonte de interrupção, conforme é mostrado na figura (3).

O endereço de atendimento das interrupções mascaráveis deve ser previamente armazenado nas posições  $FFF8_{16}$  e  $FFF9_{16}$  de memória.



(\*) As interrupções mascaráveis geradas pelo formatador são devidas a erros ou pedido de transferência de um byte, já lido ou para ser gravado, quando não estiver funcionando no modo ADM.

Figura 3 - Detalhamento do mapa do vetor de interrupções mascaráveis

## II.2.2 Interrupções Não Mascaráveis

Estas interrupções são de uso exclusivo das rotinas de leitura e gravação em disco.

O endereço para onde deve ser transferido o controle quando ocorre uma interrupção não mascarável deve ser armazenado nas posições  $AFFC_{16}$  e  $AFFD_{16}$  de memória.

No item II.4, que descreve o formatador, estas interrupções serão consideradas de forma mais detalhada.

## II.3 Interfaces de Discos Flexíveis

Os interfaces de disco são os elementos de comunicação entre o microprocessador e as unidades de disco.

Constam de 3 registradores (sendo que dois deles correspondem ao mesmo endereço): registrador de COMANDO/ESTADO e registrador de POSIÇÃO.

Desta maneira para cada disco tem-se duas posições contíguas de memória que fornecem as informações necessárias ao seu controle, conforme figura (4).

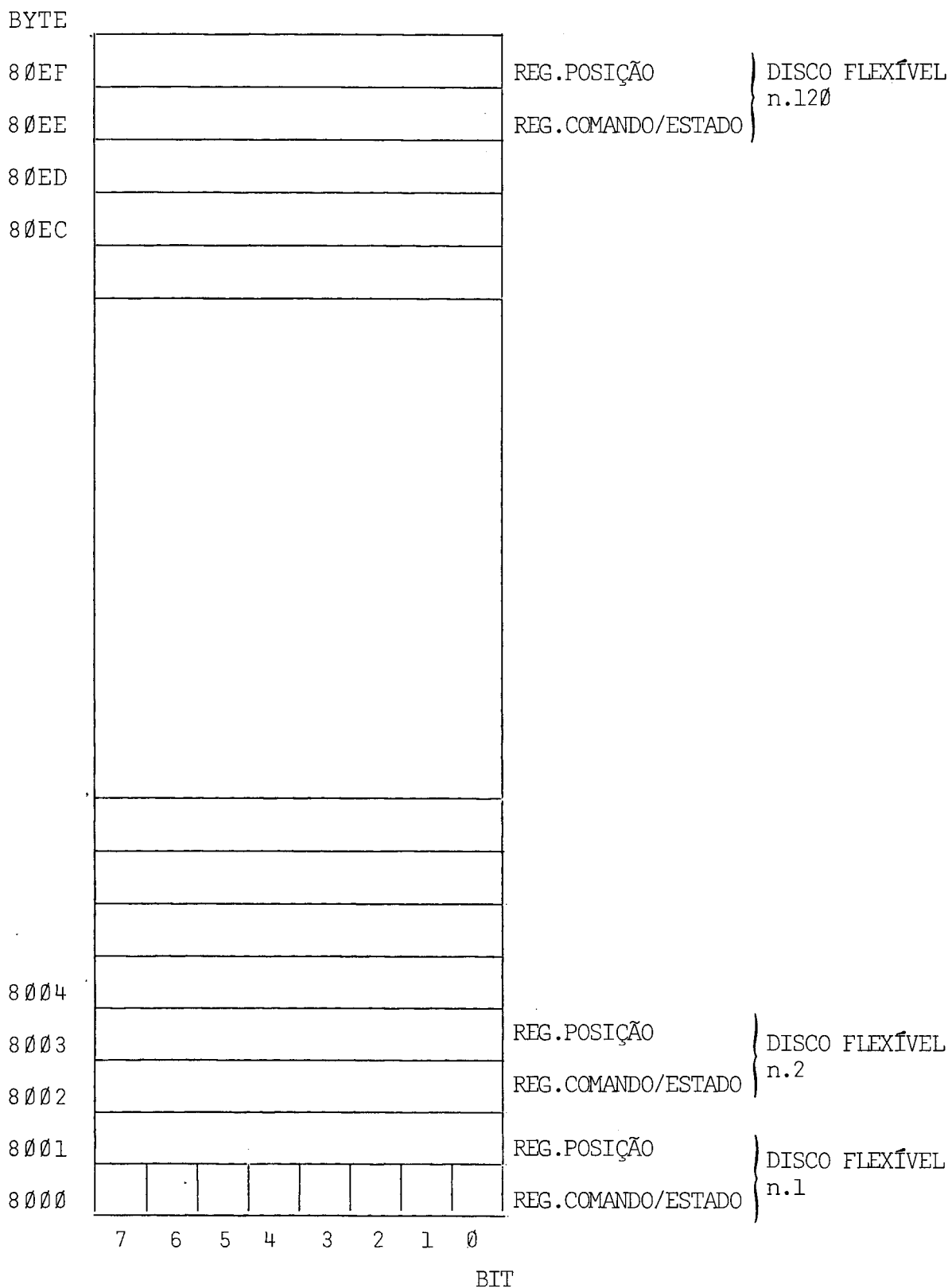


Figura 4 - Registradores de controle das unidades de disco flexível

## II.3.1

## Registrador de Comando/Estado

São na verdade 2 registradores, onde o registrador de Comando só responde à escrita e o registrador de Estado só responde à leitura.

O registrador de Estado é zerado após a leitura.

O significado de cada bit é dado na figura (5).



BIT	REGIST.COMANDO	REGIST.ESTADO
7	seleciona	pronto
6	passo	índice
5	protege	tempo de setor
4	transferindo	tempo de trilha
3	além da trilha 43	trilha 00
2	sincronismo PLL	protegido
1	carga da cabeça	não utilizado
0	direção	não utilizado

Figura 5 - Registrador de Comando/Estado dos interfaces dos discos flexíveis

A interpretação de cada bit é feita da seguinte maneira:

- REGISTRADOR DE COMANDO:

- SELECIONA:            0 = sem efeito  
                          1 = seleciona unidade de disco
- PASSO:                0 = sem efeito  
                          1 = comanda movimento da cabeça
- PROTEGE:             0 = sem efeito  
                          1 = liga o Flip-Flop de disco protegido
- TRANSFERINDO:        0 = sem efeito  
                          1 = chaveia o interface correspondente ao formatador
- ALÉM TRILHA 43:      0 = para leitura e gravação nas trilhas de 0 a 43  
                          1 = para leitura e gravação nas trilhas de 44 a 76
- SINCRONISMO PLL:     0 = sem efeito  
                          1 = faz com que o PLL ("Phase Locked Loop") se sincronize num preâmbulo de zeros (circuito não existente)
- CARGA DA CABEÇA:    0 = determina a elevação da cabeça de leitura/gravação do disco, se esta estava baixa.  
                          1 = determina a carga da cabeça de leitura/gravação contra o disco (habilita transferências).
- DIREÇÃO:             0 = o sentido de deslocamento da cabeça é para dentro.  
                          1 = o sentido de deslocamento da cabeça é para fora.

- REGISTRADOR DE ESTADO:

- PRONTO:                    1 = indica que o disco não está em condições de operar  
                                   0 = indica que o disco está em condições de operar
- ÍNDICE:                    1 = sem efeito  
                                   0 = indica que a cabeça de leitura/gravação está no começo de uma trilha
- TEMPO DE SETOR:        1 = sem efeito  
                                   0 = indica que o disco foi ligado ou desligado ou que atingiu o setor desejado (tempo estipulado no registrador de posição)
- TEMPO DE TRILHA:      1 = sem efeito  
                                   0 = indica que houve um pulso de passo (movimento da cabeça)
- TRILHA 00:                1 = indica que a cabeça de leitura/gravação não está na trilha 00 do disco  
                                   0 = indica que a cabeça de leitura/gravação está na trilha 00 do disco
- PROTEGIDO:                1 = indica que o disco não está protegido contra gravações  
                                   0 = indica que o disco não pode ser gravado

Os sinais de ÍNDICE, TEMPO DE SETOR e TEMPO DE TRILHA, quando em 0, causam uma interrupção no microprocessador.

### II.3.2 Registrador de Posição

Este registrador fornece ao interface do disco o tempo em que o microprocessador deverá ser interrompido por "Tempo de Setor" contado a partir de um sinal de Índice (começo da Trilha).

Com isto pode-se selecionar o setor do disco desejado.

O valor a ser armazenado neste registrador deve ser expresso em milisegundos (a cada 6ms alcança-se um novo setor do disco).

### II.4 O Formatador

O formatador é o circuito que trata dos dados durante as transferências dos mesmos entre os discos e o sistema externo ou entre os discos e a memória do microprocessador.

O formatador é utilizado por todos os interfaces, mas somente por um de cada vez, conforme ilustrado na figura (6).

Durante a gravação, o formatador serializa a informação que é enviada ao disco, insere os sinais de relógio, gera as marcas de endereço, calcula e grava o Caráter de Conferência de Bloco ("Cyclic Redundancy Check"). Durante a leitura, entra em sincronismo com os dados do disco, reconhe-

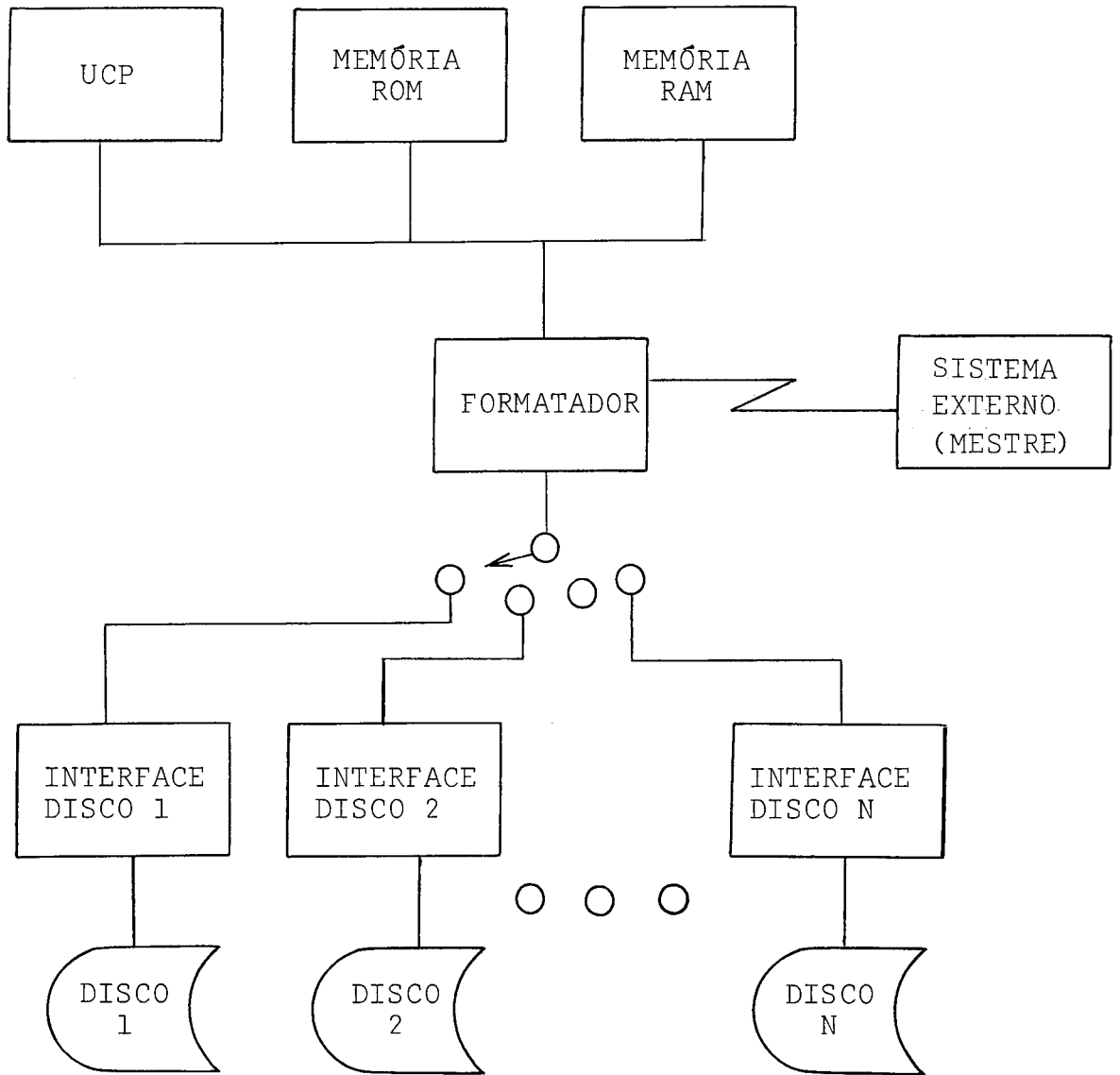


Figura 6 - Ligação Formatador x Interfaces

ce a marca de endereço, verifica o CCB (Caráter de Conferência de Bloco), analisa os bits de relógio e paraleliza os bits de dados.

Estas funções são monitoradas pelo microprocessador, o qual se comunica com o formatador através de quatro registradores:

- Registrador de Comando/Estado
- Registrador Contador de Palavras
- Registrador de Dado
- Registrador de Comunicação

Além dos registradores, mais três linhas estabelecem comunicação entre o Formatador e microprocessador:

- Interrupção não mascarável
- Interrupção mascarável
- Sinal de byte completo

#### II.4.1

#### Registrador de Comando/Estado

O microprocessador pode atuar sobre o formatador através do registrador de comando e pode ser informado, sobre o que está acontecendo, através do registrador de estado.

O significado de cada bit do registrador de Comando/Estado aparece na figura (7).

BIT	COMANDO	ESTADO
7	INICIA	CCB/MARCA END
6	TIPO MARCA END	TIPO MARCA END
5	LEITURA	FALTOU RELÓGIO
4	ESPERA ÍNDICE	OVERRUN
3	ADM	D
2	D	C
1	C	B
0	B	A

Figura 7 - Registrador de Comando/Estado do formatador

Descrição da função dos bits de ESTADO

- CCB/MARCA END: Indica, durante os estados 3 e 4 do formatador (ver bits A, B, C e D, abaixo), que os bits de relógio estão satisfazendo o padrão de MARCA DE ENDEREÇO. No estado 7 informa o resultado do cálculo do caracter de conferencia de bloco CCB(0=OK , 1=ERRO).
- TIPO MARCA DE ENDEREÇO: Junto com MARCA DE ENDEREÇO, permite distinguir entre os padrões  $C7_{16}$  ou  $D7_{16}$  dos bits de relógio.
- FALTOU RELÓGIO: 0 = os bits de relógio estão OK;  
1 = indica que faltou um bit de relógio no interior de um bloco de dados (estados 4, 5 e 6 do formatador).
- OVERRUN: 0 = atendimento OK;  
1 = notifica que o atendimento não foi oportuno durante uma transferência.
- A, B, C e D: Indicam o estado atual do formatador segundo o quadro definido na figura (8).

Descrição da função dos bits de COMANDO

- INICIA: 0 = leva o Formatador ao estado 0 (zero) indicionalmente;



A	B	C	D	ESTADO DO FORMATADOR
0	0	0	0	ESTADO 0
1	0	0	0	ESTADO 1
1	1	0	0	ESTADO 2
1	1	1	0	ESTADO 3
1	1	1	1	ESTADO 4
0	1	1	1	ESTADO 5
0	0	1	1	ESTADO 6
0	0	0	1	ESTADO 7

Figura 8 - Estados do formatador

1 = na passagem para 1 desencadeia um ciclo do formatador.

TIPO DE MARCA DE

ENDEREÇO: 0 = padrão de relógio será  $D7_{16}$ ;

1 = padrão de relógio será  $C7_{16}$ .

LEITURA: 0 = indica pedido de gravação no disco;

1 = indica pedido de leitura no disco.

ESPERA ÍNDICE: 0 = não sincroniza com sinal de índice;

1 = sincroniza o início de uma operação de transferência com a borda anterior do sinal de ÍNDICE (início de trilha) que vem do disco.

ADM: 0 = pedidos de atendimento durante uma transferência vão para o microprocessador;

1 = pedidos de atendimento durante uma transferência vão para o computador mestre.

D: 0 = não altera bit 3 do registrador de estado;

1 = desliga bit 3 do registrador de estado.

C: 0 = não altera bit 2 do registrador de estado;

1 = desliga bit 2 do registrador de estado.

B: 0 = não altera bit 1 do registrador de estado;

1 = desliga bit 1 do registrador de estado.

Os oito estados do formatador são determinados por um contador em anel de 4 bits representados por A, B, C e D no registrador de estado.

Uma passagem por todos os estados corresponde a um Ciclo do Formatador. A cada estado do formatador corresponde uma atividade, conforme é mostrado na figura (9).

É possível alterar a seqüência normal, mas somente nos estados 4 a 1 e neste sentido, desligando um ou mais dos bits B, C ou D do contador de estados, pela atuação no registrador de comando.

#### II.4.2 Registrador Contador de Palavras

O Contador de Palavras é um registrador de 8 bits que define o tamanho dos blocos a serem transferidos. Seu conteúdo é decrementado de uma unidade a cada byte transferido. O conteúdo deste registrador pode ser lido e carregado por programa.

#### II.4.3 Registrador de Dado

O registrador de dado é, fisicamente, composto de 3 registradores de 8 bits e corresponde a um endereço de memória do microprocessador. Os dados a serem gravados são escritos no registrador temporário de saída, daí passam para o registrador deslocador, de onde são enviados ao disco. Simetricamente, as mesmas operações são realizadas para a leitura, usando um registrador temporário de entrada. Este registrador

ESTADO	LEITURA	GRAVAÇÃO
0	PARADO	PARADO
1	CONTA 16 ZEROS	GRAVA 1 BYTE
2	ESPERA MARCA DE ENDEREÇO	REPETE O BYTE GRA- VADO
3	LÊ MARCA DE END	GRAVA MARCA DE EN- DEREÇO
4	LÊ DADOS	GRAVA DADOS
5	PROCESSA CCB (BYTE 1)	GRAVA CCB (BYTE 1)
6	PROCESSA CCB (BYTE 2)	GRAVA CCB (BYTE 2)
7	ANALISA CCB	CONTINUA GRAVAÇÃO DE DADOS

Figura 9 - Estados e Atividades do formatador

possui dois caminhos de entrada e dois de saída, sendo um para o microprocessador e o outro para o sistema externo (modo ADM).

#### II.4.4 Registrador de Comunicação

Serve para transferência a baixa velocidade entre o microprocessador e o sistema externo. Desta forma é possível transferir mensagens e/ou dados, byte a byte, entre os processadores.

Cada transferência de um byte é sinalizada por um pedido de interrupção.

#### II.4.5 Comunicação Adicional entre o Microprocessador e o Formatador

Muitas vezes é conveniente iniciar-se uma operação e deixar o microprocessador livre para efetuar outras tarefas, retomando o controle em caso de anormalidade ou fim de operação. Para isto existem, além dos registradores descritos anteriormente (que permitem a interação entre o microprocessador e o formatador), mais três linhas de comunicação entre o Formatador e o Microprocessador:

- a. Interrupção não mascarável:

São três as razões que levam o formatador a interromper o microprocessador através de uma interrupção não mascarável:

a.1 - Detecção de marca de endereço na leitura

Neste caso a rotina de atendimento deve verificar se o dado lido satisfaz ao padrão da marca procurada.

a.2 - Contador de palavras com underflow

Quando já foram transferidos todos os bytes solicitados (fim de bloco).

a.3 - Fim de inicialização de trilha

Durante a inicialização de um disco, a gravação começa e termina na borda anterior do sinal de índice (fim ou início da trilha). No estado 7, o controlador grava continuamente um padrão, gerando "intervalos" ("GAPS"). Haverá a interrupção quando for alcançado o fim de trilha.

b. Interrupção mascarável:

A ocorrência de erro (falta de bit de relógio e subatendimento) provoca um pedido de interrupção mas não interferem na seqüência normal do formatador.

c. Byte completo:

O sincronismo de leitura e gravação, em disco, a nível de byte é feito pelo sinal "byte completo". A sistemática para isto, na leitura por exemplo, é a seguinte: o microprocessador lê um byte, armazena na memória e coloca-se em

estado de "parado" ("HALT").

O sinal de byte completo reativa o microprocessador. Então o microprocessador lê mais um byte e repete a operação até o fim do bloco. Para a gravação o processo é similar.

### III. CARACTERÍSTICAS DO SISTEMA

O Sistema descrito neste trabalho (controlador de discos flexíveis e seu "software") se constitui, basicamente, em um canal de entrada/saída inteligente, capaz de executar tarefas de alto nível sobre arquivos armazenados em disques.

O "software" foi projetado para facilitar a utilização do controlador de disco flexíveis em um sistema de computação. Conforme será descrito adiante, o controlador funciona conectado (como "escravo") a um computador "mestre", o qual fica liberado do controle das operações sobre os discos flexíveis ligados ao Controlador.

A coordenação das atividades do Controlador e computador mestre é obtida através da troca de mensagens de controle. As mensagens definem tarefas a serem efetuadas ou indicam condições detectadas durante a execução de tarefas.

O projeto de "software" objetivou capacitar o sistema a:

- controlar vários discos flexíveis;
- gerenciar a utilização simultânea de vários arquivos;
- otimizar o tempo de acesso aos discos;
- executar várias tarefas em paralelo;
- diminuir o envolvimento do processador do computador mestre em operações de E/S;
- facilitar a utilização do sistema em qualquer computador.



Com as características básicas estruturadas, partiu-se para uma definição mais realista, tendo sido impostas certas restrições ao Sistema, já que, não haveria espaço suficiente na memória do controlador para a implementação de todas as rotinas que se tinha em mente.

