

SISTEMAS DIGITAIS

UM CONTROLADOR E FORMATADOR DE
DISCO FLEXÍVEL

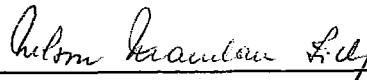
PAULO HENRIQUE DE AGUIAR RODRIGUES

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.).

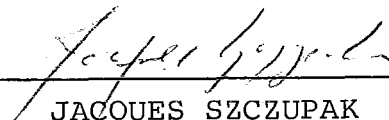
Aprovada por:



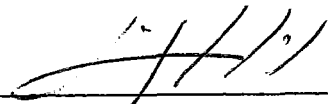
DIOGO FUJIO TAKANO



NELSON MACULAN FILHO



JACQUES SZCZUPAK



LUIS PEREIRA CALOBA

RIO DE JANEIRO, RJ - BRASIL

JULHO DE 1977

RODRIGUES, PAULO HENRIQUE DE AGUIAR
Sistemas Digitais: Um Controlador e For
matador de Disco Flexível (Rio de Janeiro)
1977.

ix, 166 p. 28,0cm (COPPE/UFRJ - M.Sc.)
Tese - Universidade Federal do Rio de Ja
neiro. Núcleo de Computação Eletrônica

1. Controlador; I. COPPE/UFRJ; II. Títu-
lo (série).

A G R A D E C I M E N T O S

Aos amigos Newton Faller e Eber Assis Schmitz pela primeira orientação neste trabalho e ao colega José Carlos Vida Cura pela participação na definição do software do sistema.

R E S U M O

É apresentada a implementação de um Controlador e Formatador de Disco Flexível para o Terminal Inteligente do NCE/UFRJ capaz de gerenciar até duas unidades Calcomp-14Ø.

O meio é o diskette IBM ou equivalente, codificação em frequência dupla, sendo utilizados bytes especiais (address marks) para demarcar campos sobre a trilha e os setores, de tamanho variável, são identificados logicamente. Ao lado de um conjunto amplo de instruções e estados que propiciam uma completa manipulação e recuperação da informação, o uso de microprogramação para realizar as operações de escrita resultou em uma versatilidade bastante grande na formatação das trilhas, com aproveitamento de até 81%.

Após uma descrição completa do software, o hardware, implementado com lógica TTL e a um custo de componentes da ordem de Cr\$ 4.000,00, é minuciosamente detalhado.

Ao final encontramos algumas considerações a programas de teste de vida e performance do sistema.

A B S T R A C T

The thesis presents the implementation of a Floppy Disk Controller/Formatter for the NCE's Intelligent Terminal (TI), capable of controlling a Calcomp-140 two drives configuration.

The magnetic mean is the IBM diskette or equivalent specified with single density, double frequency (FM) codification, special bytes or address marks identifying fields in the track and soft sectors of variable length. In addition to a full set of instructions and states that permits a complete data management and recovery, the use of microprogramming to perform write operations results in extended formatter versatility, with efficiencies up to 81%.

After a software description, the hardware, implemented with TTL logic and components costs around Cr\$ 4.000,00, is fully detailed.

Finally, there are some considerations about time life and performance of the system.

Í N D I C E

	<u>PÁGINA</u>
1 - INTRODUÇÃO	1
2 - O MEIO E SUA ORGANIZAÇÃO	5
3 - UNIDADE CALCOMP-14Ø	13
3.1 - Características Técnicas	13
3.2 - Sinais da Interface	15
3.2.1 - SINAIS DE SAÍDA	15
3.2.1.1 - RAWDATA	15
3.2.1.2 - INDEX	15
3.2.1.3 - TRACKØ	15
3.2.1.4 - READY	16
3.2.1.5 - WRITE ENABLE	16
3.2.2 - SINAIS DE ENTRADA	16
3.2.2.1 - SELECT	16
3.2.2.2 - WRITE ENABLE	16
3.2.2.3 - WRITE DATA	16
3.2.2.4 - ABOVE TRACK 43	16
3.2.2.5 - STEP	17
3.2.2.6 - DIRECTION	17
3.2.2.7 - HEAD LOAD	17
3.2.3 - TERMINAÇÃO DOS SINAIS	18
4 - ENTRADA/SAÍDA DO TERMINAL INTELIGENTE	22
4.1 - Controle e Seleção de Periféricos	22
4.2 - Canal	27