Desta maneira resolveu-se, em uma primeira versão, implementar o "software" com algumas das limitações impostas pelo Sistema Operacional Básico Multiprogramado definido para o Terminal Autônomo (desenvolvido pelo Curso de Pós-Graduação em Ciência da Computação da UFRGS), computador para o qual o Controlador de Discos Flexíveis será memória de massa.

Deste modo, as definições a seguir apresentadas, levam em conta as necessidades existentes para a utilização do controlador como memória de massa do Terminal Autônomo, as restrições provenientes das próprias limitações do "hardware" disponível no Controlador Inteligente e, finalmente as premissas propostas neste trabalho.

### III.1 Comunicação entre Controlador e Computador Mestre

Toda ordem de execução de tarefas é iniciada pelo computador mestre através do envio de mensagem ao controlador. Para cada mensagem enviada, o "mestre" pode receber até duas mensagens (respostas) do controlador. Estas respostas podem indicar: (1) a existência de problemas pré-operati-

vos, que impedem o início da execução da tarefa, (2) a necessidade de ativação do mecanismo de acesso direto a memória (ADM), (3) a ocorrência de erros pós-operativos, detectados durante a execução da tarefa ou (4) a conclusão da tarefa, com êxito.

As mensagens e respostas respeitam os seguintes formatos:

MENSAGEM:

STX	- caracter "start of text"
<NC>	- número de caracteres do texto
<TEXT0>	- texto (<NC> "bytes")
EOT	- caracter "end of text"

RESPOSTA:

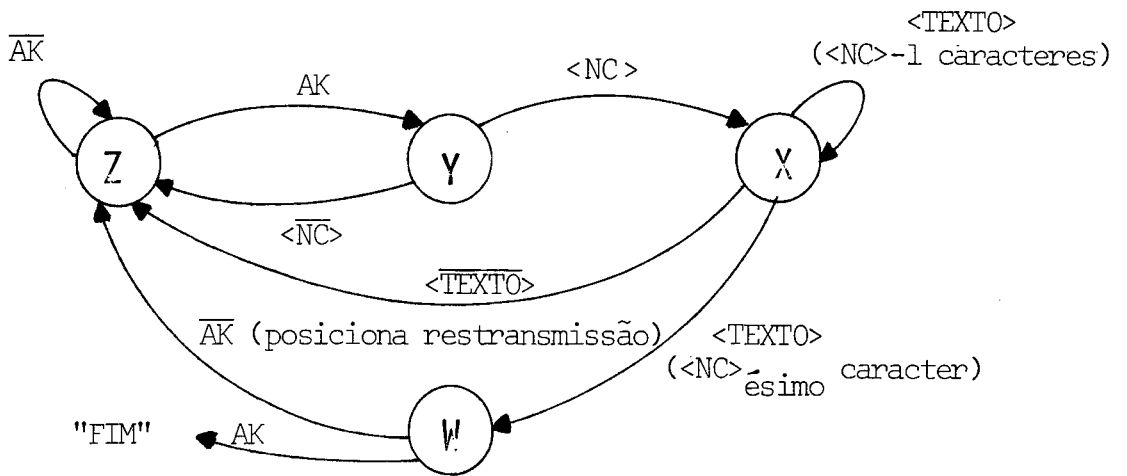
SOH	- caracter "start of header"
<NC>	- número de caracteres de texto
<TEXT0>	- texto (<NC> "bytes")
EOT	- caracter "end of text"

Na comunicação através de mensagens, a todo caracter transmitido corresponde um eco, o qual é o próprio caracter enviado, com exceção dos caracteres STX, SOH e EOT que tem como eco o caracter "acknowledge" (ACK).

Desta maneira, os erros decorrentes da troca de bits na transmissão são captados pelos sistemas, possibilitando a retransmissão das mensagens.

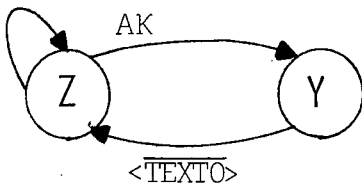
As rotinas de transmissão e recepção de mensagens podem ser descritas pelos autômatos mostrados nas figuras (10) e (11), respectivamente.

Além disso, a rotina de transmissão do computador mestre deve ser tal que, quando receba um caracter SOH como resposta a um STX enviado, desvie automaticamente para a



### NOTAÇÃO

$\overline{AK}$



Z e Y = estados do automato

AK = caracter ACK recebido como eco

$\overline{AK}$  = caracter diferente de ACK recebido como eco

$\overline{\langle TEXTO \rangle}$  = não (caracter do texto enviado)

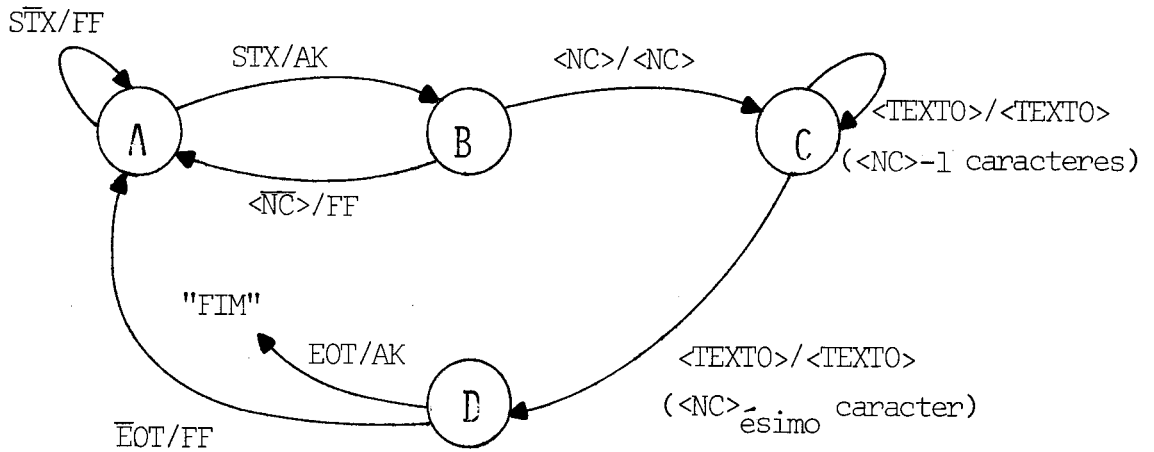
estado Z - envia SOH (para iniciar mensagem, para sincronização ou para retransmissão de mensagem)

estado Y - envia <NC>

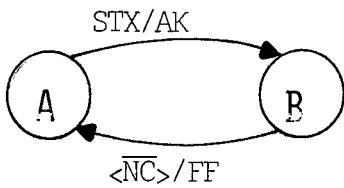
estado X - envia os <NC> caracteres do texto

estado W - envia EOT

Figura 10 - Automato Transmissão



### NOTAÇÃO



A e B □ estados do automato

STX/AK= recepção do caracter STX e envio do caracter AK, como eco.

<NC>/FF= recepção de um valor fora dos limites para número de caracteres do texto e envio de caracter FF, como eco.

estado **A** - espera STX

estado **B** - espera <NC>

estado **C** - espera os <NC> caracteres do texto

estado **D** - espera EOT

Figura 11 - Automato Recepção

rotina de recepção de mensagens originadas pelo controlador.

Esta característica é importante porquanto permite a otimização dos acessos a disco, pois é possível a ocorrência de situações em que os dois sistemas estejam tentando iniciar uma mensagem. Neste caso deve ser dada preferência ao controlador já que a mensagem, oriunda do controlador, pode indicar, por exemplo, que o ADM do computador mestre deva ser preparado para a transferência de um setor já lido ou a ser gravado nos discos flexíveis.

### III.2

#### Textos de Comando

Os textos contidos nas mensagens definem, para o controlador, as tarefas a serem executadas.

Serão apresentados, a seguir, os textos de comando das tarefas que podem ser executadas pelo controlador.

Os campos "indexador do processo solicitante" e "número do processo solicitante", que aparecem no texto de quase todos os comandos, correspondem, respectivamente, aos campos "identificação do processo" (número entre zero e 7) e "número de criação do processo" (valor entre zero e 9999) que servem para definir, de forma unívoca, dentro do Sistema Operacional do Terminal Autônomo (SCHLABITZ & CORSO<sup>2</sup>) os vários processos que são criados durante uma jornada de trabalho do Sistema Operacional.

## III.2.1

## Tarefa abre arquivo

Sua finalidade é abrir um arquivo. O texto para a execução desta tarefa é:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4: CONTERÃO 0 BITS 3, 2, 1 e 0: INDEXADOR DO PROCESSO SOLICITANTE
1	2	NÚMERO DO PROCESSO SOLICITANTE
3	8	NOME DO ARQUIVO
11	2	NÚMERO DO DISCO FLEXÍVEL (VOLUME)
13	2	NÚMERO DE TRILHAS
15	1	MODO DE ABERTURA

## MODO DE ABERTURA:

0 = Indica criação de um arquivo temporário para entrada e saída.

1 = Indica criação de um arquivo permanente para entrada e saída .

2 = Indica a abertura de um arquivo existente para operações de entrada e saída.

3 = indica a abertura de um arquivo existente para operações de entrada apenas.

## NÚMERO DE TRILHAS:

Somente utilizado quando se está criando um arquivo, isto é, para os modos de abertura iguais a 0 e 1.

Ao ser aberto um arquivo, o controlador devolve, no texto da resposta, o índice do arquivo aberto. Este "índice do arquivo", um número entre 0 e 15, servirá para permitir um acesso mais rápido nas próximas operações a serem executadas sobre o arquivo.

## III.2.2

## Tarefa faz Entrada ou Saída

Sua finalidade é executar as operações de entrada ou saída sobre os arquivos. A unidade máxima de transferência de dados corresponde a um setor, ou seja, 128 bytes. Caso existam mais de um registro lógico em um setor, função do fator de bloco, é responsabilidade do "software" do computador mestre dar o tratamento adequado.

O texto para a execução desta tarefa é:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4: CONTERÃO 1 BITS 3, 2, 1 e 0: INDEXADOR DO PROCESSO SOLICITANTE
1	2	NÚMERO DO PROCESSO SOLICITANTE
3	1	ÍNDICE DO ARQUIVO (definido quando da abertura do arquivo)
4	2	NÚMERO DO SETOR (relativo ao início do arquivo)
6	2	NÚMERO DE BYTES A TRANSFERIR (máximo de 128)
8	1	MODO: 0 - LEITURA DO DISCO 1 - GRAVAÇÃO NO DISCO



## III.2.3

## Tarefa fecha arquivo

Sua finalidade é fechar um arquivo que anteriormente tenha sido aberto. O texto para a execução desta tarefa é:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4: VALOR 2 BITS 3, 2, 1 e 0: INDEXADOR DO PROCESSO SOLICITANTE
1	2	NÚMERO DO PROCESSO SOLICITANTE
3	1	ÍNDICE DO ARQUIVO
4	2	MODO DE FECHAMENTO

## MODO DE FECHAMENTO:

0 = O arquivo a ser fechado pode ou não permanecer no disco. Se o modo de abertura foi para arquivo temporário, ele será removido, caso contrário será mantido.

1 = O arquivo será removido, a menos que ele esteja sendo usado por outro processo.

Convém salientar aqui que para se remover um arquivo é necessário, primeiro, que o mesmo tenha sido aberto.

## III.2.4

## Tarefa remove processos

Esta tarefa foi implementada por solicitação do grupo que desenvolveu o Sistema Operacional Básico Multiprogramado para o Terminal Autônomo. O Sistema Operacional permite que um processo remova a si próprio (auto-remoção) ou a qualquer de seus filhos e, para qualquer processo removido, o Sistema remove automaticamente todos os descendentes desse processo. Neste caso os recursos que estiverem sendo usados por um processo a ser removido e pelos seus descendentes terão que ser liberados. Para o controlador de discos flexíveis esta tarefa nada mais é que fechar os arquivos, de acordo com o modo de abertura, que estavam sendo utilizados pelos processos discriminados no texto da mensagem definida a seguir:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4: VALOR 3 BITS 3, 2, 1 e 0: INDEXADOR DO PROCESSO SOLICITANTE
1	1	NÚMERO DE PROCESSOS A REMOVER (ENTRE 1 E 8)
2	1	PROCESSO <i>i</i>
3	1	PROCESSO <i>j</i>
4	1	PROCESSO <i>k</i>
5	1	PROCESSO <i>l</i>
6	1	PROCESSO <i>m</i>
7	1	PROCESSO <i>n</i>
8	1	PROCESSO <i>o</i>
9	1	PROCESSO <i>p</i>

INDEXADORES DOS PROCESSOS PARA OS QUAIS DEVERÃO SER FECHADOS OS ARQUIVOS

No caso de não haver nenhum arquivo aberto para os processos relacionados no texto, não será executada nenhuma operação.

## III.2.5

## Tarefa carrega módulo

Esta tarefa também foi implementada por solicitação do grupo que desenvolveu o Sistema Operacional Básico Multiprogramado para o Terminal Autônomo. Ela tem a finalidade de ler um arquivo com código objeto e é composta por três subtarefas, conforme será explicado adiante.

Arquivos com código objeto possuem registros lógicos de 64 bytes e são divididos em três partes (formato definido no projeto do Assembler e Ligador - Carregador de Programas para o Terminal Autônomo - SCHLABITZ & CORSO<sup>2</sup>). A primeira parte (registro zero) contém a identificação do código objeto (nome, tamanho do código, indicação de relocabilidade e tamanho da tabela de relocação), a segunda parte contém o código objeto propriamente dito (cada registro contendo um endereço de carga, um número N, menor que 62, e N bytes com código a ser carregado) e a terceira parte contém a tabela de relocabilidade (cada registro contendo, no máximo, a indicação de 32 endereços a serem relocados).

Para facilitar o processo de carga de código objeto no computador mestre, esta tarefa foi dividida em três subtarefas, cada uma identificada por um código diferente. Deste modo, cada subtarefa ficou sendo responsável pela execução de uma das três fases da tarefa de carga do código objeto: leitura do registro de identificação do código objeto, leitura do código objeto e leitura da tabela de relocabilidade.

O texto para execução de uma das três subtarefas é o seguinte:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4: =4 SUBTAREFA 1 =5 SUBTAREFA 2 =6 SUBTAREFA 3  BITS 3, 2, 1 e 0: INDEXADOR DO PROCESSO SOLICITANTE
O restante dos bytes da mensagem segue a mesma forma do texto da mensagem para a tarefa "FAZ ENTRADA OU SAÍDA".		

## III.2.6

## Tarefa inicializa disco

Esta tarefa, como diz seu nome, tem a finalidade de inicializar um disco flexível com formato e dados de controle para sua utilização com o "software" aqui apresentado ou com equipamentos IBM. Este formato será apresentado em detalhes no item III.4.

O texto da mensagem que solicita esta tarefa é apresentado abaixo:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4 = VALOR 7 BITS 3, 2, 1 e 0 = INDEXADOR DO PROCESSO SOLICITANTE
1	2	NÚMERO DO VOLUME <input type="checkbox"/> 0 SE TIPO DO FORMATO FOR ZERO <input type="checkbox"/> ≠ 0 SE TIPO DO FORMATO FOR UM
3	1	TIPO DE FORMATO <input type="checkbox"/> 0 FORMATO IBM PADRÃO <input type="checkbox"/> 1 FORMATO PARA ESTE SOFTWARE
4	1	TIPO DE INICIALIZAÇÃO <input type="checkbox"/> 0 = SÓ TRILHA ZERO (REMOVER ARQUIVOS) <input type="checkbox"/> 1 = TODO O DISCO
5	1	NÚMERO DA UNIDADE EM QUE ESTÁ O DISCO

## III.3

## Texto de resposta

À toda tarefa recebida pelo controlador, através de uma mensagem, deve necessariamente corresponder uma

resposta, nos mesmos moldes, para o computador mestre.

O texto de resposta pode indicar, para o mestre, situações do tipo:

- erros pré-operativos;
- ativação de ADM;
- erros pós-operativos;
- execução de tarefa concluída com sucesso.

A tarefa "Faz entrada ou saída" é a única que necessita do envio de duas mensagens de resposta. Uma relativa a erros pré-operativos ou ativação de ADM e outra relativa a erros pós-operativos ou conclusão da tarefa.

O texto de resposta, em qualquer caso, tem a seguinte estrutura:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4: CÓDIGO DA TAREFA BITS 3, 2, 1 e 0: INDEXADOR DO PROCESSO SOLICITANTE
1	1	BIT 7: LIGADO - INDICA ERRO DESLIGADO - INDICA EXECUÇÃO NORMAL BIT 6: LIGADO - INDICA FASE PRÉ-OPERATIVA (*) DESLIGADO - INDICA FASE PÓS-OPERATIVA BIT 5: LIGADO - INDICA ERRO DO SOFTWARE DO CONTROLADOR DESLIGADO - TUDO OK. BITS 4, 3, 2, 1 e 0: FORNECEM O CÓDIGO DO ERRO SE BIT 7 LIGADO. NA TAREFA ABRE ARQUIVO FORNECE ÍNDICE DO ARQUIVO ABERTO SE BIT 7 DESLIGADO.

"\*": O BIT 6 LIGADO só é utilizado nas tarefas "FAZ ENTRADA OU SAÍDA" e "CARREGA MÓDULO".

### III.4 Escolha do formato para o disco

Serão apresentados, neste ítem, o formato utilizado pela IBM, o formato adotado para este trabalho e os motivos que determinaram esta escolha.

#### III.4.1 Formato IBM

Este formato é adotado nas máquinas IBM da série 3740.

Todas as 77 trilhas, numeradas de 0 à 76, possuem o mesmo "lay-out". A trilha zero é reservada para informações de controle e as demais para armazenar os dados.

A figura (12) mostra o formato de trilha e define os seus diversos campos e tamanhos, os quais, terão seus conteúdos apresentados a seguir:

ESPAÇO 1:	SINCRONIZAÇÃO	
	relógio: FF	valor: 00
ME 1:	MARCA DE ENDEREÇO (ÍNDICE)	
	relógio: D7	valor: FC
ESPAÇO 2:	SINCRONIZAÇÃO	
	relógio: FF	valor: 00
ME 2:	MARCA DE ENDEREÇO (SETOR)	
	relógio: C7	valor: FE

IDENTIFICAÇÃO: composto pelos campos:



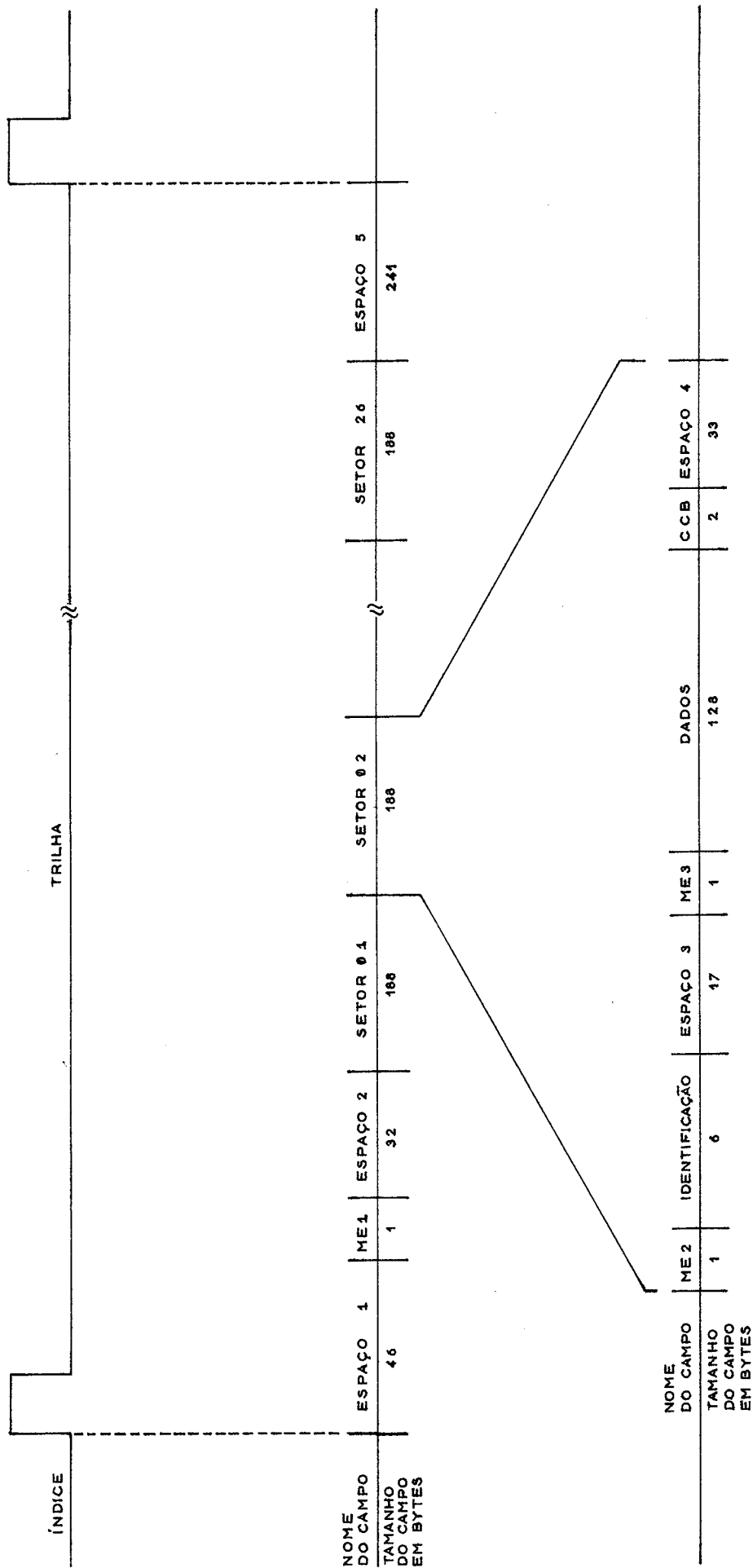


Figura 12 - "Lay-out" de trilha (Formato IBM)

## - NÚMERO DA TRILHA

relógio: FF            valor: &lt;número de trilha&gt;

## - ESPAÇO

relógio: FF            valor: 00

## - NÚMERO DO SETOR

relógio: FF            valor: &lt;número do setor&gt;

## - ESPAÇO

relógio: FF            valor: 00

## - CCB (CARÁTER DE CONFERÊNCIA DE BLOCO)

relógio: FF            valor: &lt;calculado&gt;

## ESPAÇO 3: SINCRONIZAÇÃO

relógio: FF            valor: 00

## ME 3: MARCA DE ENDEREÇO (DADOS)

relógio: C7            valor            FB - dado OK  
F8 - dado apagadoDADOS: detalhados nos itens III.4.1.1, III.4.1.2 e  
III.4.2.

## CCB: CARÁTER DE CONFERÊNCIA DE BLOCO

relógio: FF            valor: &lt;calculado&gt;

## ESPAÇO 4: SINCRONIZAÇÃO

relógio: FF            valor: 00

## ESPAÇO 5: SINCRONIZAÇÃO

relógio: FF            valor: 00

O conteúdo de inicialização do campo "DADOS" é igual para todos os setores com exceção dos existentes na trilha zero. Devido a esta característica, este campo será apresentado em separado.

### III.4.1.1 Campo "DADOS" em setores da trilha zero

A diferença de inicialização deste campo, para os setores da trilha zero, resulta de sua utilização como controle de funcionamento (advém, daí, inicializações diferentes mesmo dentro da trilha).

A seguir serão descritos os conteúdos, de inicialização, destes campos em relação aos 26 setores da trilha zero.

SETORES 01 ATÉ 04:

O "lay-out" está apresentado na figura (13). Estes quatro setores não são utilizados.

X.1 - Inicializado com brancos em EBCDIC.

X.2 - Inicializado com zeros binários.

SETOR 05:

Seu "lay-out" está apresentado na figura (14). O campo de "Dados" deste setor é destinado ao controle das trilhas defeituosas do disco. Até duas trilhas defeituosas podem ser indicadas neste campo.

5.1 - Identifica a função do campo como Mapa de Erros em trilhas. É inicializado com o valor "ERMAP" em EBCDIC.

5.2 - Separador de campo. É inicializado com valor 40 em EBCDIC (branco).

5.3 - Indicador da primeira trilha defeituosa. Contém o valor da trilha em EBCDIC. Inicialmente recebe brancos.

5.4 - Campo Zero. Deverá conter o valor binário zero se existe uma primeira trilha defeituosa. É inicializado com branco.

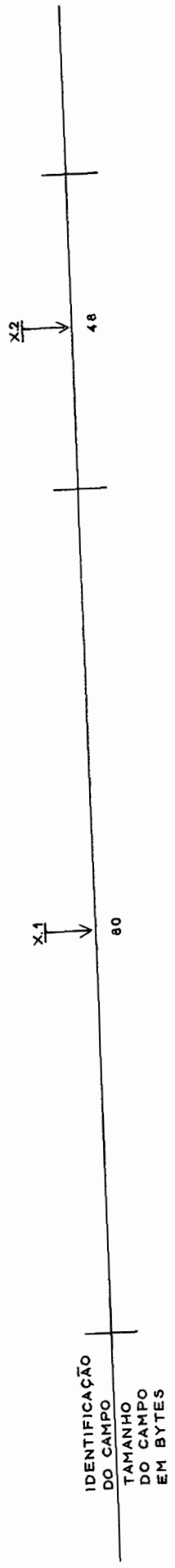


Figura 13 - Campo de "Dados" geral

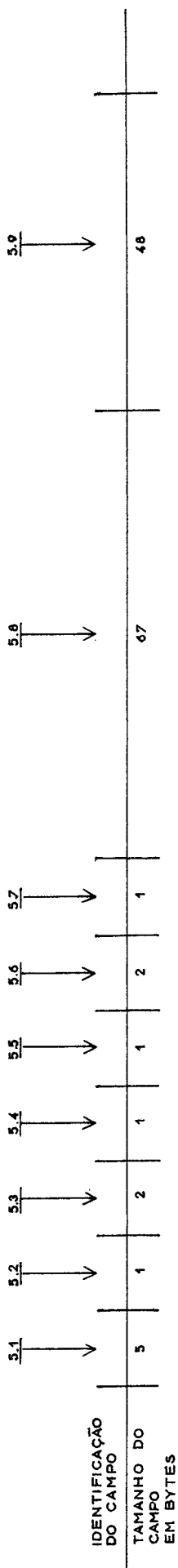


Figura 14 - Campo de "Dados" do setor 5 da trilha zero.

- 5.5 - Separador de Campo. É inicializado com branco.
- 5.6 - Indicador da segunda trilha defeituosa. Contém o valor da trilha em EBCDIC. Inicialmente contém brancos.
- 5.7 - Campo Zero. Deverá conter zero binário se existe a segunda trilha defeituosa. É inicializado com branco.
- 5.8 - Campo não usado. É inicializado com brancos.
- 5.9 - Campo não usado. É inicializado com zeros binários.

SETOR 06:

O "lay-out" está apresentado na figura (13). Este setor não é utilizado. A inicialização é idêntica a dos setores 01 a 04.

SETOR 07:

Seu "lay-out" está apresentado na figura (15). O campo de "dados" deste setor é destinado a identificação do disco e manutenção de alguns parâmetros de controle.

- 7.1 - Identifica a função do campo como Descritor do Disco. É inicializado com o valor "VOL1" em EBCDIC.
- 7.2 - Identificador do Disco. Este campo permite identificar um disco através de um nome. É inicializado com "IBMIRD" em EBCDIC mas pode ser alterado pelo usuário.
- 7.3 - Acessabilidade. Este campo contendo qualquer valor diferente de branco, indica que o disco não poderá ser acessado. É inicializado com branco.
- 7.4 - Campo não usado. É inicializado com brancos.
- 7.5 - Seqüência lógica de setores. Este campo informa a seqüência dos registros nos setores de uma trilha, para aplicações especiais. As seqüências possíveis estão mostradas na figura (16). É inicializado com branco.

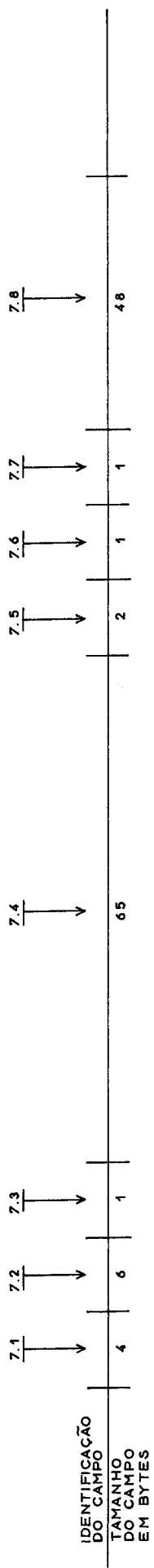


Figura 15 - Campo de "Dados" do setor 7 da trilha zero.

VALOR DO CAM- PO "SEQUÊN- CIA LÓGICA DE SETORES"	REGISTROS A GRAVAR EM UMA TRILHA																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
BRANCO ou 01	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
02	1	3	5	7	9	11	13	15	17	19	21	23	25	2	4	6	8	10	12	14	16	18	20	22	24	26
03	1	4	7	10	13	16	19	22	25	2	5	8	11	14	17	20	23	26	3	6	9	12	15	18	21	24
04	1	5	9	13	17	21	25	2	6	10	14	18	22	26	3	7	11	15	19	23	4	8	12	16	20	24
05	1	6	11	16	21	26	2	7	12	17	22	3	8	13	18	23	4	9	14	19	24	5	10	15	20	25
06	1	7	13	19	25	2	8	14	20	26	3	9	15	21	4	10	16	22	5	11	17	23	6	12	18	24
07	1	8	15	22	2	9	16	23	3	10	17	24	4	11	18	25	5	12	19	26	6	13	20	7	14	21
08	1	9	17	25	2	10	18	26	3	11	19	4	12	20	5	13	21	6	14	22	7	15	23	8	16	24
09	1	10	19	2	11	20	3	12	21	4	13	22	5	14	23	6	15	24	7	16	25	8	17	26	9	18
10	1	11	21	2	12	22	3	13	23	4	14	24	5	15	25	6	16	26	7	17	8	18	9	19	10	20
11	1	12	23	2	13	24	3	14	25	4	15	26	5	16	6	17	7	18	8	19	9	20	10	21	11	22
12	1	13	25	2	14	26	3	15	4	16	5	17	6	18	7	19	8	20	9	21	10	22	11	23	12	24
13	1	14	2	15	3	16	4	17	5	18	6	19	7	20	8	21	9	22	10	23	11	24	12	25	13	26

Figura 16 - Sequência de registros nos setores de uma trilha



- 7.6 - Campo não usado. É inicializado com brancos.
- 7.7 - Campo constante. É inicializado com "W" em EBCDIC.
- 7.8 - Campo não usado. É inicializado com zeros binários.

SETOR 08 ATÉ 26:

Seu "lay-out" está apresentado na figura (17). Os campos de dados destes setores são destinados ao armazenamento dos descritores dos arquivos gravados no disco (formam o diretório do disco). São 19 setores disponíveis, limitando, com isso, em 19, o número de arquivos que podem ser gravados. Os conteúdos dos campos serão apresentados a seguir:

- C.1 - Identifica a função do campo como descritor de arquivos. Para o setor 8 é inicializado com "HDR1" em EBCDIC. Os setores de 9 a 26 são inicializados com "DDR1" em EBCDIC para representar arquivos apagados.
- C.2 - Campo não usado. Inicializado com branco.
- C.3 - Identifica o nome do arquivo. Deve ser definido pelo usuário mas os seus 8 bytes são inicializados como segue:  
Setor 8: nos 4 primeiros bytes, "DATA" em EBCDIC e nos 4 restantes, brancos.  
Setores 9-26: nos 4 primeiros bytes, "DATA" em EBCDIC, nos dois seguintes, o número do setor em EBCDIC e nos dois últimos, brancos.
- C.4 - Campo não usado. Inicializado com brancos.
- C.5 - Tamanho dos registros. Inicializado com 080 em EBCDIC. Tamanho máximo permitido é 128.
- C.6 - Campo não usado. Inicializado com branco.
- C.7 - Início da Área Reservada para o Arquivo. Identifica a trilha e setor onde se encontra o início da área do arquivo. Os dois primeiros bytes fornecem o número de tri

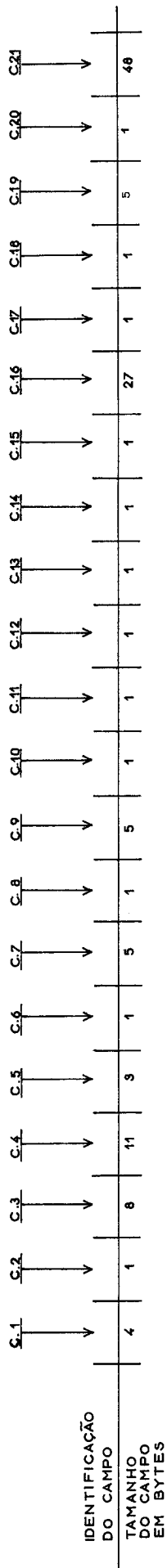


Figura 17 - Campo de "Dados" dos setores 08-26 da trilha zero.

lha, o terceiro byte deve conter zero binário e os dois últimos bytes fornecem o número do setor. No setor 8 este campo é inicializado com "01001" e nos setores 9 a 26 com "74001".

- C.8 - Campo não utilizado. Inicializado com branco.
- C.9 - Final da Área Reservada para o Arquivo. Identifica a trilha e setor onde termina a área do arquivo. A subdivisão do campo tem a mesma estrutura do campo "Início da Área reservada para o Arquivo" (C.7). É inicializado com "73026" em EBCDIC.
- C.10- Campo não usado. Inicializado com branco.
- C.11- Ignorar o arquivo. Se este campo contém um "B" em EBCDIC, o arquivo é ignorado. É inicializado com branco.
- C.12- Acessabilidade. Este campo deverá conter um branco para que o arquivo seja processado. Inicializado com branco.
- C.13- Proteção de Gravação. Se este campo contém um "P" em EBCDIC, este arquivo está protegido contra gravações. Inicializado com branco.
- C.14- Campo não usado. Inicializado com branco.
- C.15- Indicador de Multivolume. Um branco indica que este arquivo não é continuação nem continua em outro disco. Um "C" indica que o arquivo continua em outro disco e um "L" indica que este disco é o último que contém o arquivo. É inicializado com branco.
- C.16- Campo não usado. Inicializado com brancos.
- C.17- Marca de Conferência. Um "V" neste campo indica que o arquivo já foi conferido. Inicializado com branco.
- C.18- Campo não usado. Inicializado com branco.

- C.19- Final do Arquivo. Indica a primeira trilha e setor não utilizados pelo arquivo (fornecendo o fim lógico do arquivo). A subdivisão do campo é idêntica a do campo Início da Área Reservada para o Arquivo (C.7). No setor 8 é inicializado com "01001" e nos setores 9-26 com "74001".
- C.20- Campo não usado. Inicializado com brancos.
- C.21- Campo não usado. Inicializado com zeros binários.

III.4.1.2 Campo "DADOS" em setores das trilhas  
01-76

Todos os campos "DADOS" dos 26 setores de todas as outras trilhas (não zero) são inicializados da mesma maneira. Isto porque estes são os campos nos quais são armazenados os dados dos arquivos. O "lay-out" deste campo é mostrado na figura (13).

As inicializações efetuadas são:

- X.1 - Inicializado com "V" em EBCDIC.
- X.2 - Inicializado com "V" em EBCDIC.

III.4.2 O formato escolhido

Analisando a utilização de discos flexíveis, constata-se que os mesmos são utilizados, principalmente, para entrada de dados e que a grande maioria das máquinas que

trabalham com discos flexíveis são IBM. Portanto, seria interessante que o formato escolhido para este trabalho fosse com patível com o formato IBM.

Partindo-se dessa idéia, foi verificada a possibilidade de se definir um formato que permitisse a utili zação dos disquetes em qualquer dos dois sistemas. Constatou-se que, com alguns cuidados extras, isto seria possível. Como a utilização de um formato compatível aumenta a potenciali dade de utilização dos discos flexíveis (permitindo, inclusive, que disquetes gravados neste formato sejam convertidos pa ra fita magnética em unidades IBM), resolveu-se adotar, neste trabalho, o formato compatível que será descrito a seguir.

Usando a definição IBM, foi elaborada uma adaptação que, pela não utilização de alguns campos daquele formato, torna os disquetes perfeitamente utilizáveis, quer por este "software" quer por máquinas IBM.

Como no formato IBM, utiliza-se a trilha zero para informações de controle e as demais trilhas para ar mazenamento de dados.

Para simplificar, a seguir serão considerada s somente as modificações introduzidas em relação ao formato IBM. Essas modificações são relativas, apenas, aos campos de "dados" dos setores da trilha zero.

SETOR 05:

Ao invés de apenas duas trilhas com proble mas, pode-se armazenar até 18 trilhas defeituosas. Neste caso a filosofia de funcionamento é diferente. Sempre que se qui zer salvar uma trilha que apresente problemas, isto terá que ser feito, por solicitação do computador mestre, através de

uma tarefa. Neste caso esta trilha não será mais alocada para nenhum arquivo, O novo "lay-out" deste setor está mostrado na figura (18).

SETOR 07:

O "lay-out" está apresentado na figura (15).

7.2 - Identificador do Disco. Os quatro primeiros bytes recebem "PCFD" (Programa Controlador de Floppy Disks) em EBCDIC e os dois bytes restantes receberão o número do volume, em binário.

7.3 - Campo não usado. Inicializado com brancos.

7.5 - Campo não usado. Inicializado com brancos.

SETOR 08-26:

O "lay-out" está apresentado na figura (17).

C.1 - Na inicialização ou na remoção do arquivo descrito neste campo, recebe "DDR1". Na criação de um arquivo, recebe "HDR1".

C.5 - Inicializado com 080. Na criação de um arquivo, é alterado para 128.

C.7 - O número do setor não é utilizado, pois assume-se que o arquivo sempre comece no setor 01 de uma determinada trilha. É inicializado como no formato IBM.

C.9 - O número do setor não é utilizado pois assume-se que a área do arquivo termine no setor 26 de uma determinada trilha. É inicializado como no formato IBM.

C.11- Campo não utilizado. Inicializado com branco.

C.12- Campo não utilizado. Inicializado com branco.

C.13- Campo não utilizado. Inicializado com branco.

C.15- Campo não utilizado. Inicializado com branco.



C.17- Campo não utilizado. Inicializado com branco.

C.19- Este campo é inicializado no formato IBM. Na criação de um arquivo, seu valor é: número do setor = 01 e número da trilha = número da trilha existente em C.9 mais 1. Isto é feito para permitir a compatibilidade.

### III.4.3

### Comentários

Com as modificações introduzidas, arquivos gravados em máquinas IBM podem ser lidos por este "software" e vice-versa, desde que sejam respeitadas as seguintes limitações:

- o disco não poderá ter trilhas defeituosas;
- alguns campos utilizados no formato IBM, mas que não são utilizados por este "software", logicamente, não serão considerados por este "software";
- todo arquivo deverá estar armazenado em apenas um disco e, além disso, deverá iniciar sempre no setor 1 de uma trilha qualquer e ter seus registros constituídos por 128 bytes;
- ao converter um arquivo gravado por este "software" para fita magnética, em máquinas da IBM, esta fita conterá todos os registros do arquivo até o final da área reservada para o mesmo, independentemente do registro existir logicamente ou não.

Sugere-se, para maior segurança, que qualquer disco que tenha que ser utilizado para processar um arquivo gravado em outra máquina, possua as seguintes característica



cas:

- seja inicializado;
- receba apenas um arquivo.

Isto se deve ao fato de que, se um disco é muito utilizado (por exemplo, se possui muitos arquivos) o mesmo pode conter em um campo qualquer, não utilizado ou utilizado de maneira diferente por este "software", um valor que determine um processamento totalmente errado deste arquivo. Um exemplo de campo utilizado de maneira diferente, por este "software", é o mapa de trilhas defeituosas no setor 5 da trilha zero.

### III.5

#### Gerenciamento dos arquivos

Para fazer o gerenciamento dos arquivos nos discos flexíveis, foram criadas as seguintes estruturas de dados:

- cópia dos diretórios dos discos;
- descritores dos arquivos em uso;
- vetores de ocupação das trilhas dos discos;
- vetores de ocupação dos descritores de arquivos na trilha zero de cada disco.

#### III.5.1

##### Definição das estruturas

As estruturas de dados mantidas na memória

do controlador armazenam de forma eficiente as informações necessárias ao gerenciamento de arquivos, permitindo rápido acesso a essas informações.

### III.5.1.1 Cópia dos diretórios

Para cada disco ligado ao sistema, é montada, na memória, uma cópia de seu diretório. O resultado é uma tabela, onde cada entrada contém as seguintes informações:

- NOME DO ARQUIVO (8 BYTES)
- TRILHA INICIAL DO ARQUIVO (1 BYTE)
- TRILHA FINAL DO ARQUIVO (1 BYTE)
- SETOR FINAL DO ARQUIVO (1 BYTE)
- SETOR ONDE ESTÁ O DESCRITOR NO DIRETÓRIO (1 BYTE)
  - = zero (para arquivos temporários)
  - ≠ zero (indica o setor do disco onde está ou será armazenado o seu descritor) ( $1 \leq \text{SETOR} \leq 19$ )
- NÚMERO DE PROCESSOS QUE ABRIRAM O ARQUIVO (5 BITS)
- EXISTÊNCIA DO DESCRITOR EM DISCO (1 BIT)
  - = 0 descritor não existe no disco
  - = 1 descritor existente no disco
- TIPO DO ARQUIVO (1 BIT)
  - = 0 arquivo permanente
  - = 1 arquivo temporário
- REMOÇÃO DO DESCRITOR (1 BIT)
  - = 0 remoção não prevista

= 1 remoção em andamento

- APONTADOR PARA A PRÓXIMA ENTRADA NA CÓPIA DO DIRETÓRIO (2 BYTES)

Esta tabela, cujo tamanho foi limitado, nesta implementação, em 100 entradas, é compartilhada por todos os discos flexíveis a serem controlados. O número de entradas poderá ser aumentado de acordo com a necessidade do usuário e disponibilidade de área de memória.

### III.5.1.2 Descritores dos arquivos em uso

Quando um processo solicita a abertura de um arquivo, cria-se um descritor, na memória, para este arquivo, que contém:

- NÚMERO DO PROCESSO (2 BYTES)
- ABERTURA E INDEXADOR
  - MODO DE ABERTURA (2 BITS)
  - INDEXADOR DO PROCESSO (4 BITS)
- UNIDADE EM QUE ESTÁ O ARQUIVO (1 BYTE)
- APONTADOR PARA A ENTRADA NA CÓPIA DO DIRETÓRIO DO DISCO (2 BYTES)

O limite de descritores de arquivos em uso é 16 (máximo número de arquivos que podem ser abertos concomitantemente).

O "modo de abertura" poderá assumir os seguintes valores:

- $\emptyset\emptyset_2$  - CRIAÇÃO DE UM ARQUIVO TEMPORÁRIO PARA ENTRADA E SAÍDA;
- $\emptyset 1_2$  - CRIAÇÃO DE UM ARQUIVO PERMANENTE PARA ENTRADA E SAÍDA;

- 10<sub>2</sub>- ABERTURA DE UM ARQUIVO EXISTENTE PARA OPERAÇÕES DE ENTRADA E SAÍDA;
- 11<sub>2</sub>- ABERTURA DE UM ARQUIVO EXISTENTE PARA OPERAÇÕES DE ENTRADA APENAS.