	<u>PÁGINA</u>
5 - DEFINIÇÃO DO SOFTWARE	32
5.1 - Estados	32
5.1.1 - ESTADOS DO CONTROLADOR	33
5.1.1.1 - PERIFÉRICO PRESENTE	33
5.1.1.2 - OCUPADO	33
5.1.2 - ESTADOS DA UNIDADE	33
5.1.2.1 - UNIDADE SELECIONADA	34
5.1.2.2 - UNIDADE NÃO-PRONTA	34
5.1.2.3 - TRILHA ZERO	35
5.1.2.4 - ESCRITA PROTEGIDA	35
5.1.3 - ESTADOS RELACIONADOS	35
5.1.3.1 - INSTRUÇÃO REJEITADA (IR)	36
5.1.3.2 - ERRO DE DIMENSÃO (ERD)	36
5.1.3.3 - ERRO DE CRC NA IDENTIFICA- ÇÃO (ERS)	36
5.1.3.4 - ERRO DE CRC NA INFORMAÇÃO (ERI)	37
5.1.3.5 - TRILHA INCORRETA (TRI)	37
5.1.3.6 - FALTA MARCA ESPECIAL (FME)	38
5.1.3.7 - SETOR NÃO-ENCONTRADO (SNE)	38
5.1.3.8 - INFORMAÇÃO ELIMINADA (IEL)	38
5.1.3.9 - OPERAÇÃO INCORRETA (OPI)	38
5.2 - Instruções	39
5.2.1 - IMEDIATAS	39
5.2.1.1 - RESET	40
5.2.1.2 - TAMANHO DO SETOR (TAS)	40
5.2.1.3 - SETOR (SET)	40

	<u>PÁGINA</u>
5.2.1.4 - SELECIONA (SEL)	41
5.2.2 - POSICIONAMENTO (POS)	41
5.2.3 - LEITURA/ESCRITA	44
5.2.3.1 - LE SETOR (LES)	45
5.2.3.2 - LE IDENTIFICAÇÃO (LEI)	45
5.2.3.3 - LE TODAS IDENTIFICAÇÕES (LETI)	46
5.2.3.4 - TRANSFERE CAMPO DE INFORMA- ÇÃO (TCI)	46
5.2.3.5 - ESCREVE SETOR (ESC)	47
5.2.3.6 - ELIMINE INFORMAÇÃO ZERANDO (ELIZ)	47
5.2.3.7 - ELIMINE INFORMAÇÃO (ELI)	47
5.2.4 - FORMATAÇÃO (FOR)	48
5.2.4.1 - RESTRIÇÃO AOS PREÂMBULOS	49
5.2.4.2 - RELAÇÃO ENTRE O NÚMERO DE SE- TOR E O TÁS PARA APROVEITAMEN TO MÁXIMO	53
5.2.4.3 - MICROINSTRUÇÕES	55
6 - IMPLEMENTAÇÃO DO HARDWARE	62
6.1 - Decodificador de Instruções	72
6.2 - Registros de Armazenamento: Setor e Tamanho do Setor	78
6.3 - Posicionamento	79
6.4 - Controle da Cabeça	90
6.5 - Sincronismo	92
6.5.1 - SEPARADOR PLL	93