### III.5.1.3 Vetores de ocupação de trilhas

Para cada disco ligado ao sistema é montado um vetor de 76 bits (assume-se que a trilha zero estará sempre ocupada) correspondentes a cada uma das trilhas do disco, onde cada bit tem o seguinte significado:

valor	significado
0	trilha livre
1	trilha ocupada

### III.5.1.4 Vetores de ocupação de descritores

Para cada disco ligado ao sistema é montado um vetor de 19 bits (correspondentes aos setores disponíveis para armazenamento de descritores de arquivos, no diretório do disco) onde cada bit tem o seguinte significado:

valor	significado
0	posição livre
1	posição ocupada

## III.5.2

## Funcionamento

Quando o controlador sente, através de uma interrupção, que um disco flexível foi ligado ao sistema (estado de pronto), é gerada uma mensagem para execução da tarefa (no caso, uma tarefa interna) denominada "monta diretório do disco".

Deste modo a trilha zero é lida, sendo montada, na memória, uma cópia do diretório. Convém salientar que, nesta ocasião, só existirão, na cópia, os arquivos permanentes e que os valores assumidos, para cada item, serão:

- NOME DO ARQUIVO ▣ <nome do arquivo>
- TRILHA INICIAL DO ARQUIVO = <trilha inicial> (\*)
- TRILHA FINAL DO ARQUIVO ▣ <trilha final> (\*\*)
- SETOR FINAL DO ARQUIVO = <setor final> (\*\*)
- NÚMERO DO SETOR DO DESCRITOR ▣ <número do setor>
- NÚMERO DE PROCESSOS QUE O ABRIRAM ▣ zero
- EXISTÊNCIA DO DESCRITOR EM DISCO = um (existe no disco)

---

(\*) É carregado com o valor existente em "número de trilha" do campo C.7 (ver figura (17)) que contém o Início da Área Reservada para o Arquivo.

(\*\*) Para carregá-lo utiliza-se os campos "número da trilha" e "número do setor" do campo C.19 (ver figura (17)), que indica o final do arquivo, da seguinte forma:

"número do setor" ← "número do setor" - 1;

decremento ← 0

se "número do setor" = 0 faz-se: "número do setor" ← 26 e  
decremento ← 1;

"número da trilha" ← "número da trilha" - decremento;

TRILHA FINAL DO ARQUIVO ← "número da trilha";

SETOR FINAL DO ARQUIVO ← "número do setor".

Desta maneira os arquivos gravados por este "software" ficam compatíveis com os gravados por máquinas IBM.

- TIPO DO ARQUIVO NO DISCO = zero (arquivo permanente)
- REMOÇÃO DO DESCRITOR = zero (remoção não prevista)
- APONTA A PRÓXIMA ENTRADA NA CÓPIA = <apontador>

Durante a montagem do diretório do disco, será criado, também, o vetor de ocupação de trilhas e o vetor de ocupação de descritores, correspondentes a este disco.

### III.5.2.1 Abertura de um arquivo

Durante a abertura de um arquivo, duas situações devem ser consideradas:

a) o arquivo será criado

Neste caso deverá ser verificada a existência de espaço em disco, através do vetor de ocupação de trilhas e, para arquivo permanente, a existência de setor vago para armazenar seu descritor, através do vetor de ocupação de descritores. Caso não haja nenhuma condição que inviabilize a abertura do arquivo, as seguintes operações devem ser efetuadas:

- ocupação de espaço em disco, através do vetor de ocupação de trilhas;
- se arquivo a ser criado é permanente, ocupação do setor para armazenar seu descritor no diretório, através do vetor de ocupação de descritores;
- criação de uma entrada na cópia do diretório do disco contendo:

- NOME DO ARQUIVO = <nome do arquivo>

- TRILHA INICIAL DO ARQUIVO = <trilha inicial>
- TRILHA FINAL DO ARQUIVO ▣ <trilha final>
- SETOR FINAL DO ARQUIVO = 26
- NÚMERO DO SETOR DO DESCRITOR:
  - ▣ zero, se criação de arquivo temporário;
  - = número do setor alocado para armazenar seu descritor quando o arquivo for fechado (número entre 1 e 19), se criação de arquivo permanente;
- NÚMERO DE PROCESSOS QUE ABRIRAM O ARQUIVO = um
- EXISTÊNCIA DO SEU DESCRITOR EM DISCO = zero
- REMOÇÃO DO DESCRITOR ▣ zero
- TIPO DO ARQUIVO NO DISCO:
  - ∅ - se modo de abertura for "∅1"
  - 1 - se modo de abertura for "∅∅"
- APONTADOR PARA A PRÓXIMA ENTRADA NA CÓPIA = <apontador>
- criação de uma entrada nos descritores de arquivos em uso contendo:
  - NÚMERO DO PROCESSO = <número do processo>
  - MODO DE ABERTURA = <modo de abertura>
  - INDEXADOR DO PROCESSO = <indexador>
  - UNIDADE EM QUE ESTÁ O ARQUIVO ▣ <unidade>
  - APONTADOR PARA ENTRADA NA CÓPIA DO DIRETÓRIO = <apontador>

b) abertura de um arquivo já existente

- Caso não haja nenhuma condição que inviabilize a operação, serão executados os seguintes procedimentos:
- criação de uma entrada nos descritores de arquivos em uso contendo os dados relativos ao arquivo e o modo de abertura solicitado;
  - adição de uma unidade no campo "número de processos que abri

ram o arquivo" da entrada (correspondente ao arquivo) da cópia do diretório do disco.

Ao ser aberto o arquivo, o controlador devolve ao computador mestre o índice do arquivo aberto (entre 0 e 15) que servirá para permitir um acesso mais rápido nas próximas operações que forem efetuadas sobre o arquivo. Este índice indica a posição, nos descritores dos arquivos em uso, da entrada correspondente ao arquivo.

Qualquer tentativa de abertura de arquivo que tenha as condições:

- EXISTÊNCIA DE SEU DESCRITOR EM DISCO = 0 (não existe)

ou

- TIPO DO ARQUIVO NO DISCO = 1 (temporário)

será sinalizada ao computador mestre como um erro do tipo "arquivo não existente".

### III.5.2.2 Fechamento de um arquivo

Quando um processo, que mantinha um arquivo aberto, solicita seu fechamento, é executado o seguinte procedimento:

- a) remove-se sua entrada nos descritores de arquivos em uso;
- b) decrementa-se de uma unidade o campo "número de processos que abriram o arquivo" da entrada, correspondente ao arquivo, da cópia do diretório do disco;
- c) se "modo de fechamento" for igual a 1 (fechamento com remoção), faz-se "tipo do arquivo", em sua entrada na cópia do



diretório, igual a 1 (arquivo temporário). Se a situação anterior do "tipo do arquivo" era "zero" (arquivo permanente) e o campo "existência de seu descritor em disco" indica "zero" (descritor não está gravado no diretório do disco), libera-se o setor alocado para armazenar seu descritor, no vetor de ocupação de descritores, e faz-se o campo "setor onde está o descritor no diretório", na entrada de cópia do diretório, igual a "zero";

d) se "tipo do arquivo no disco" for igual a 1 (arquivo temporário) e "existência de seu descritor" não for igual a "zero" (descritor do arquivo está gravado no diretório do disco), faz-se:

d.1) se "remoção do descritor" igual a um (remoção em andamento) o fechamento é encerrado avisando o computador mestre que o arquivo foi fechado;

d.2) se "remoção do descritor" igual a "zero" (remoção não prevista) faz-se

- "remoção do descritor" = 1;
- apaga-se seu descritor do disco (através de acesso ao setor correspondente da trilha zero);
- atualiza-se o vetor de ocupação de descritores, liberando o espaço correspondente;
- "existência de seu descritor" = zero;
- "remoção do descritor" = zero;
- "número do setor do descritor" = zero;

obs.: remove-se o descritor do arquivo do diretório do disco, mantendo-se a sua entrada na cópia do diretório, na memória.

- e) se "número de processos que abriram o arquivo" não for igual a zero (outros processos estão usando o arquivo), será enviada uma mensagem ao computador mestre, indicando que o arquivo foi fechado e que, se for o caso, o mesmo será removido quando todos os processos terminarem de usá-lo. Após isto a tarefa é encerrada;
- f) se "número de processos que abriram o arquivo" for igual a zero, faz-se:
- f.1) se "tipo de arquivo" for igual a zero (permanente) e "existência de seu descritor em disco" não for igual a zero (existe no diretório do disco), o computador mestre será avisado que o fechamento foi executado e a tarefa será encerrada;
- f.2) se "tipo de arquivo" é igual a 1 (temporário) remove-se sua entrada na cópia do diretório e libera-se as trilhas ocupadas pelo arquivo. Avisa-se que o fechamento foi completado e a tarefa é encerrada;
- f.3) se "tipo do arquivo" é igual a "zero" (permanente) e "existência do seu descritor" é igual a zero (não existe no diretório do disco), grava-se seu descritor no diretório do disco, faz-se "existência do seu descritor" igual a 1 e avisa-se ao computador mestre que a tarefa foi completada.

### III.6

### Resumo das limitações

Na implementação do Sistema foram impostas

algumas restrições, as quais, entretanto, não prejudicam o atendimento das necessidades reais do Terminal Autônomo. Estas restrições, que decorrem das características definidas para o Sistema e da limitação da capacidade de memória do controlador (presentemente, em 8K bytes), são apresentadas a seguir:

- podem existir, no máximo, 16 arquivos abertos simultaneamente;
- apesar do projeto de "hardware", do controlador, prever a ligação de até 120 unidades de discos flexíveis, o "software" foi definido para operar com até 8 unidades (este número pode, se for o caso, ser facilmente expandido);
- as mensagens de comunicação entre o computador mestre e o controlador poderão ter, no máximo, 16 bytes de texto;
- a unidade mínima de alocação de área, para os arquivos nos discos flexíveis, é uma trilha;
- a área alocada para um arquivo deve ser constituída por trilhas contíguas;
- o formato de gravação nos discos flexíveis é o mesmo utilizado pela IBM, só alterando a parte de identificação do disco;
- podem existir, no máximo, dez tarefas sendo executadas compartilhadamente;
- o número máximo de arquivos que podem ser armazenados em um disco é 19 (\*);
- o número máximo de entradas na cópia dos diretórios, na memória do controlador, é 100 (podendo, também, ser expandido).

---

(\*) Foi implementada, como recurso adicional, uma filosofia que permite que arquivos temporários possam ser criados, dependendo da existência de área disponível em disco, sem esta limitação imposta.

#### IV. O SISTEMA IMPLEMENTADO

Partindo das características definidas no capítulo III, o sistema foi programado de maneira a possibilitar fácil manutenção, no caso da necessidade de eliminar algumas de suas limitações.

Considerando que as execuções das tarefas são originadas pelo computador mestre, o sistema implementado será apresentado em uma ordem que é natural para as atividades desenvolvidas pelo controlador.

##### IV.1 Recepção e Enfileiramento de Mensagens

Como já foi descrito, a comunicação entre o computador mestre e o controlador é efetuada através do envio de mensagens no modo byte-a-byte.

Para enviar um byte ao controlador, o computador mestre escreve em seu registrador de comunicação. A informação fica disponível ao controlador e um pedido de interrupção sinaliza a operação. O controlador lê o byte e escreve no seu registrador de comunicação, o que interrompe o mestre, avisando que a operação foi completada. A operação segue até que todos os bytes da mensagem tenham sido enviados.

Na rotina de recepção de mensagens vindas do computador mestre, os bytes do texto, a medida que chegam (estado "C" do automato de recepção - ver figura (11)), são

inseridos no "buffer" apontado por "LIVRE" da fila de "buffers" de recepção, conforme é mostrado na figura (19).

Esta fila é encadeada, através do apontador P1, e tem capacidade para receber até 10 mensagens.

Ao ser recebido o caráter de fim de mensagem "EOT" (estado "D" do automato de recepção), é executada uma rotina de inserção de mensagem na "fila de mensagens recebidas".

Esta rotina, definida na figura (20), faz com que a mensagem recebida, apontada por "LIVRE", seja inserida no fim da "fila de mensagens recebidas", cuja primeira mensagem é apontada por "MRECB" e a última é apontada por "ULTRC", ver figura (21).

Ao ser recebido um caráter de início de uma mensagem "STX" (estado "A" do automato de recepção), e se não houver mais "buffers" disponíveis para recepção ( $LIVRE=0$ ), será enviado um caráter "FF" como eco do byte recebido. Deste modo, a rotina de transmissão do computador mestre, como não recebeu um "ACK" como eco, ficará constantemente tentando iniciar a transmissão da mensagem.

Esta situação vai perdurar até que o controlador termine de executar uma tarefa e tenha a necessidade de começar a transmitir uma resposta (ver item III.1), com isto liberando um "buffer de recepção".



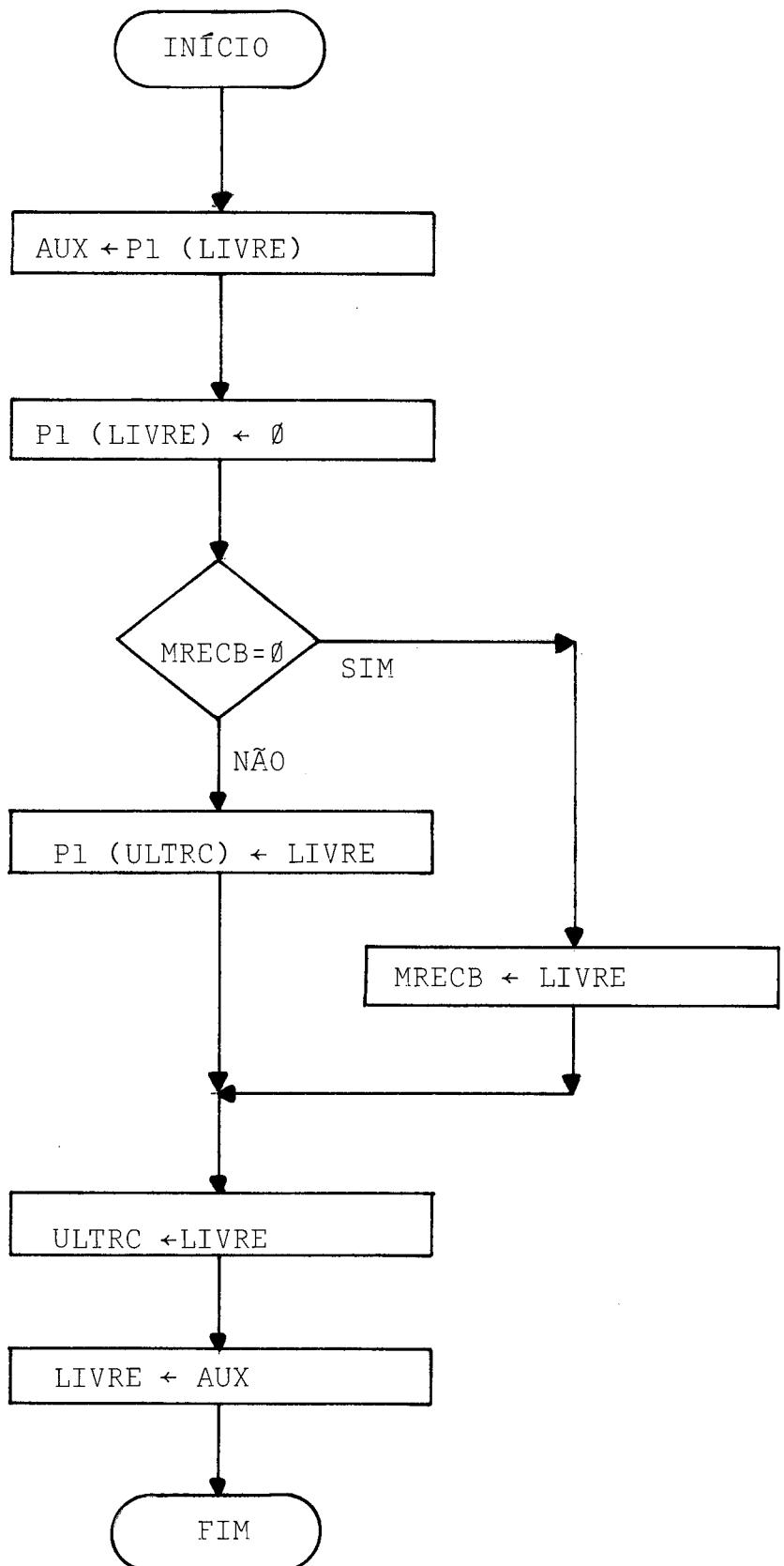


Figura 20 - Rotina de inserção de uma mensagem na "fila de mensagens recebidas" (INSRCB).

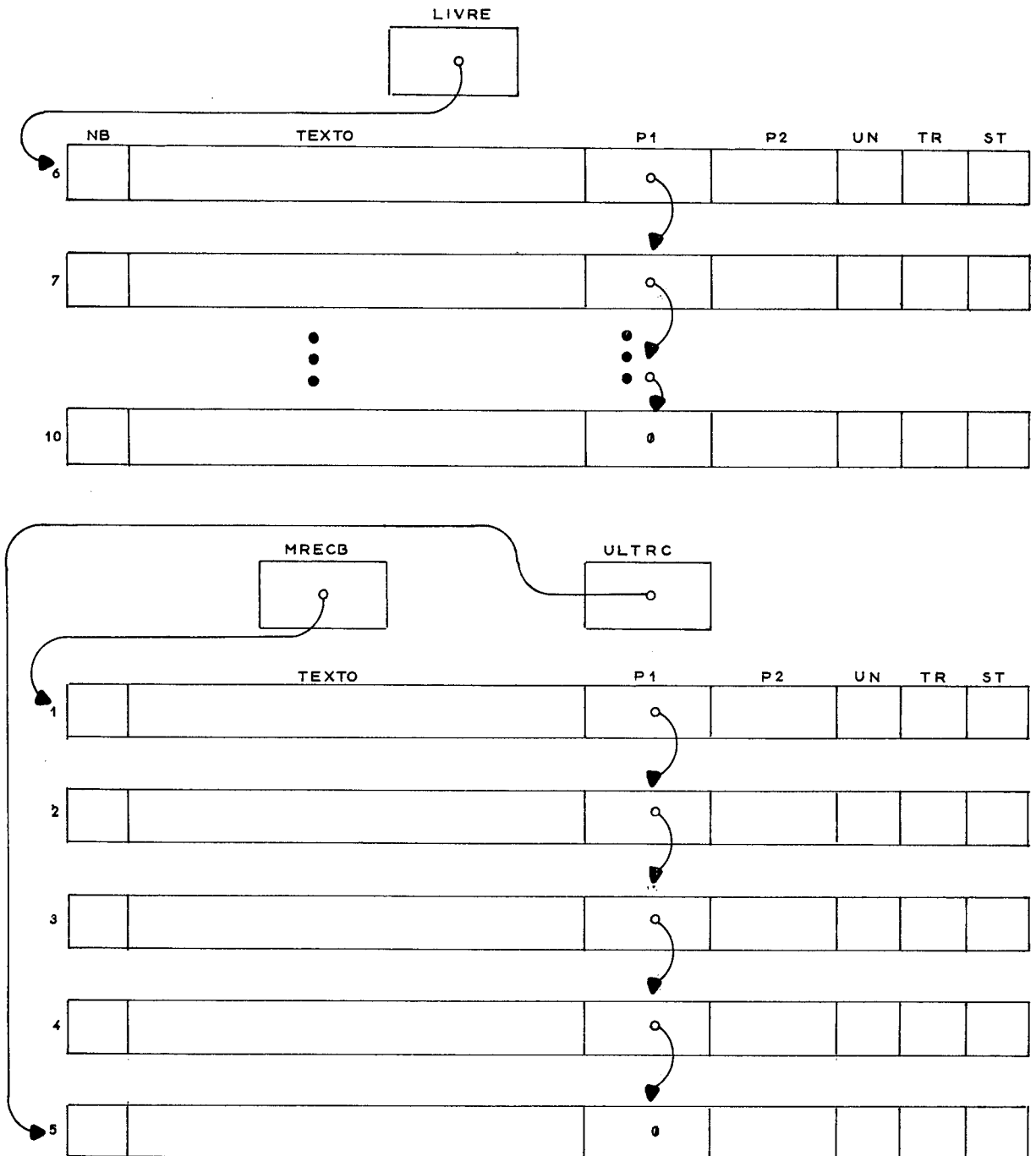


Figura 21 - Fila de mensagens recebidas. (Está representada a situação da fila após a inserção de 5 mensagens)



## IV.2 Análise do Texto e Geração da Estrutura para Otimização do Acesso aos Discos

As mensagens recebidas pelo controlador, ao serem analisadas, geram as tarefas a serem executadas.

Para uma melhor compreensão de como estas tarefas são geradas, este item será apresentado em duas etapas.

### IV.2.1 Análise do texto

Nesta fase, o texto de uma mensagem recebida é analisado, passando pela verificação de viabilidade de execução. Deste modo, todos os erros pré-operativos são aqui detectados.

No caso de detecção de erro, serão geradas as mensagens de resposta para o computador mestre. Isto é feito através da remoção da mensagem da "fila de mensagens recebidas" (ver figura (22)) e inserção, da mesma, com um código de erro, no fim da "fila de mensagens a transmitir". A primeira mensagem desta fila é apontada por "RESPO" e a última é apontada por "ULTRSP" (ver figura (23)). Um código de tarefa não reconhecido, por exemplo, recebe o código de erro  $DE_{16}$ .