	<u>PÁGINA</u>
6.5.2 - CODIFICADOR	109
6.5.3 - DETEÇÃO DO ENABLE	111
6.6 - Contador de Bits em Leitura (CBITS)	114
6.7 - Controle da Escrita	117
6.7.1 - ESTADO DE ESCRITA	117
6.7.2 - RELÓGIO E CODIFICADOR	122
6.7.3 - CONTADOR DE BITS	124
6.7.4 - SEQUENCIADOR E MATRIZ DE DIODOS	126
6.7.5 - DECODIFICADOR DAS MICRO	128
6.8 - Contador de Bytes (CBYTES)	131
6.9 - Registro de Informação	136
6.10 - Registro de Clock	139
6.11 - Cheque de Setor	141
6.12 - Cheque de Trilha	142
6.13 - Detecção das Marcas Especiais	143
6.14 - Geração do CRC	144
6.15 - Controle de Transferências	150
6.16 - Estados	154
7 - Considerações Finais	157
B I B L I O G R A F I A	159
A P Ê N D I C E	160

Í N D I C E D E F I G U R A S

PÁGINA

CAPÍTULO 2:

FIG. 1 - FORMATO DO SETOR IBM	12
FIG. 2 - FORMATO DA TRILHA IBM	12

CAPÍTULO 3:

FIG. 3 - DIAGRAMA DE TEMPO POS/TRANSF	19
FIG. 4 - DIAGRAMA DE TEMPO PARA ESCRITA	20
FIG. 5 - DIAGRAMA DE TEMPO PARA LEITURA	21

CAPÍTULO 4:

FIG. 6 - ARQUITETURA DO TERMINAL INTELIGENTE	23
FIG. 7 - INTERFACE PADRÃO	26
FIG. 8 - DIAGRAMA DE TEMPO DO CANAL	31

CAPÍTULO 6:

FIG. 9 - FLUXO GERAL	71
FIG. 10 - DECODIFICAÇÃO DE INSTRUÇÕES	74
FIG. 11 - CIRCUITO DE POSICIONAMENTO	80
FIG. 12 - SEPARADOR PLL - Diagrama de Blocos	99
FIG. 13 - SEPARADOR PLL - Diagrama de Tempo	102
FIG. 14 - SEPARADOR PLL - Esquema Detalhado	103
FIG. 15 - CODIFICADOR	110
FIG. 16 - CONTADOR DE BITS - Diagrama de Tempo	116
FIG. 17 - GERAÇÃO DO CRC	149

1 - INTRODUÇÃO

Este trabalho de tese apresenta o projeto e a implementação de um Controlador e Formatador de Disco Flexível como periférico para o Terminal Inteligente desenvolvido no Núcleo de Computação Eletrônica da UFRJ.

O disco flexível ou "Floppy Disk" é uma memória periférica barata (~\$5/unidade) de capacidade média de armazenamento (300 KBytes) e de tempo de acesso relativamente baixo (249ms). A sua facilidade de transporte e armazenamento tornaram-se em pouco tempo uma constante no mercado e elemento indispensável principalmente em sistemas de Entrada de Dados (DATA ENTRY) onde se encontra, atualmente, a sua mais forte aplicação. A previsão do seu uso é bem mais ampla como se -
der da tabela abaixo, relacionando as previsões de unidades vendidas para 1975, 1978 e 1980 nas diversas aplicações (Modern Data, August/75):

MERCADO DE FLOPPIES		(MILHARES DE UNIDADES)		
USO		1975	1978	1980
Data Entry		12.8	21	32
Small Business Systems		3.3	12	20
Inteligente Terminal/Remote				
Batch		10.2	28	36
Point of Sale		.5	10	22
Programmable Calculators		.7	5	8
Word Processing		.8	7	10
Control and Test Systems		.5	4	11
Microcomputers		3.1	16	8
System Loader		1.4	2	3
Total	U.S.	33.3	105	150
Total	Non U.S.	18.4	68	126
Total	World	51.7	173	276

Sendo o Terminal Inteligente do NCE um microcomputador que se aplica a maioria das aplicações acima, justifica-se plenamente o desenvolvimento de um Controlador/Formatador de Disco Flexível. Baseou-se o projeto em quatro (4) pontos básicos:

a - Compatibilidade com o Mercado

Para que o sistema pudesse ser intercambiado com outros já existentes, procurou-se adotar as codificações "standards" para o mercado de "Floppy Disks". Daí a utilização dos setores lógicos, frequência dupla e marcas especiais (Address Marks) para delimitação dos campos.

b - Potencialidade

O conjunto de instruções a definir deveria ser completo para a manipulação e recuperação da informação.

c - Versatilidade

Além de manter a compatibilidade com os sistemas "standards" (IBM) que só utilizam um aproveitamento de 68% da trilha, o CFDF deveria ser capaz de trabalhar com outras formatações e tamanho de setores variáveis de modo a conseguir um melhor aproveitamento da trilha e uma utilização mais racional para determinadas aplicações.

d - Confiabilidade

A confiabilidade acaba surgindo da utilização da gravação

em frequência dupla (FM). Um sistema de comparação de fase permite que a recuperação e separação da informação seja absoluta.

O Controlador tem a possibilidade de manipular até duas unidades CALCOMP-14Ø, embora uma expansão futura seja prevista, sem grandes alterações no hardware e no software.

Inicialmente é dada uma descrição do meio, sua organização, da codificação da informação e dos bytes especiais de sincronismo (CAPÍTULO 2).

No CAPÍTULO 3 são apresentadas as características técnicas e de interfaceamento da unidade CALCOMP-14Ø, justificando, paralelamente, a sua escolha para o sistema.

O Controlador é um periférico rápido do TI e, como tal, utiliza o Canal de Acesso Direto à Memória. A descrição do funcionamento deste e do sistema de E/S do TI é fornecida, mostrando-se, também, a Interface Padrão utilizada por todo e qualquer periférico (CAPÍTULO 4).

A escolha das instruções e estados foi o primeiro passo na definição do sistema. A partir daí, procurou-se implementar o hardware de maneira a satisfazer o desempenho pré-estabelecido. De igual forma, a descrição do software (CAPÍTULO 5) antecede a descrição do hardware (CAPÍTULO 6). Para quem se interessar apenas pelos aspectos de programação da máquina, a descrição do software, é completa e satisfaz plenamente, sem que haja necessidade de consulta aos diagramas de lógica do

CAPÍTULO 6. A descrição do hardware parte do Fluxo Geral (FIG. 9), aprofundando-se, passo-a-passo, em cada um dos blocos principais, com o funcionamento deles detalhado por diagramas de tempo e esquemas de circuito simplificados.

As Considerações Finais (CAPÍTULO 7) abordam questões relacionadas com o teste e depuração do sistema, fatores de custo e performance.

No Apêndice encontramos algumas convenções adotadas no texto e uma listagem de um programa de formatação, programado/escrito em Linguagem ASSEMBLER do Terminal Inteligente (TI).

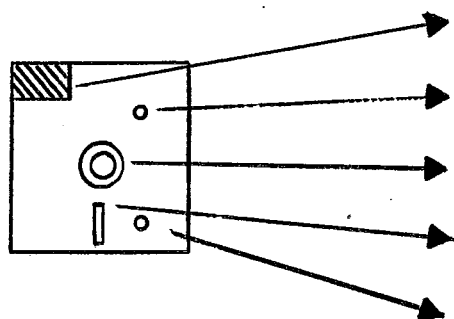
2 - O MEIO E SUA ORGANIZAÇÃO

O DISKETTE consiste de um disco magnético flexível inserido num invólucro de papel. Este invólucro possui um recobrimento interno que minimiza as descargas de eletricidade estática e serve como um meio de baixa fricção para o disco magnético. Poderá ser usado qualquer disco que satisfaça às especificações do "standard" IBM.

É previsto que o disco suporte um número mínimo de 200.000 acessos por trilha, sendo uma trilha considerada gasta se o nível do sinal cai a 20% do seu valor nominal.

As condições de temperatura podem variar entre 10,0°C e 51,5°C. É necessário que a unidade e o disco estejam nas mesmas condições de temperatura antes de sua utilização para se evitar choque térmico.