Se não foi encontrado nenhum erro durante a análise do texto, então tem-se uma tarefa a ser executada. Nesta análise são apropriados os valores correspondentes aos campos "UN", "TR" e "ST", definidos na figura (19). A mensagem analisada é então retirada da "fila de mensagens recebidas" (ver figura (22)) e inserida no fim da "fila de mensagens ana-

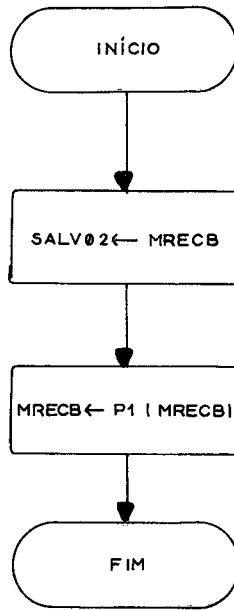


Figura 22 - Rotina de remoção de uma mensagem da "fila de mensagens recebidas" (REMRCB).

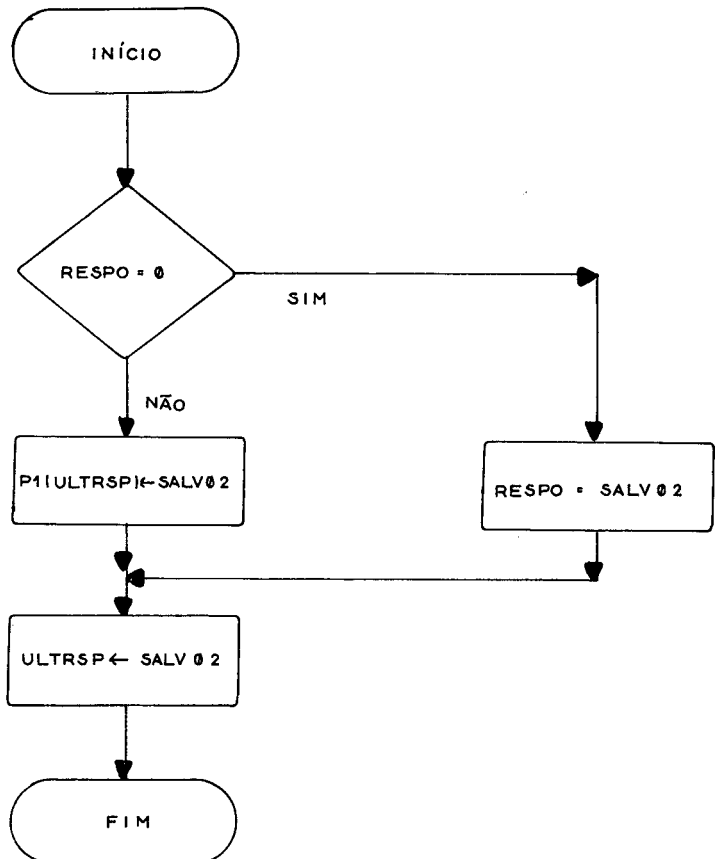


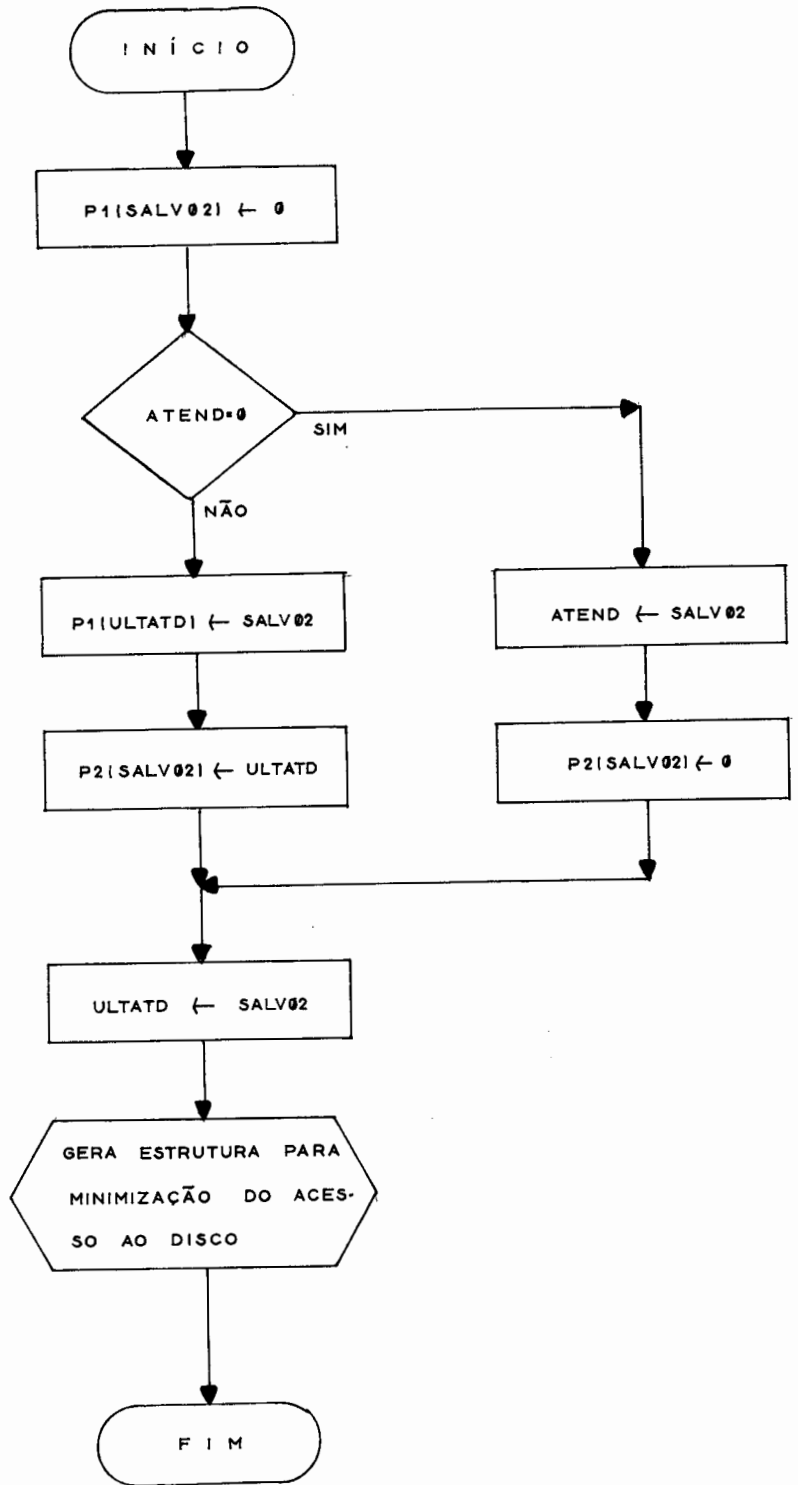
Figura 23 - Rotina de inserção de uma mensagem na "fila de mensagens a transmitir" (INSRSP).

lisadas". Nesta fila, a primeira mensagem é apontada por "ATEND" e a última é apontada por "ULTATD" (ver figura (24)). Neste instante é criada, baseada nos dados da análise do texto, uma nova entrada na estrutura que contém as trilhas (classificadas em ordem crescente, por unidade de disco) a serem acessadas, com a finalidade de prover meios para otimizar o acesso aos discos. Esta criação de nova entrada na lista de trilhas a serem acessadas está descrita no próximo item.

O procedimento descrito refere-se exclusivamente às tarefas que exigem atividades de leitura ou escrita nas unidades de disco flexível. Tarefas como "abrir arquivo" ou "remover processo", que não exigem acesso a disco, são executadas e respondidas na rotina de análise correspondente. Neste caso, após a análise, a mensagem é retirada da fila de mensagens recebidas, ver figura (22), e inserida, com a resposta conveniente, na fila de mensagens a transmitir, ver figura (23).

#### IV.2.2                    Geração da estrutura para otimização do acesso aos discos

Na definição desta estrutura, pensou-se primeiramente em utilizar listas duplamente encadeadas (uma para cada disco, classificadas em ordem crescente das trilhas a serem acessadas) pois em tempo de execução esta organização poderia facilitar a remoção de um elemento da lista, já que, a tarefa a ser executada não corresponde, em geral, ao primeiro elemento da lista (depende da posição da cabeça de leitura e



"SALV02" aponta para o primeiro byte da mensagem a ser inserida.

Figura 24 - Rotina de Inserção de uma mensagem na "fila de mensagens analisadas" (INSANA).

gravação do disco). Além disto, esta organização poderia facilitar a determinação da tarefa a ser executada se a lista fosse percorrida pelo encadeamento correspondente ao movimento da cabeça (ordem crescente com movimento para dentro e ordem decrescente com movimento para fora).

Quando, no entanto, passou-se a utilizá-las, notou-se que o processo de percorrer a lista pelo encadeamento correspondente ao movimento da cabeça, tornou-se bastante grande, além de serem necessárias posições adicionais de memória para o segundo apontador.

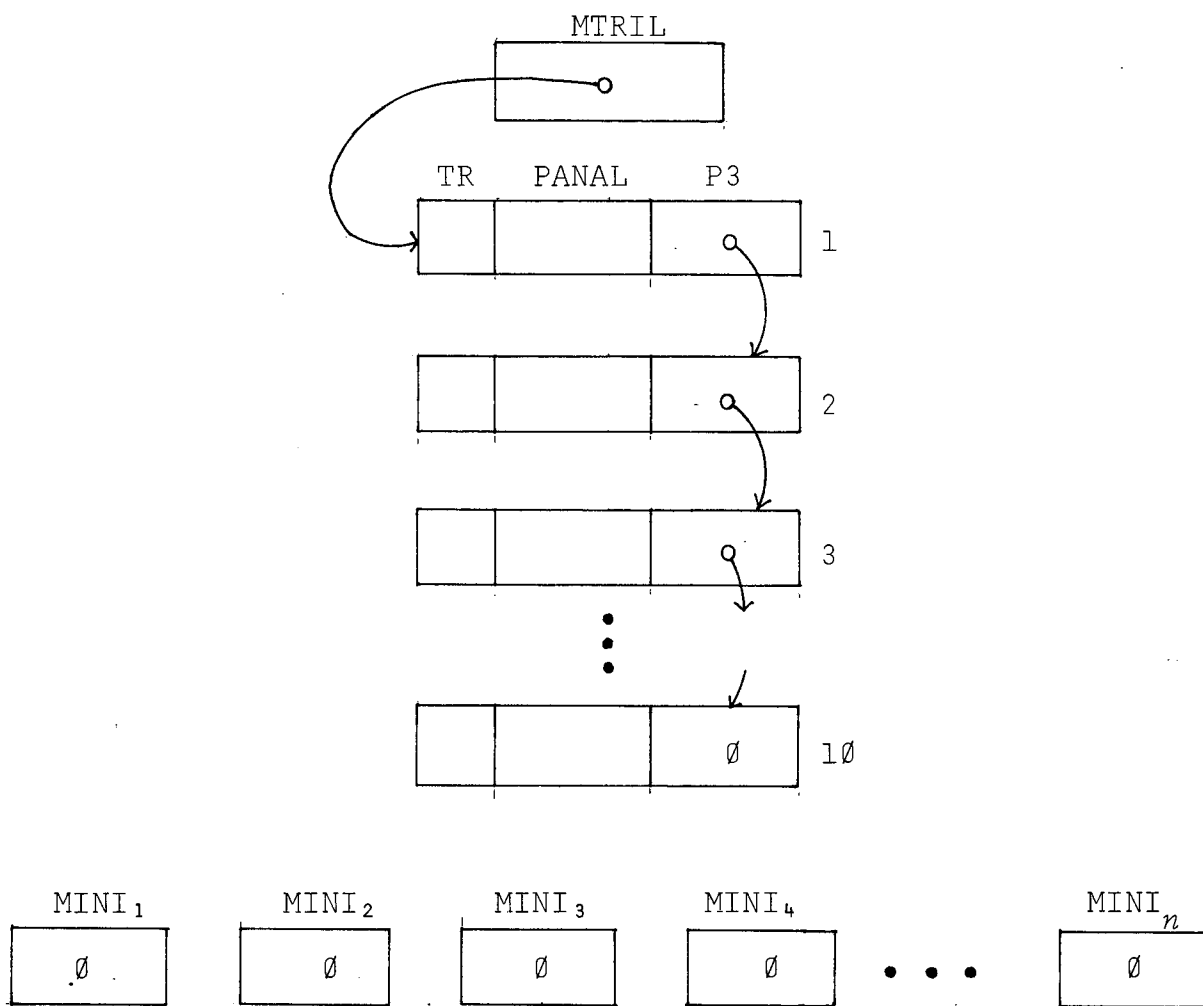
O problema então foi encontrar uma forma de estrutura que permitisse fácil remoção de um elemento conhecido, sem ter que percorrer toda a estrutura para acertar apontadores.

Pensou-se então em fazer a remoção do elemento quando de sua escolha (sendo conhecido, neste caso, quem aponta para ele) em vez de removê-lo quando do término de uma tarefa.

Deste modo pode-se usar uma lista com encadeamento único. A estrutura definida está apresentada na figura (25).

Ao incluir-se elementos na estrutura, gera-se filas encadeadas, uma para cada disco (unidade que deverá ser acessada para a execução da tarefa), com a característica de que os elementos, em cada uma das filas, estarão classificados em ordem crescente de trilha a ser acessada.

Para fazer a inclusão de uma nova entrada, chama-se a rotina de geração de uma entrada para a fila de otimização (entrada esta que será vinculada à fila do disco que



MTRIL: apontador para o próximo elemento disponível a ser utilizado para a criação de uma nova entrada na estrutura de otimização de acessos.

TR: contém o número da trilha a ser acessada para a execução da tarefa definida na área apontada por "PANAL". Esta informação é gerada quando se cria uma nova entrada.

PANAL: aponta para o início da mensagem de definição de tarefa na "fila de mensagens analisadas". Esta informação é gerada quando se cria uma nova entrada.

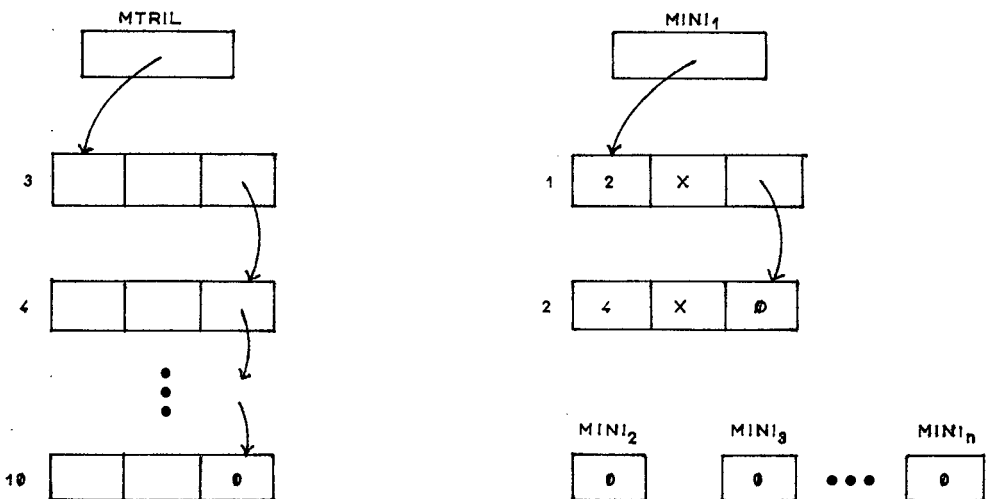
P3: aponta para o próximo elemento da fila.

MINI<sub>i</sub>: é o nodo cabeça da lista para otimização do acesso à unidade de disco "i". "n" é o número de discos ligados ao sistema.

Figura 25 - Estrutura para otimização do acesso aos discos.

deverá ser acessado), que está definida na figura (26).

Para exemplificar o processo de inserção, de modo classificado, na estrutura, suponha-se a seguinte condição inicial da mesma:

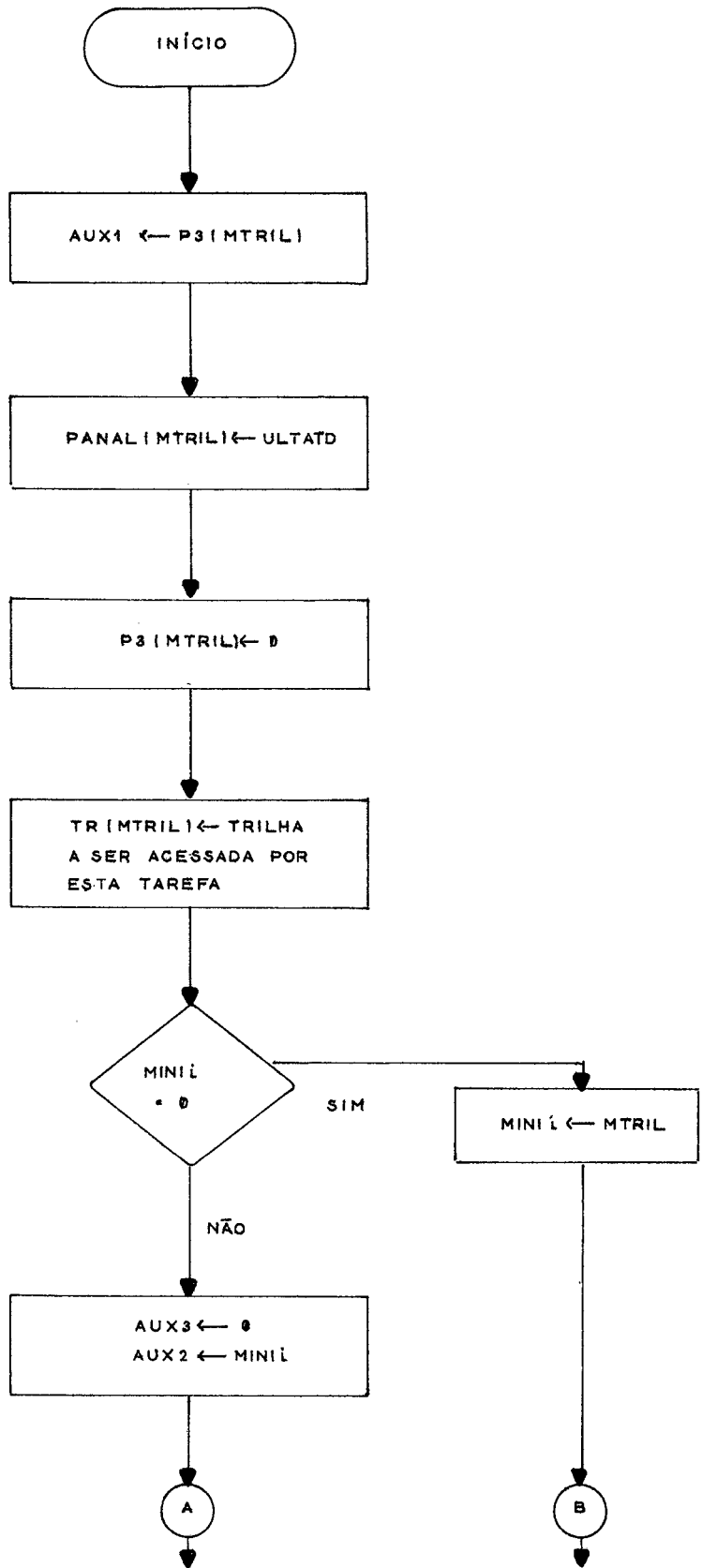


Nesta condição, tem-se duas tarefas a serem executadas no disco flexível de número lógico 1 (acessar trilhas 2 e 4) e nenhuma tarefa para os outros discos do sistema.

Tendo que inserir uma nova tarefa para o disco lógico número 1, dependendo da trilha a ser acessada, a situação final da estrutura será uma das três indicadas na figura (27).

Para se ter uma visão geral do funcionamento, até este ponto, é interessante o seguinte exemplo.

Partindo-se da situação existente na figura (21), suponha-se que ao ser analisado o texto da mensagem número 1, encontre-se um erro que inviabilize a execução da tarefa nele solicitada e que nas análises dos textos das mensagens 2, 3 e 4 não seja detectado nenhum erro. Supondo que as características de execução das tarefas solicitadas sejam:



NOTA: "i" indica o número da unidade de disco que será acessada para a leitura/gravação da trilha desejada.

Figura 26 - Rotina para geração de uma entrada na fila para otimização de acesso a disco - GERTRI.



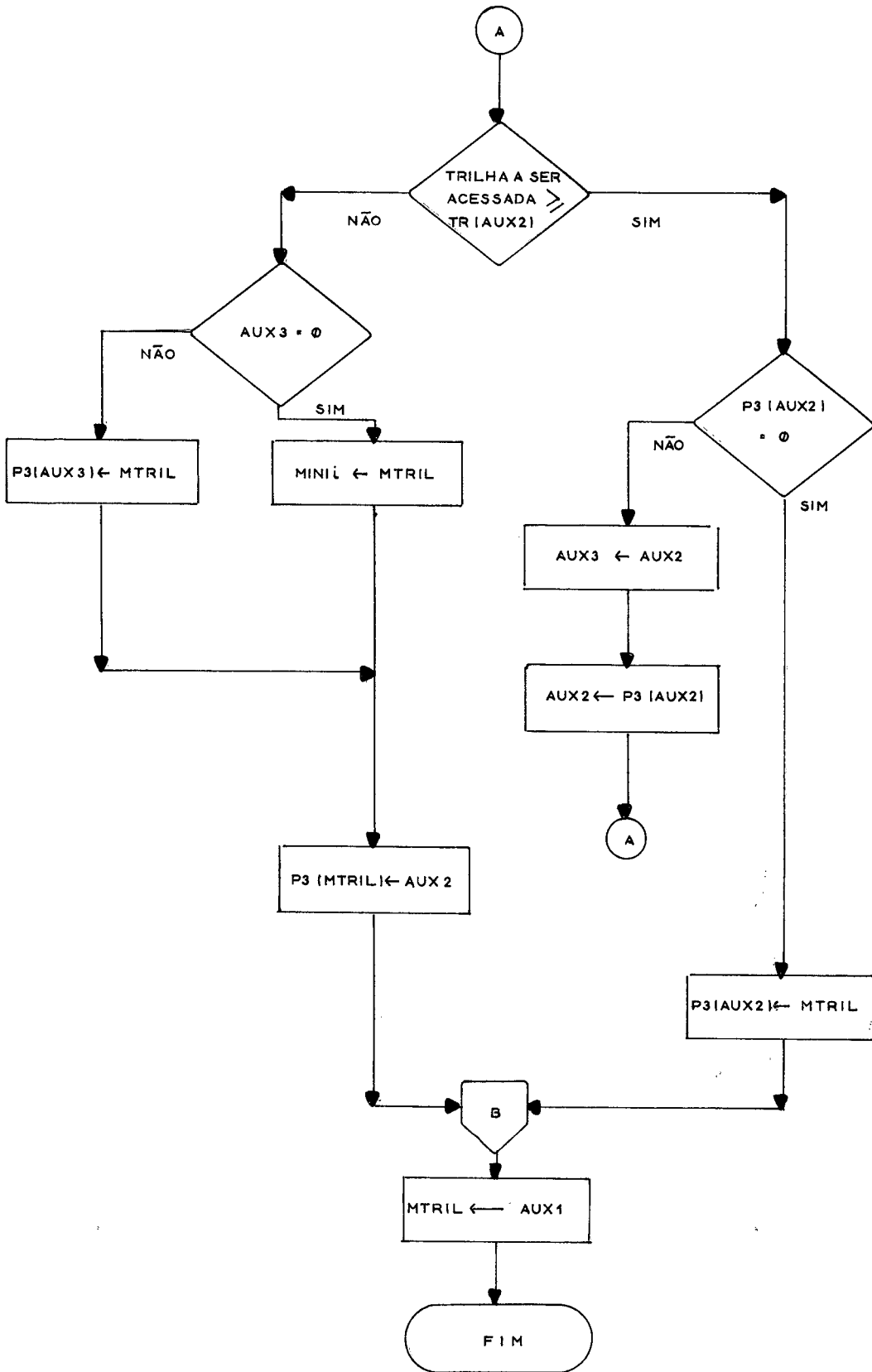
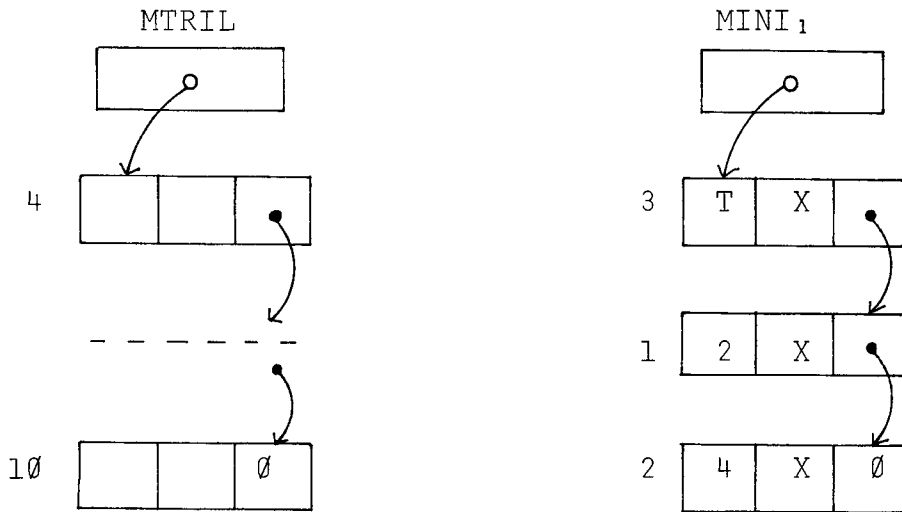
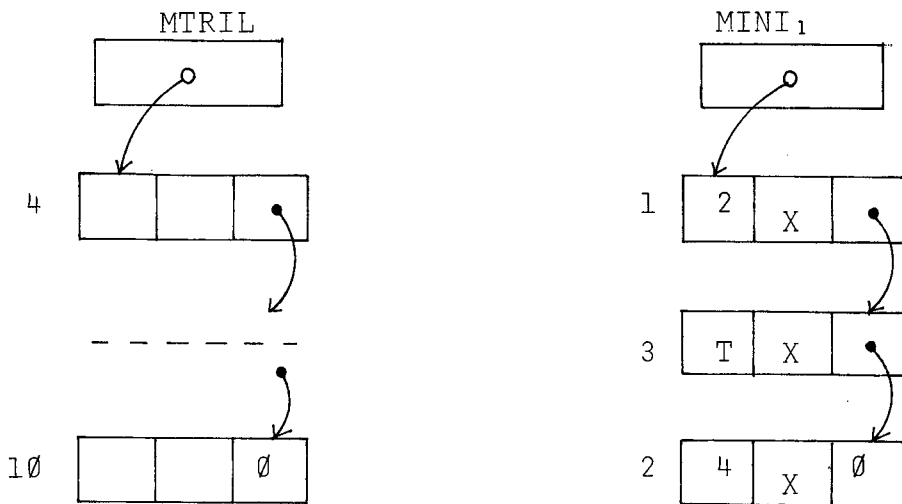


Figura 26 - Continuação

a) Trilha T a ser acessada pela nova tarefa é menor que 2:



b) Trilha T a ser acessada pela nova tarefa é maior ou igual a 2 e menor do que 4:



c) Trilha T a ser acessada pela nova tarefa é maior ou igual a 4:

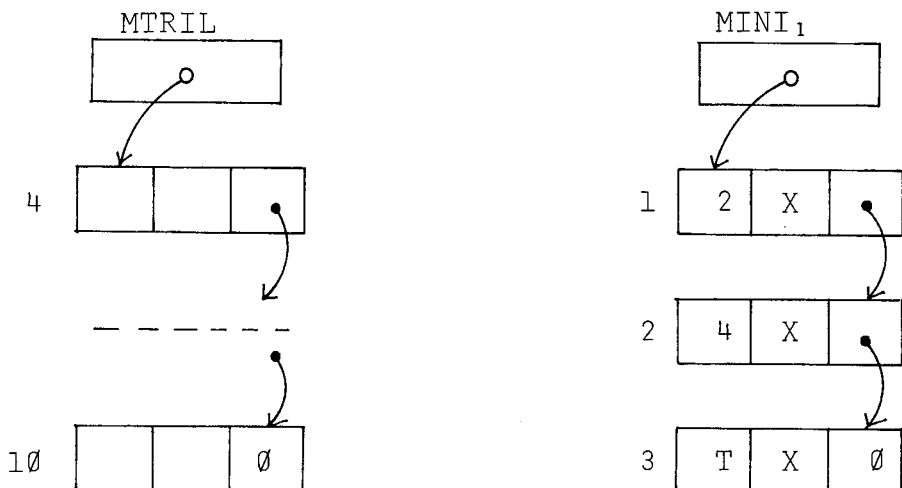


Figura 27 - Situações da inserção na estrutura para otimização do acesso a disco.

MENSAGEM 2 ▣ unidade a acessar = 1

trilha a acessar = 5

setor a acessar = 1

MENSAGEM 3 ▣ unidade a acessar ▣ 3

trilha a acessar = 1

setor a acessar = 20

MENSAGEM 4 ▣ unidade a acessar = 1

trilha a acessar ▣ 0

setor a acessar ▣ 1

Neste caso a situação final das estruturas, até agora descritas, ficará como a mostrada na figura (28).

#### IV.3

#### Corpo do Programa do Controlador

O corpo do programa do controlador de disco flexível é um loop eterno onde são comandadas as suas diversas atividades. A figura (29) mostra o fluxograma básico do corpo e apresenta os diversos módulos que o compõem.

O módulo "B", da figura (29), é o módulo de recepção e análise das mensagens, descrito nas secções IV.1 e IV.2.

O módulo "D", da mesma figura, faz a seleção de uma tarefa, para o disco correspondente, objetivando a otimização do acesso, isto é, seleciona a tarefa que necessita acessar a trilha que está mais próxima da cabeça de leitura/gravação, considerando o sentido do movimento da mesma. Este módulo está descrito em detalhes na secção IV.4.

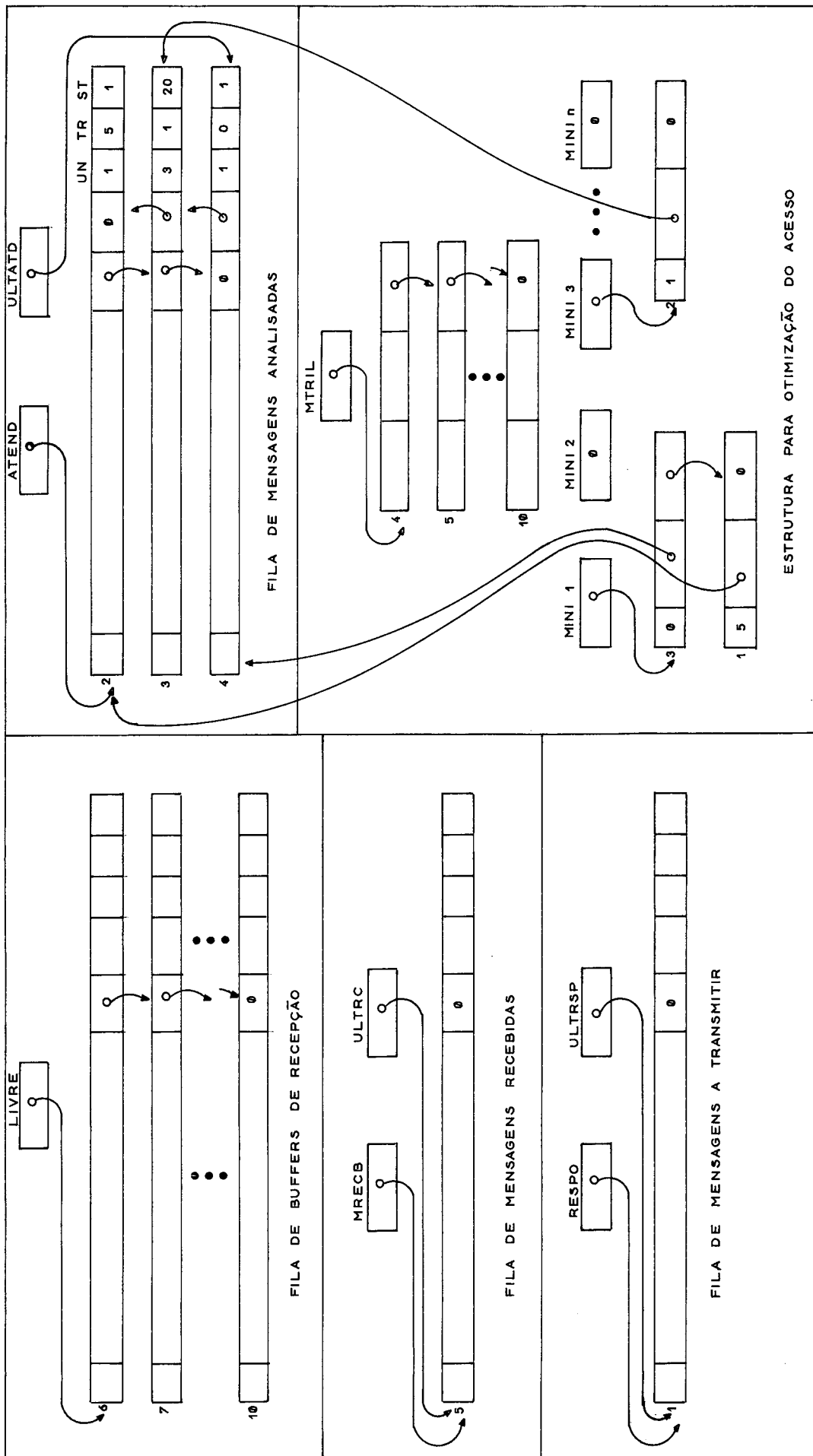


Figura 28 - Uma situação para as estruturas de dados definidas na seção IV.2.

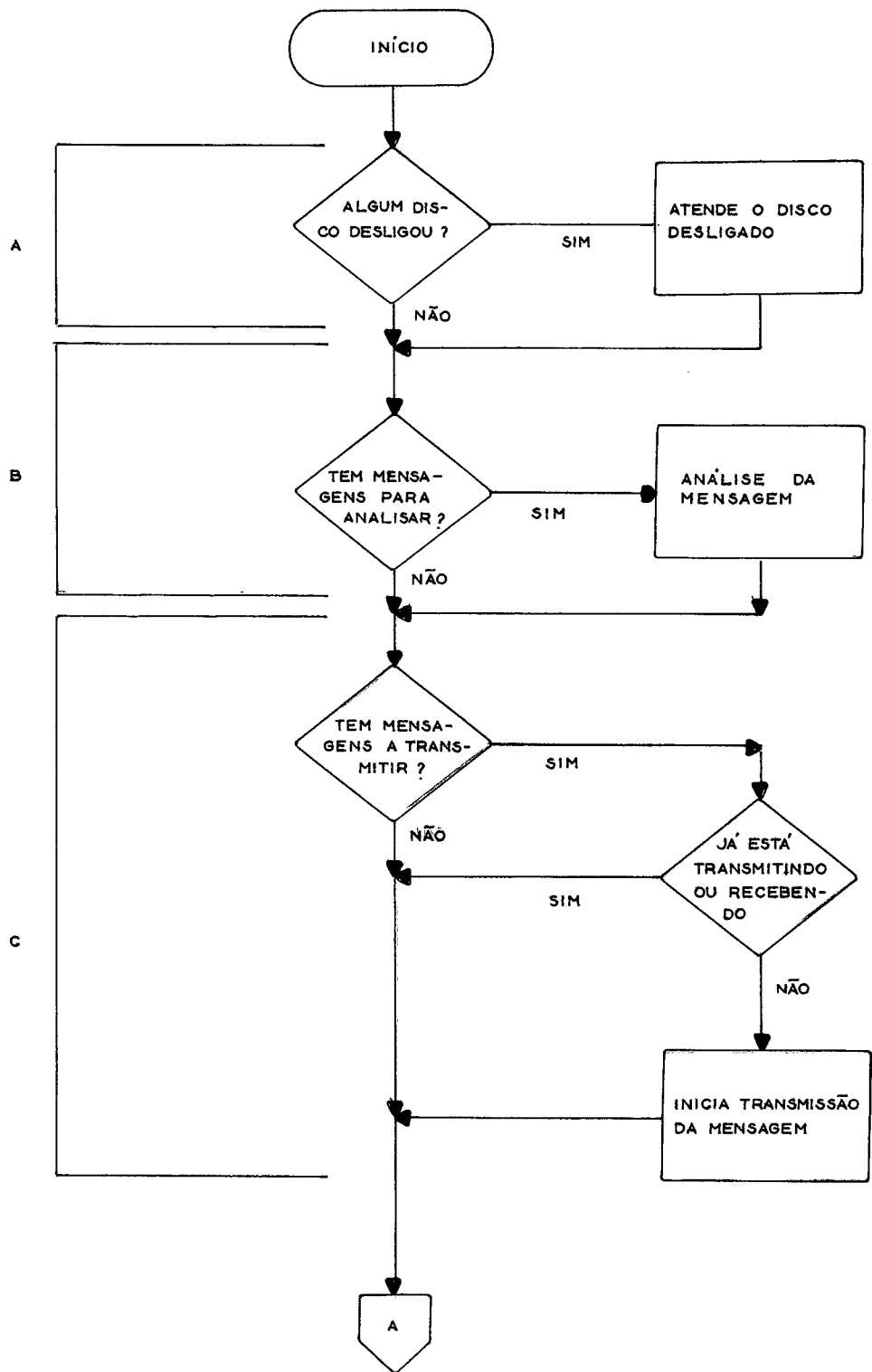


Figura 29 - Fluxograma Básico do Corpo do Controlador

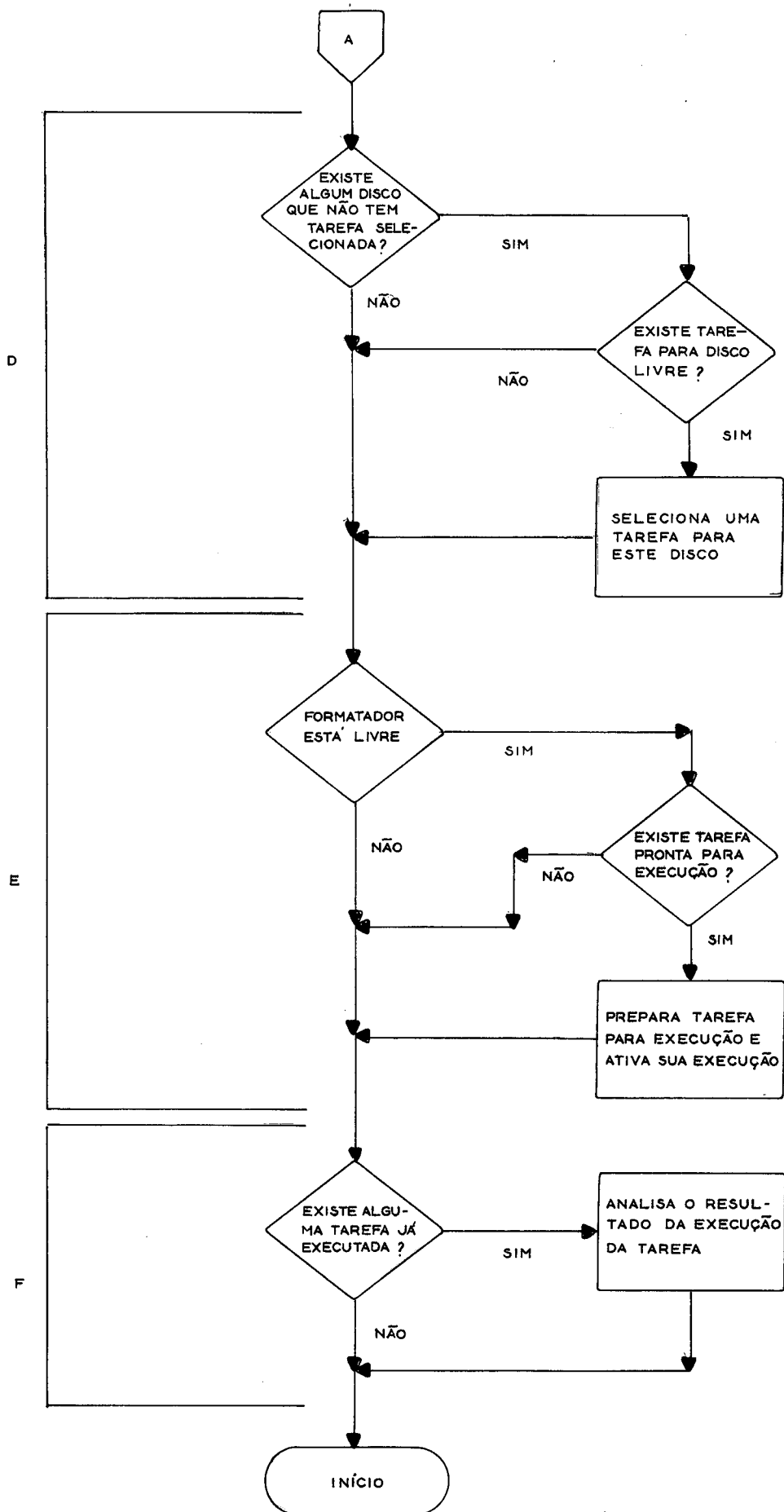


Figura 29 - Continuação

O módulo "E" faz a preparação e ativação da execução de uma tarefa. Note que, neste caso, só existirá uma tarefa sendo executada, de cada vez. Isto se deve a característica de "hardware" do controlador que só admite o formata-dor estar ligado a uma unidade de disco de cada vez (ver figura (09)). Este módulo está descrito na secção IV.5.

O módulo "C" faz a transmissão das mensagens com respostas para computador mestre e está descrita na secção IV.6.

O módulo "A", descrito na secção IV.7, executa as atividades que são necessárias por ocasião do desligamento de um disco.

#### IV.4 Seleção da Tarefa que Otimiza o Acesso aos Discos

Quando o controlador tentar executar uma tarefa, já estará criada uma fila de tarefas analisadas e uma fila, por unidade de disco, de trilhas a serem acessadas (estrutura para otimização do acesso).

Como o "software" tem, na memória, em qualquer instante, a exata posição da cabeça de leitura/gravação para cada unidade de disco flexível a ele ligada, a minimização do tempo de acesso pode ser conseguida escolhendo-se para execução, a tarefa que possa ser executada no menor espaço de tempo.

Como, no caso desta implementação, um aces

so a disco sempre promove a transferência de um setor de dados (128 bytes), a otimização pode ser efetuada através de uma política de atendimento, que considera três fatores:

- atender em primeiro lugar a unidade de disco flexível que está com a cabeça de leitura/gravação mais próxima de uma trilha a ser acessada (tempo de UNIDADE);
- atender em primeiro lugar a tarefa, dentre as destinadas para uma mesma unidade de disco, que proporcione o menor movimento de braço (tempo de SEEK);
- atender em primeiro lugar, dentre as tarefas que se destinam a uma mesma unidade de disco cuja cabeça de leitura/gravação já está posicionada na trilha desejada, aquela, cuja execução pode ser iniciada no menor tempo, isto é, aquela cujo setor se encontra mais próximo da cabeça de leitura/gravação (tempo de SETOR).

O primeiro fator (tempo de unidade) pode ser esquecido, pois o mesmo é atendido, indiretamente, no atendimento do segundo fator: a execução de uma tarefa somente é iniciada quando a cabeça de leitura/gravação atinge a trilha desejada pela tarefa e a próxima trilha a ser acessada, dentre as que tiverem sendo requeridas por tarefas, é obtida pela minimização relativa ao tempo de SEEK (segundo fator), para cada unidade. Desta maneira, a unidade que primeiro atingir sua trilha desejada, já considerando que sempre será feito o menor movimento possível dentro de cada unidade, será a primeira a ser atendida.