Fisicamente o disco tem a forma abaixo:



etiqueta

detetor do início da trilha

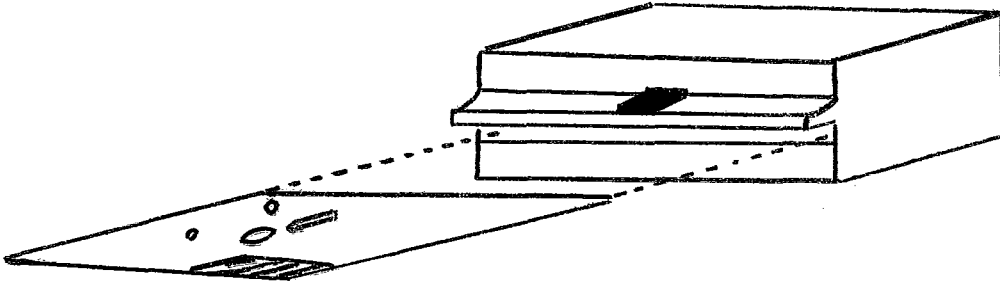
orifício central (tração)

rasgo para acesso da cabeça

proteção contra escrita

Para o seu manuseio e armazenamento devem ser observadas todas as precauções recomendadas pelo fabricante.

A correta inserção na unidade é mostrada na figura abaixo:



a - O primeiro lado a entrar é o da ranhura e do orifício de proteção da escrita;

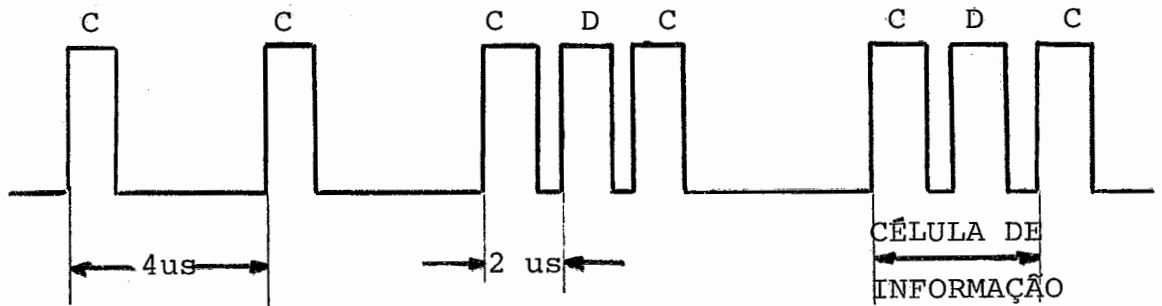
b - O lado no qual está afixada a etiqueta deve estar faceando a tampa da abertura.

O disco flexível compatível IBM é denominado Diskette. De agora em diante passarei a me referenciar somente a este, pois é o único aceito pelo sistema.

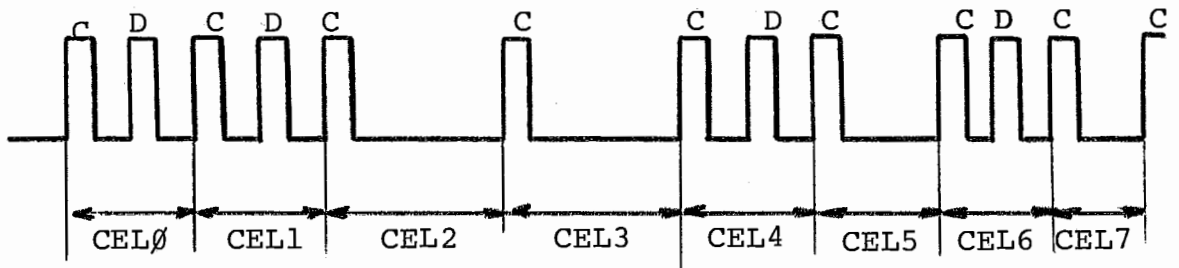
O diskette contém 77 trilhas concêntricas numeradas de 0 a 76, sendo a trilha 0 a mais externa. Esta ordenação é fixa e não pode ser alterada.

A técnica de gravação é a dupla frequência (FM). Sobre toda a trilha são gravados bits de sincronismo (CLOCK) a intervalos de 4 microsegundos. Cada bit de clock define uma célula

la de informação. A existência ou não de um pulso no meio da célula caracteriza um bit de informação 1 ou 0.