O terceiro fator não será considerado nesta implementação pois não se tem, a todo instante, a posição da cabeça de leitura/gravação dentro de uma trilha, no disco. A



idéia que levou a desconsiderar este fator é a de que as instruções adicionais necessárias para esta função (controle da posição de setor), acrescida do tempo em que o controlador deveria se preparar para acessar um setor (deve estar preparado dois setores antes do desejado) e as modificações de "hardware" que seriam necessárias, não compensariam o tempo que se iria ganhar, já que, nesta implementação, não poderá haver mais de um arquivo em uma trilha e a probabilidade de dois processos estarem acessando um mesmo arquivo é pequena.

O segundo fator (minimização do tempo de SEEK) permite otimizar o tempo de acesso, mas tem um grave inconveniente: a minimização da movimentação da cabeça de leitura/gravação pode originar o problema da espera indefinida ("indefinite postponement"). Isto aconteceria, por exemplo, se a cabeça de leitura/gravação estivesse posicionada em uma trilha  $K$  e ocorresse uma seqüência de tarefas requerendo acesso a trilhas próximas e em torno de  $K$  (por exemplo,  $K+1$ ,  $K$ ,  $K-1$ ,  $K$ ,  $K+1$ , etc). Isto ocasionaria que tarefas requerendo acesso a trilhas distantes de  $K$  ficassem um longo tempo a espera de execução (inclusive, teoricamente, este tempo poderia ser infinito).

As considerações do parágrafo anterior levaram a escolha de um algoritmo que otimiza (no sentido de reduzir) o tempo de acesso aos discos e o tempo médio de espera das tarefas, através da minimização da freqüência de mudanças no sentido de movimentação dos braços dos discos. Este algoritmo que é uma variação do algoritmo apresentado por HOARE<sup>3</sup> utiliza como referência, o sentido de movimentação da cabeça em cada disco, con

forme será explicado a seguir.

Se o movimento da cabeça de leitura/gravação, em um disco, está sendo "para dentro", então atende-se os pedidos (tarefas) que necessitam de acesso a trilhas cujas posições sejam iguais ou superiores a posição atual da cabeça nesse disco. Caso não exista nenhuma tarefa nesta condição, inverte-se o sentido do movimento da cabeça e atende-se a todas as tarefas cujas trilhas a serem acessadas sejam inferiores ou iguais a posição atual da cabeça (processo semelhante ao funcionamento de um elevador).

A tarefa a ser escolhida deverá, ainda e evidentemente, ser a que exija o menor deslocamento possível do braço.

Convém observar que, teoricamente, ainda existe a possibilidade de certas tarefas ficarem indefinidamente a espera de atendimento. Isto aconteceria, por exemplo, no caso em que existisse um fluxo intenso e contínuo de pedidos de acesso a uma mesma trilha de disco. Neste caso, as tarefas que necessitassem fazer acesso a outras trilhas poderiam ficar eternamente sem ser executadas. Uma solução para esse problema seria manter duas filas para cada disco, cada uma correspondendo a um sentido de movimento do braço. Toda a tarefa que necessitasse acesso a um trilha com posição maior que a posição atual da cabeça de leitura/gravação seria colocada na fila correspondente ao sentido "para dentro". Tarefas que necessitassem acesso a trilhas menores que a correspondente à posição atual da cabeça seriam colocadas na fila correspondente ao sentido "para fora". Tarefas que necessitassem acesso à trilha correspondente à posição atual da

cabeça seriam colocadas na fila correspondente ao sentido contrário do sentido atual de movimentação do braço. Desta maneira também se minimizaria a freqüência de mudanças no sentido de movimentação do braço e se conseguiria eliminar o problema da espera indefinida, conseguindo-se reduzir as movimentações da cabeça (não minimizá-las) e os tempos médios de espera das tarefas.

Como, no caso do Sistema aqui descrito, é pequena a probabilidade de ocorrer uma seqüência de várias tarefas necessitando acesso a uma mesma trilha, optou-se pelo algoritmo de otimização que minimiza a freqüência de mudanças no sentido de movimentação do braço, sem se preocupar com o problema da espera indefinida.

A rotina que faz a busca e escolha da tarefa a ser executada, para uma unidade de disco flexível, é mostrada no fluxograma da figura (30).

Esta rotina tem a característica de atender, para duas ou mais tarefas que acessem a mesma trilha de um disco, a tarefa que foi solicitada em primeiro lugar, independentemente do sentido de movimento da cabeça de leitura/gravação.

No fluxograma, "i" representa o número da unidade de disco flexível e no ponto "D" (encontrada a tarefa que ocasiona o menor movimento) tem-se:

AUX1 = aponta para o primeiro byte da tarefa a ser atendida;

AUX2 = aponta para o primeiro byte da tarefa que antecede a apontada por AUX1.

A variável EXEC<sub>i</sub> vai, então, apontar para o primeiro byte de um elemento da "lista de mensagens analisa-

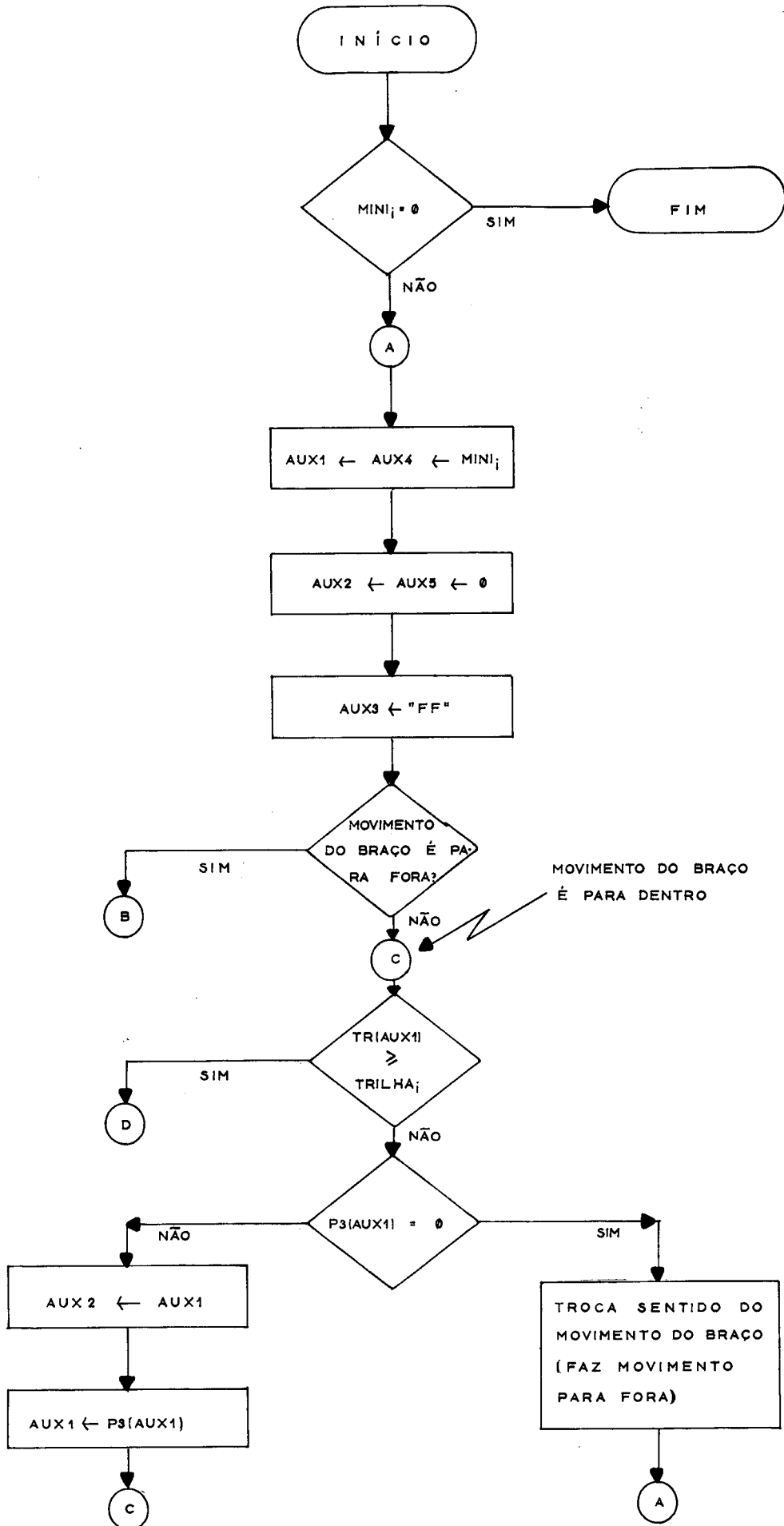


Figura 30 - Rotina de Busca da tarefa a executar (BUSTAR).

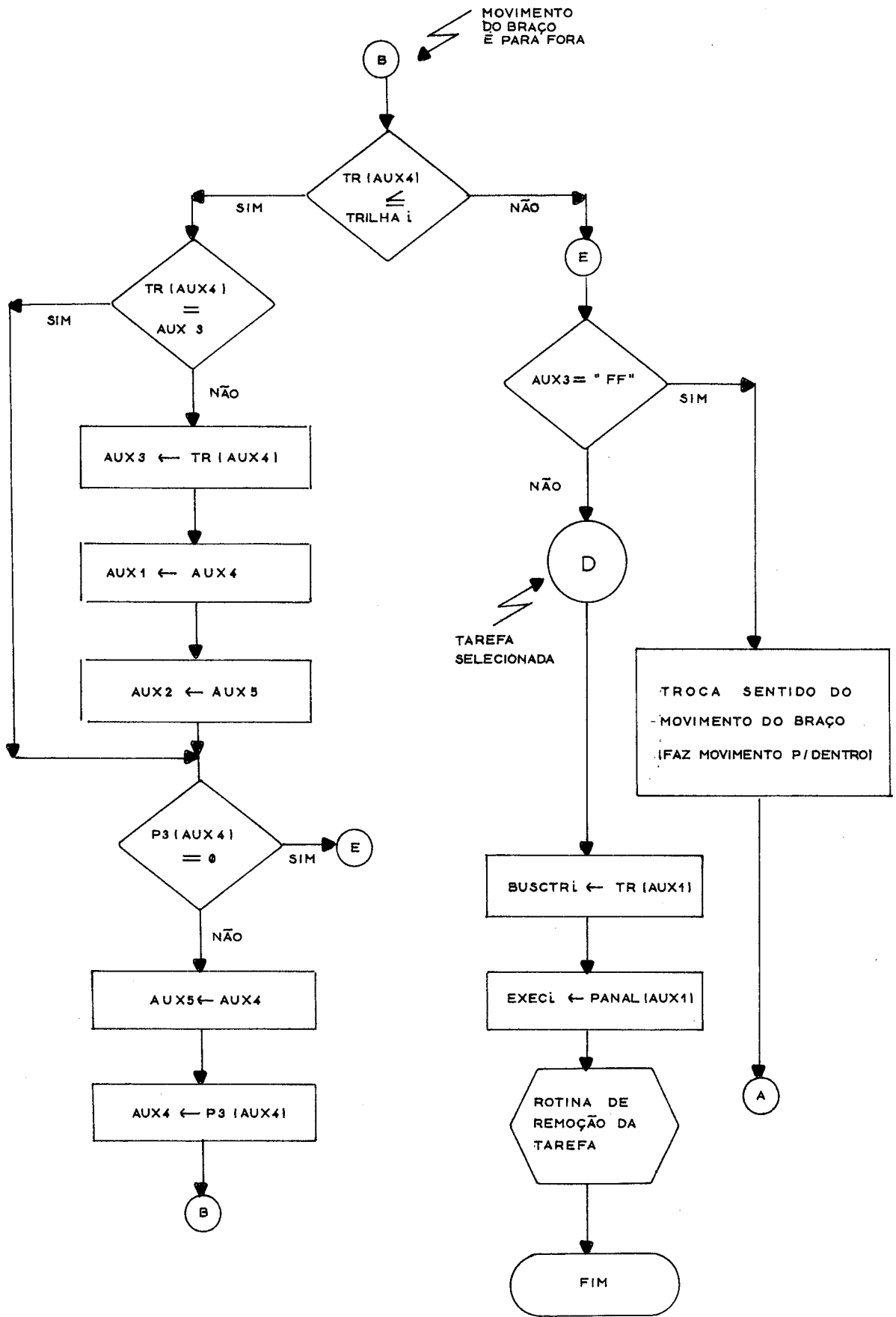


Figura 30 - Continuação

das" e indicará que esta mensagem será atendida (executada) pela unidade de disco "i".

A primeira providência é, deste modo, fazer com que a cabeça de leitura/gravação desta unidade, se desloque para a trilha desejada, salva em BUSCTR<sub>i</sub>. Ao ser atingida a trilha, será indicado que a tarefa está pronta para execução.

A primeira unidade de disco flexível que atingir a trilha desejada, será a unidade que primeiro começará a executar a tarefa definida no texto da mensagem apontada por "EXEC<sub>i</sub>".

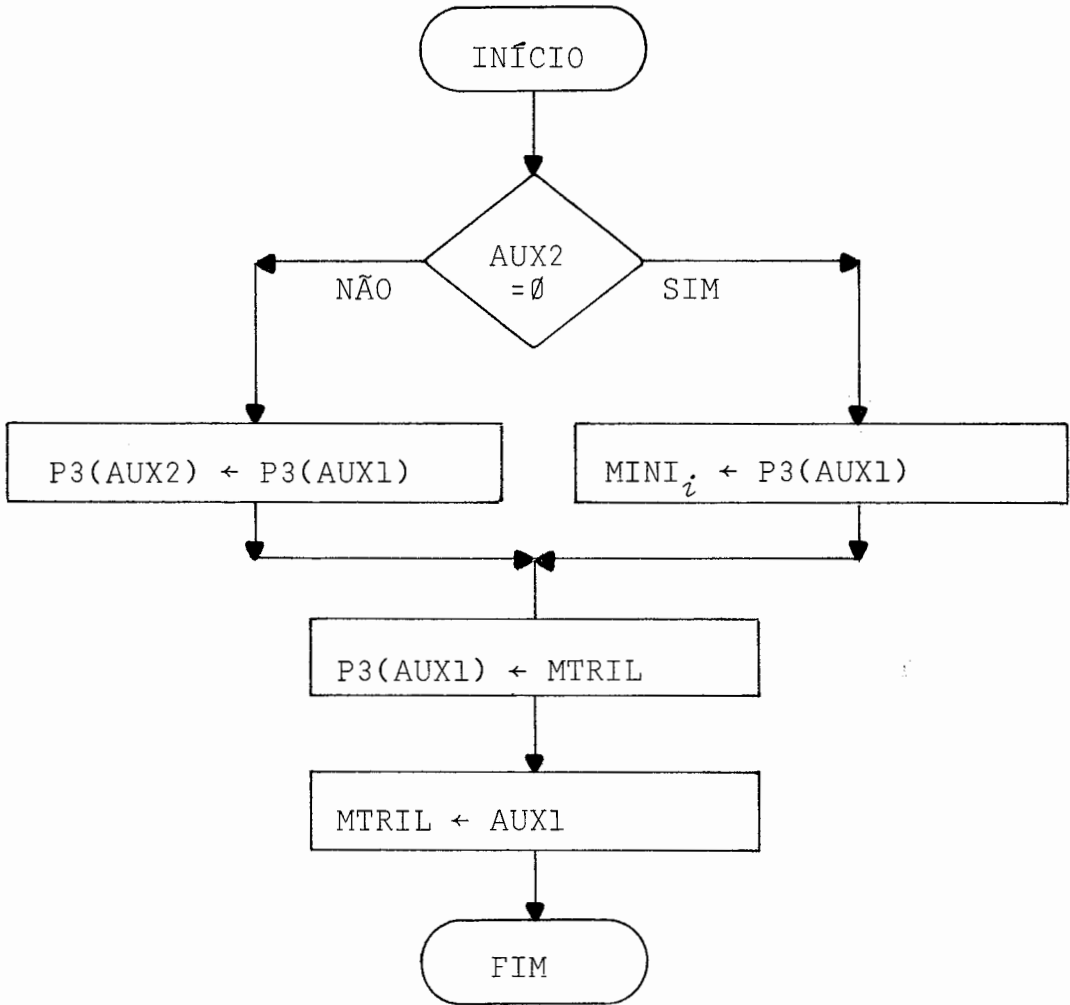
Além disso, antes de encerrar a rotina, é necessário remover a tarefa apontada por "AUX1", da estrutura para otimização do acesso, pois a mesma já foi selecionada. Isto é feito pela rotina de remoção de tarefa definida na figura (31).

#### IV.5

#### Execução das Tarefas

Nesta seção são descritas as execuções das tarefas, pelo controlador de disco flexível, conhecido o caminho percorrido pelas mesmas, até chegar a fase de execução.

Ao atingir esta fase, a tarefa a ser executada em uma unidade "i", terá em EXEC<sub>i</sub> o apontador para a mensagem na "fila de mensagens analisadas". Com os dados definidos no texto desta mensagem, pode-se, agora, executar a tarefa solicitada.



AUX1 ← Aponta para a tarefa a ser removida

AUX2 ← Aponta para a tarefa que aponta para a tarefa a ser removida (anterior).

Se  $AUX2 = \emptyset$  então a tarefa a ser removida é a primeira.

Figura 31 - Rotina para remoção da tarefa da estrutura para a otimização do acesso (REMTAR).

As execuções de todas as tarefas, com exceção das tarefas "abre arquivo" e "remove processo", envolvem acesso a discos flexíveis. Nos itens a seguir, serão descritas as execuções das várias tarefas.

Para manter compatibilidade com o Sistema Operacional desenvolvido para o Terminal Autônomo, os códigos utilizados para indicar condições de erro, foram os mesmos previstos pelo Sistema Operacional (houve liberdade de escolha somente nos casos de novos tipos de erros).

Dois destes erros serão citados em separado pois são de uso geral:

<u>Código</u>	<u>Significado</u>
DF <sub>16</sub>	= unidade de disco está/foi desligada;
DE <sub>16</sub>	= mensagem inválida.

#### IV.5.1 Tarefa monta cópia do diretório do disco

Esta tarefa é iniciada quando um disco flexível qualquer é ligado. O Controlador sente a interrupção e gera uma mensagem para montar uma cópia do diretório na memória do controlador.

O formato do texto desta mensagem é:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4 = CONTERÁ "F" BITS 3, 2, 1 e 0 = CONTERÁ "F"
1	1	NÚMERO DA TRILHA A SER ACESSADA=0



BYTE	TAMANHO	DESCRIÇÃO
2	1	NÚMERO DO SETOR A SER LIDO=7
3	1	NÚMERO DA UNIDADE A SER USADA= unidade de disco que foi ligada

Esta mensagem é inserida na "fila de mensagens recebidas", como se tivesse sido recebida do computador mestre. Desta maneira ela será atendida após uma disputa com as outras tarefas solicitadas pelo computador mestre.

Esta tarefa provocará a leitura da trilha zero, setor 7, para obter o número do disco (volume) que foi ligado.

Ao ser executada esta rotina, poderá acontecer que o disco não tenha sido inicializado no formato reconhecido pelo controlador. Neste caso será verificado se o formato de inicialização do disco flexível, atende os padrões de inicialização da IBM. Se atender, a informação do tipo do formato será armazenada e, neste caso, este disco só poderá ser acessado para leitura (gravação de arquivos não será permitida).

No caso do formato ser reconhecido como "padrão do controlador", será armazenado o número do volume.

Em qualquer dos casos será inserida, na estrutura para otimização do acesso, uma nova ordem de leitura, agora para o setor 8 da trilha zero, para começar a montar a cópia do diretório do disco na memória. Para manter a compatibilidade (leitura de um setor de cada vez), é alterado, na fila de mensagens analisadas, o setor a ser acessado relativo a

tarefa em execução. Este processo se repetirá até que todos os setores, da trilha zero, que contenham descritores de arquivos, sejam lidos e processados.

Ao ser analisado o resultado da execução da tarefa (ver módulo "F" da figura (29)), a cópia do diretório do disco será montada na memória do controlador, sendo atualizado o vetor de ocupação de trilhas e o vetor de ocupação de descritores (ver item III.5.2).

Após este procedimento será lido o setor 5 do disco através da inserção de nova ordem de leitura, para salvar possíveis trilhas defeituosas. Isto é feito através da marcação das mesmas, como trilhas ocupadas, no vetor de ocupação de trilhas. Deste modo será evitado que qualquer arquivo receba uma trilha defeituosa.

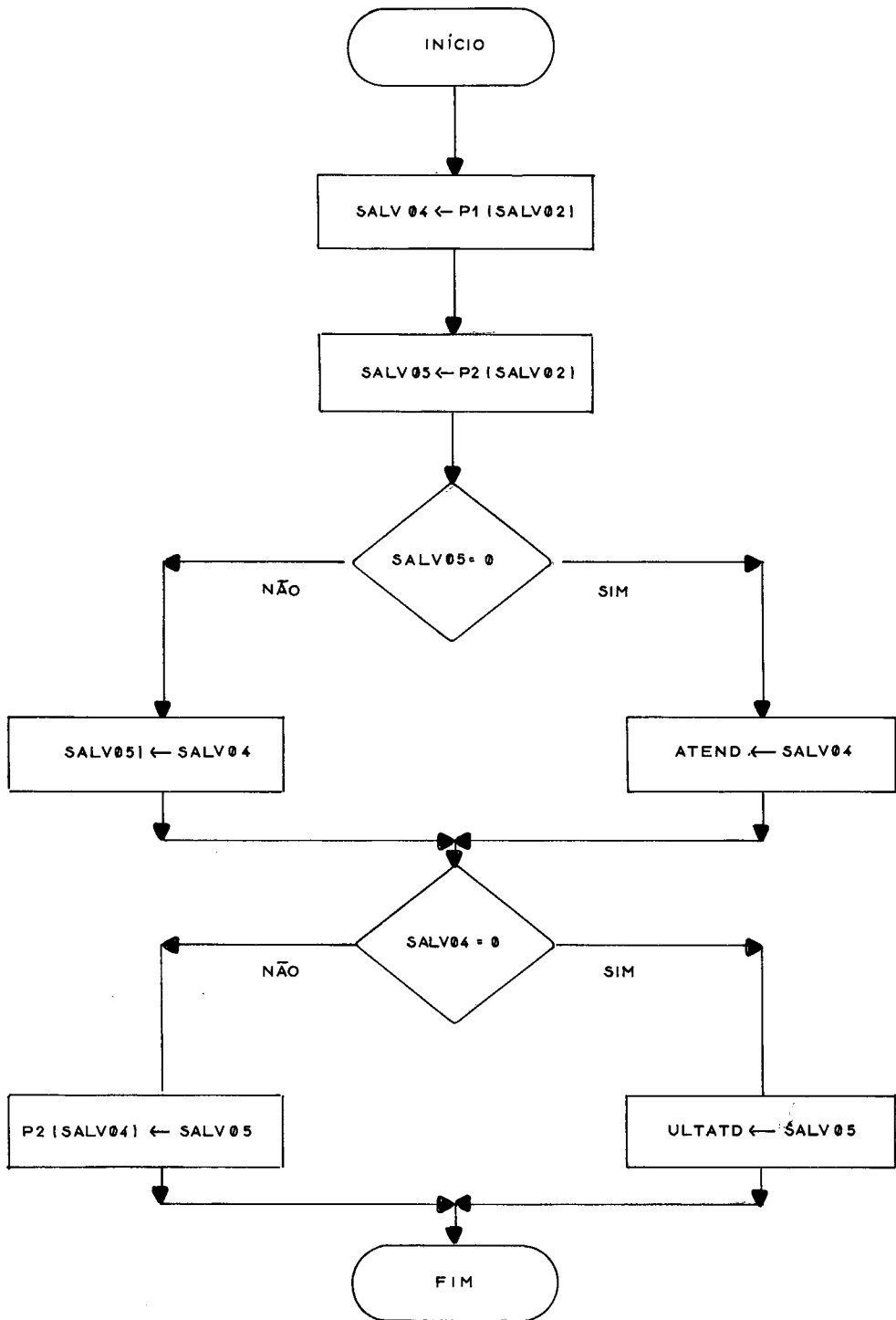
Se o formato do disco for IBM e houver uma trilha defeituosa, esta informação será armazenada e este disco não poderá ser utilizado.

Completada esta operação, a tarefa é considerada concluída. A mensagem existente na "fila de mensagens analisadas" é removida (ver figura (32)) e o buffer é inserido na "fila de buffers disponíveis" (ver figura (34)).

#### IV.5.2

#### Tarefa Inicializa disco

Esta tarefa, cujo texto está definido no item III.2.6, tem a característica particular de poder acessar todas as trilhas de uma unidade de disco flexível.



SALV02 ▣ aponta para a mensagem a ser removida da "fila de mensagens analisadas".

Figura 32 - Rotina para remoção de mensagens da "fila de mensagens analisadas" (REMANA).

Durante a análise do texto da mensagem recebida, o controlador gera uma tarefa de acesso a trilha zero, do disco, com o fim específico de inicialização.

A execução da tarefa faz a inicialização da trilha zero e quando esta termina, se necessário, cria-se uma nova entrada na estrutura de otimização do acesso, apontando para a mesma tarefa, na fila de mensagens analisadas, agora para a inicialização da trilha número 1 do disco. Este processo se repete até que tenha sido inicializada a última trilha do disco.

Desta maneira consegue-se desmembrar uma tarefa única em 76 sub-tarefas, cada uma inicializando uma trilha de disco, com o objetivo de fazer com que todas elas corram igualmente com outras tarefas solicitadas pelo computador mestre.

Para que esta tarefa seja exequível, é necessário que não exista nenhuma outra solicitação de tarefa para a unidade que contém o disco a ser inicializado.

Durante sua execução, o volume do disco não fica disponível, o que impede qualquer outra tarefa para a unidade correspondente.

Após completar a inicialização, a mensagem correspondente a esta tarefa, é retirada da "fila de mensagens analisadas" (ver figura (32)) e inserida, com o resultado, na "fila de mensagens a transmitir" (ver figura (23)).

Se for encontrado algum erro durante a execução, o código do erro é inserido na mensagem e esta, após ser retirada da fila de mensagens analisadas, é inserida na fila de transmissão.

Os erros possíveis de serem encontrados durante sua execução estão relacionados na tabela (1).

VALOR (HEXA)	SIGNIFICADO
C1	tipo de formato diferente de 0 e 1;
C2	número de volume é zero e tipo de formato é 1 ou número de volume não é zero e tipo de formato é zero;
C3	tipo de inicialização diferente de 0 e 1;
C4	número do disco está fora dos limites;
C5	disco a inicializar está sendo utilizado por outro processo,

Tabela 1 - Erros da tarefa "Inicializa disco"

#### IV.5.3 Tarefa abre arquivo

Esta tarefa, cujo texto está definido no item III.2.1, só poderá ser executada se o disco estiver ligado e a cópia do seu diretório estiver montada na memória do controlador.

A descrição do seu funcionamento esta apresentada, em detalhes, no item III.5.2.1.

Caso tenha sido encontrado algum erro ou caso a tarefa tenha sido completada com sucesso, sua mensagem é retirada da "fila de mensagens recebidas" (ver figura (22)) e inserida na "fila de mensagens a transmitir" (ver figura (23)).

Os erros possíveis de serem encontrados durante sua execução estão relacionados na tabela (2).

VALOR (HEXA)	SIGNIFICADO
A1(*)	esgotou o número de entradas na cópia do diretório;
A2(*)	não foi achado o descritor do arquivo aberto;
A3(*)	descritor achado não corresponde a unidade de disco desejada;
A4(*)	contador de processos que mantém o arquivo aberto ficou maior do que 31;
C2	modo de abertura inválido;
CA	tentativa de criar um arquivo cujo nome já existe;
CB	disco com o número desejado não está ligado;
CC	não há área disponível para a criação deste arquivo;
CD	modo de abertura é dois e outro processo está com o arquivo aberto;
CE	o arquivo está aberto para E/S, não pode ser compartilhado;
CF	abertura para arquivo não existente em disco;
D0	nome do arquivo está inválido;
D1	existe mais de um disco ligado que possui o número desejado;

Tabela 2 - Erros da tarefa "Abre arquivo"

(\*) Estes erros são devidos a problemas do "software" e não devem ocorrer em situações normais.



VALOR (HEXA)	SIGNIFICADO
D2	pedido de abertura é para E/S e o forma <u>to</u> é IBM ou o disco está protegido;
D3	não há espaço para armazenar o descri- tor de arquivo no disco (já tem 19).

Tabela 2 - (continuação)

## IV.5.4

## Tarefa faz entrada ou saída

Esta tarefa, cujo texto está definido no item III.2.2, promove as leituras e gravações em arquivos.

Ao ser tentada sua execução, existe a possibilidade de encontrar um erro pré-operativo na fase de análise do texto.

Neste caso a mensagem é retirada da fila de mensagens recebidas e, após a inserção da identificação do erro, a mesma é inserida na fila de mensagens a transmitir.

O texto da mensagem, neste caso, é:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4: CONTERÁ 1 BITS 3, 2, 1 e 0: INDEXADOR DO PROCESSO SOLICITANTE
1	1	BIT 7: LIGADO (INDICA ERRO) BIT 6: NÃO IMPORTA BITS 5, 4, 3, 2, 1 e 0: DÃO O CÓDIGO DO ERRO

Os códigos de erros relativos a esta tarefa, na fase pré-operativa, estão relacionados na tabela (3).

Se não for encontrado nenhum erro pré-operativo, a tarefa seguirá o caminho normal até chegar ao ponto em que realmente será executada (leitura ou gravação). Para isto é enviada uma mensagem, ao mestre, com a finalidade de prepará-lo para a transmissão ou recepção dos dados em modo ADM (esta mensagem é enviada quando a cabeça de leitura/gravação está posicionada na trilha desejada e não existe outra tarefa sendo executada). O texto desta mensagem é:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4: CONTERÁ 1 BITS 3, 2, 1 e 0: INDEXADOR DO PROCESSO SOLICITANTE
1	1	BIT 7: DESLIGADO (INDICA OPERAÇÃO OK) BIT 6: LIGADO (INDICA FASE PRÉ-OPERATIVA) BITS 5, 4, ..., 1 e 0: VALOR 0

Durante a transferência dos dados, entre a memória do computador mestre e disco flexível, executada em ADM, podem ocorrer erros que deverão ser avisados ao computador mestre. Estes erros são os da fase pós-operativa.

A mensagem de sinalização de erros pós-operativos tem a seguinte estrutura:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4: CONTERÁ 1 BITS 3, 2, 1 e 0: INDEXADOR DO PROCESSO SOLICITANTE
1	1	BIT 7: LIGADO (INDICA ERRO) BIT 6: DESLIGADO (INDICA FASE PÓS-OPERATIVA) BITS 5, 4, ..., 1 e 0: DÁ O CÓDIGO DE ERRO

Os códigos de erros relativos a esta tarefa, na fase pós-operativa, estão descritos na tabela (4).

É tarefa do computador mestre tomar a atitude necessária para solucionar o problema, seja solicitando novamente a operação seja interrompendo a execução do processo que gerou a tarefa.

Caso a operação tenha sido concluída com êxito, o computador mestre é notificado com uma mensagem cujo texto é definido abaixo:

BYTE	TAMANHO	DESCRIÇÃO
0	1	BITS 7, 6, 5 e 4: CONTERÁ 1 BITS 3, 2, 1 e 0: INDEXADOR DO PROCES <u>S</u> SO SOLICITANTE
1	1	BIT 7: DESLIGADO (INDICA OPERAÇÃO OK) BIT 6: DESLIGADO (INDICA FASE PÓS-OPE <u>R</u> RATIVA) BITS 5, 4, ..., 1 e 0: VALOR 0

VALOR (HEXA)	SIGNIFICADO
C1	Índice do arquivo não está entre 0 e 15;
C3	número de bytes a transferir não está entre 1 e 128;
C5	modo diferente de 0 e 1;
CA	arquivo indicado não está aberto;
CB	arquivo não está aberto pelo processo solicitante;
CC	arquivo foi aberto para leitura e está sendo tentada uma operação de gravação;
CD	final do arquivo;
CE	pedido para acessar trilha não existente no disco;
CF	número de registro a ler deve ser maior que zero;

Tabela 3 - Erros, na fase pré-operativa, da tarefa "Faz entrada ousada".

VALOR (HEXA)	SIGNIFICADO
8F	operação de E/S com erro (paridade, problemas no disco, etc)

Tabela 4 - Erros, na fase pós-operativa, da tarefa "Faz entrada ou saída".

#### IV.5.5 Tarefa fecha arquivo

Esta tarefa, cujo texto está definido no item III.2.3, só poderá ser executada se o arquivo tiver sido aberto anteriormente.

A descrição do seu funcionamento foi apresentada, em detalhes, no item III.5.2.2.

Caso tenha sido encontrado algum erro ou a tarefa tenha sido completada com sucesso, sua mensagem será retirada da "fila de mensagens analisadas" (ver figura (32)) e inserida na "fila de mensagens a transmitir" (ver figura (23)).

Os erros possíveis durante sua execução, são os relacionados na tabela (5).

#### IV.5.6 Tarefa remove processos

Esta tarefa está definida em detalhes no item III.2.4.

Como sua execução não exige acesso a disco, a mesma é totalmente realizada na rotina de análise correspondente (similar a tarefa "abre arquivo"). Após sua execução a mensagem é retirada da fila de mensagens recebidas e inserida na fila de mensagens a transmitir.

A característica básica desta tarefa é que, caso não se encontre nenhum arquivo sendo utilizado pelos processos relacionados, nada será executado. Deste modo, não há

VALOR (HEXA)	SIGNIFICADO
01	arquivo será removido quando não houver mais nenhum processo que o esteja utilizando (apenas um aviso - não é erro);
C2	índice do arquivo não está entre 0 e 15;
C3	modo de fechamento diferente de 0 e 1;
CA	volume não está presente;
CC	arquivo indicado não foi aberto;
CD	arquivo indicado não foi aberto por este processo;
CE	remoção não executada por problemas de E/S;
CF	criação não executada por problemas de E/S.

Tabela 5 - Erros da tarefa "Fecha Arquivo".



erros previstos durante sua execução.

#### IV.5.7 Tarefa carrega módulo

Esta tarefa está descrita, em detalhes, no item III.2.5.

Sua execução segue os mesmos padrões da tarefa "faz entrada e saída".

#### IV.6 Transmissão das Respostas ao Computador Mestre

A transmissão de mensagens para o computador mestre é feita do modo byte-a-byte.

O controlador ao verificar a existência de uma mensagem a transmitir, acerta os apontadores para a transmissão da mensagem e grava no registrador de comunicação o caracter "SOH" para iniciar a transmissão.

É importante salientar que não será possível começar uma transmissão de mensagem se o controlador já estiver recebendo ou transmitindo mensagem ou dados.

A gravação do primeiro byte no registrador de comunicações do controlador provoca uma interrupção no computador mestre. Quando o mestre responder com o eco, o controlador será interrompido e os outros bytes da mensagem serão transmitidos na própria rotina de interrupção, de acordo com o au-

tomato mostrado na figura (10).

Se a mensagem foi transmitida com sucesso, a mesma é retirada da "fila de transmissão" (ver figura (33)) e inserida na "fila de buffers de recepção" (ver figura 34)).

Deste modo o ciclo do controlador, desde a recepção até a resposta da tarefa executada, é completado.

Se houver qualquer erro durante a transmissão de uma mensagem (eco recebido diferente do esperado), automaticamente a mensagem será retransmitida.

#### IV.7 Procedimento Executado quando um Disco é Desligado

Normalmente quando se desliga um disco é porque não se deseja mais trabalhar com ele ou porque o sistema será desligado.

Entretanto, podem ocorrer casos em que por pane na unidade ou por erro do operador, um disco seja desligado incorretamente.

Desta maneira deve haver um procedimento que funcione para qualquer tipo de ocorrência.

O procedimento, a seguir descrito, é executado toda vez que um disco é desligado do sistema:

1. Se o controlador estava executando uma tarefa para este disco, a mensagem correspondente é retirada da "fila de mensagens analisadas", é colocado o código da ocorrência nesta mensagem e, finalmente, a mensagem é inserida na "fila de

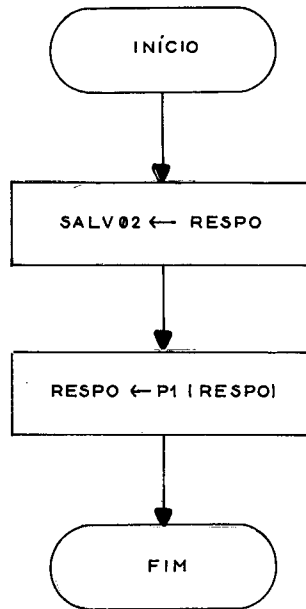
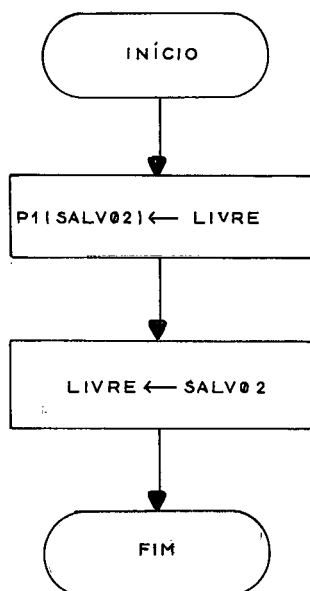


Figura 33 - Rotina de remoção de uma mensagem da "fila de mensagens a transmitir" (REMRSP).



SALV02 = aponta para o buffer a ser liberado.

Figura 34 - Rotina de liberação de área de mensagem (coloca o buffer na "fila de buffers de recepção") (LIBMEN).

mensagens a transmitir".

2. Se existirem tarefas esperando execução, relativas a este disco, na estrutura para otimização do acesso, essas tarefas serão removidas, com suas correspondentes mensagens, da "fila de mensagens analisadas". Estas mensagens, após receberem o código da ocorrência, serão inseridas na "fila de mensagens a transmitir".
3. As informações sobre o disco (volume, cópia do diretório, ocupação de trilhas, ocupação de descritores, etc) são apagadas, na memória do controlador.

As mensagens que estiverem na "fila de mensagens recebidas", ao serem analisadas, se estiverem solicitando tarefas para o disco desligado, automaticamente acusarão erro e serão colocadas na "fila de mensagens a transmitir".

Deste modo assegura-se que, mesmo em caso de engano ao desligar um disco, todas as mensagens sejam respondidas. Caso isto não acontecesse, poderia acontecer uma parada de processos, no computador mestre, sem possibilidades de se sair desta situação.

## V. CONCLUSÕES

Neste capítulo são consideradas algumas modificações que poderiam ser introduzidas no "software", a fim de aperfeiçoar o funcionamento do Sistema.

A utilização efetiva do Sistema, permitirá, após algum tempo de uso, uma melhor avaliação dos pontos a serem modificados e dos acréscimos a serem efetuados. Certamente é conveniente uma avaliação mais criteriosa do comportamento do "software" antes de serem introduzidas modificações no mesmo.

A memória disponível para o controlador, atualmente, é de 8KB, estando sendo projetada uma expansão para 12KB.

Para controlar a utilização de duas unidades de disco flexível, o "software" completo, incluindo as áreas de "buffers", ocupa, aproximadamente, 9KB. Com a organização atual, cada unidade adicional ligada ao sistema, provoca um aumento de 500 bytes no tamanho do "software". Da mesma maneira, cada unidade retirada do Sistema, ocasiona uma diminuição de 500 bytes no tamanho do "software".

Caso não se trabalhasse com cópias dos diretórios dos discos, na memória do controlador, cada disco adicionado aumentaria de 200 bytes o tamanho do "software". Isto implicaria em uma boa economia de memória no caso de ter-se várias unidades de disco ( $N \times 300$  bytes, para  $N$  unidades de disco).

Entretanto, neste caso, qualquer pedido de

abertura ou fechamento de arquivo ocasionaria um acesso às unidades de disco, o que tornaria o tempo de resposta mais lento. Dependendo da aplicação (por exemplo utilização do Sistema em uma aplicação de entrada de dados) esta demora adicional talvez não fosse notada.

A tarefa "inicializa disco", por ser muito pouco utilizada em operação normal do sistema, poderia ser eliminada do "software". Para a remoção de todos os arquivos de um disco, por exemplo, poderia ser utilizada, em seu lugar, a tarefa "fecha arquivo", com opção de remoção. Com isto se conseguiria uma redução de 900 bytes no tamanho do "software". Outra alternativa seria modificar o "software" para que a rotina "inicializa disco" fosse carregada de disco, dinamicamente, quando necessário.

As facilidades que o Sistema oferece para manipulação de arquivos e volumes poderiam ser estendidas com a introdução de novas tarefas como:

- Obtém Diretório de Volume:

que permitiria obter informações sobre os arquivos contidos em um determinado disco;

- Salva Trilhas Defeituosas:

que permitiria salvar as trilhas defeituosas em um disco. Esta tarefa já está prevista no "software", pela utilização do setor 5 da trilha zero, mas não foi implementada;

- Compacta Arquivos no Volume:

que permitiria compactar os arquivos em um disco, eliminando os intervalos deixados pelos arquivos que tivessem sido removidos.

Outra característica que poderia ser introduzida no Sistema diz respeito ao controle do fim lógico de arquivos. Para esse controle, deveriam ser mantidas as seguintes informações adicionais em cada entrada da tabela "cópia dos diretórios dos discos" (na memória):

- fim lógico atual de trilha (1 BYTE);
- fim lógico atual de setor (1 BYTE);
- fim lógico alterado de trilha (1 BYTE);
- fim lógico alterado de setor (1 BYTE).

No gerenciamento dos arquivos, as seguintes operações adicionais deveriam ser introduzidas:

1. Ao ser aberto um arquivo:

- fim lógico alterado de trilha ← fim lógico atual de trilha;
- fim lógico alterado de setor ← fim lógico atual de setor.

2. Ao ser gravado um registro qualquer em um arquivo:

se sua posição é superior ao "fim lógico alterado", altera-se o "fim lógico alterado" para este novo valor.

3. Ao ser fechado um arquivo:

se "fim lógico alterado" for maior do que "fim lógico atual", faz-se:

- fim lógico atual de trilha ← fim lógico alterado de trilha;
- fim lógico atual de setor ← fim lógico alterado de setor.

além disso, atualiza-se no descritor do arquivo em disco, através de acesso à trilha zero (que contém o diretório), o valor correspondente ao fim lógico do arquivo.



BIBLIOGRAFIA

1. SUZIM, Altamiro Amadeu. Controlador inteligente de disco flexível. Porto Alegre, CPGCC/UFRGS, 1977.
2. SCHLABITZ, Francisco F. & CORSO, Thadeu B. Um sistema multiprogramado para minicomputador. Porto Alegre, CPGCC/UFRGS, 1978.
3. HOARE, C.A.R. Monitors: an operating system structuring concept. USA, Communications of the ACM, 17(10):549-557, OCT. 1974.
4. MOTOROLA INC. M6800 microprocessor applications manual. USA, Motorola, 1975.
5. KNUTH, Donald E. The art of computer programming. USA, Addison-Wesley, Vol. 1, 1973.
6. PEATMAN, John B. Microcomputer based design. USA, McGraw-Hill, 1977.
7. IBM. The IBM diskette for standard data interchange. USA, IBM, 1973.
8. CALCOMP. Model 140 floppy disk drive. USA. CALCOMP, 1975.