BYTE - um byte, quando referindo-se a uma informação sêrie, é definido como 8 células consecutivas. A célula mais significativa é definida como 0 e a menos significativa é definida como 7. Durante uma operação de escrita a célula 0 de cada byte é transferida primeiro com a célula 7 sendo transferida por último. Correspondentemente o bit mais significativo da informação é gravado primeiro, e o menos significativo é gravado por último.



Repre
sentã
ção:

BITS DE

D A D O: 1 1 0 0 1 0 1 0

H E X A:

C

A

BITS DE
CLOKS :

1 1 1 1 1 1 1 1

H E X A:

F

F

Convém ressaltar que a existência de um pulso é fisicamente sentida como uma transição no fluxo. A alteração do fluxo corresponde à transição positiva do pulso.

Com a finalidade de se criar campos na trilha com função específica, foram definidos bytes com uma configuração especial que individualizam e tornam possível, desde que antecedidos por um determinado número de bytes zero, a sincronização com o início de cada campo.

Estas MARCAS ESPECIAIS (ME) são as definidas na formatação IBM e conhecidas como ADDRESS MARKS (AM):

ID ADDRESS MARK ou MARCA ESPECIAL DE IDENTIFICAÇÃO (MEID)

Dado: FE

Clock: C7

DATA ADDRESS MARK ou MARCA ESPECIAL DE INFORMAÇÃO VÁLIDA (MEIV)

Dado: FB

Clock: C7

DELETED DATA ADDRESS MARK ou MARCA ESPECIAL DE INFORMAÇÃO ELIMINADA (MEIE)

Dado: F8

Clock: C7

Estas três (3) marcas são usadas pelo Controlador.

Existe mais uma marca que só é utilizada na Formatação IBM.

É a MARCA ESPECIAL DE INDEX ou INDEX ADDRESS MARK (MEIX):

Dado: FC

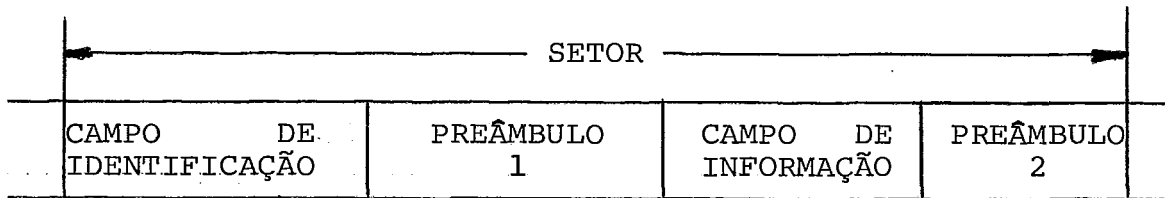
Clock: D7

Esta marca não tem nenhuma função dentro do meu sistema e em bora possa ser gravada por ele, não é detetada.

Utilizando-se dessas marcas é possível dividir a trilha em campos endereçáveis chamados setores.

FORMATO DOS SETORES

Cada setor possui dois (2) campos bem definidos e distintos seguidos de um preâmbulo de ZEROS segundo o formato abaixo:



a - Campo de Identificação

É um conjunto de sete (7) bytes:

PRIMEIRO BYTE - MEID

SEGUNDO BYTE - NÚMERO DA TRILHA

TERCEIRO BYTE - ZEROS

QUARTO BYTE - NÚMERO DO SETOR

QUINTO BYTE - ZEROS

SEXTO BYTE - PRIMEIRO BYTE DO CRC

SÉTIMO BYTE - SEGUNDO BYTE DO CRC

OBS: - A integridade da informação contida num campo (entendendo-se por informação, neste caso, o con-

junto de bits de dados que englobam o primeiro e mais significativo bit da ME, até o último bit de dado gravado) é garantida por dois bytes de cheque de erro, gerados por um Código de Redundância Cíclica (CRC). Embora a geração dos bytes do CRC seja simples em termos de circuito, existe por trás desta simplicidade, um estudo matemático complexo, que garante, dependendo do polinômio usado na geração do código, uma probabilidade bastante grande na detecção de qualquer troca envolvendo um número arbitrário de bits. Uma explicação detalhada sobre CRC pode ser encontrada no Capítulo 6, quando trata-se da implementação do Registro de CRC.

b - Preâmbulo-1

É um conjunto de 17 zeros necessários à detecção das Marcas de Informação do campo seguinte.

c - Campo de Informação

O número de bytes deste campo depende do tamanho de setor utilizado. É formado por:

PRIMEIRO BYTE	- MEIV ou MEIE
SEGUNDO BYTE AO ANTEPENÚLTIMO	- INFORMAÇÃO GRAVADA NO SETOR
PENÚLTIMO BYTE	- PRIMEIRO BYTE DO CRC
ÚLTIMO BYTE	- SEGUNDO BYTE DO CRC

d - Preâmbulo-2

É um conjunto de bytes ZERO, necessários à detecção da próxima Marca de Identificação. Deve ser considerada a flutuação da velocidade de rotação do diskette e a variação da frequência de gravação que podem, durante operações de reescrita, alterar o espaço físico ocupado pelo campo de informação anterior, variando o tamanho deste preâmbulo no sentido inverso.

FORMATO DA TRILHA

A trilha tem o formato abaixo:

PREÂMBULO INICIAL	SETOR-1	SETOR-2	SETOR-N	PREÂMBULO FINAL
----------------------	---------	---------	-------	---------	--------------------

a - Preâmbulo Inicial

É necessária a detecção da primeira marca de identificação.

b - Setores

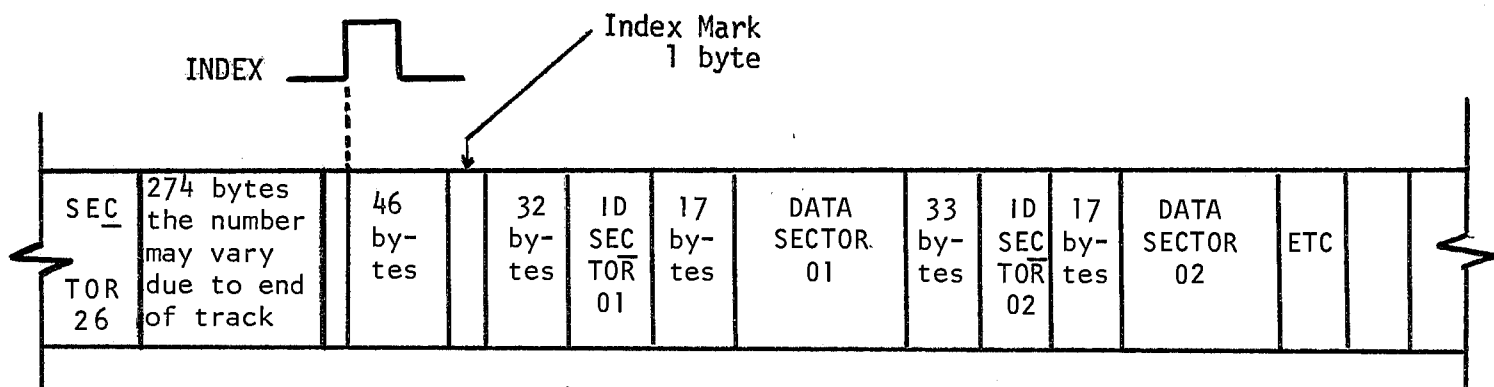
Seguem o formato descrito anteriormente.

c - Preâmbulo Final

É necessário como margem de segurança devido às flutuações de velocidade e variações na frequência de gravação.

Devido aos fatores já citados e ao método utilizado na detecção das Marcas Especiais (ME), restrições são impostas ao ta-

FIG. 1 - FORMATO DA TRILHA IBM.



manho mínimo de cada preâmbulo. Estes pontos serão abordados na descrição da instrução FOR, Capítulo 5.

As figuras 1 e 2 mostram a formatação da trilha utilizada pela IBM nos sistemas 3740.

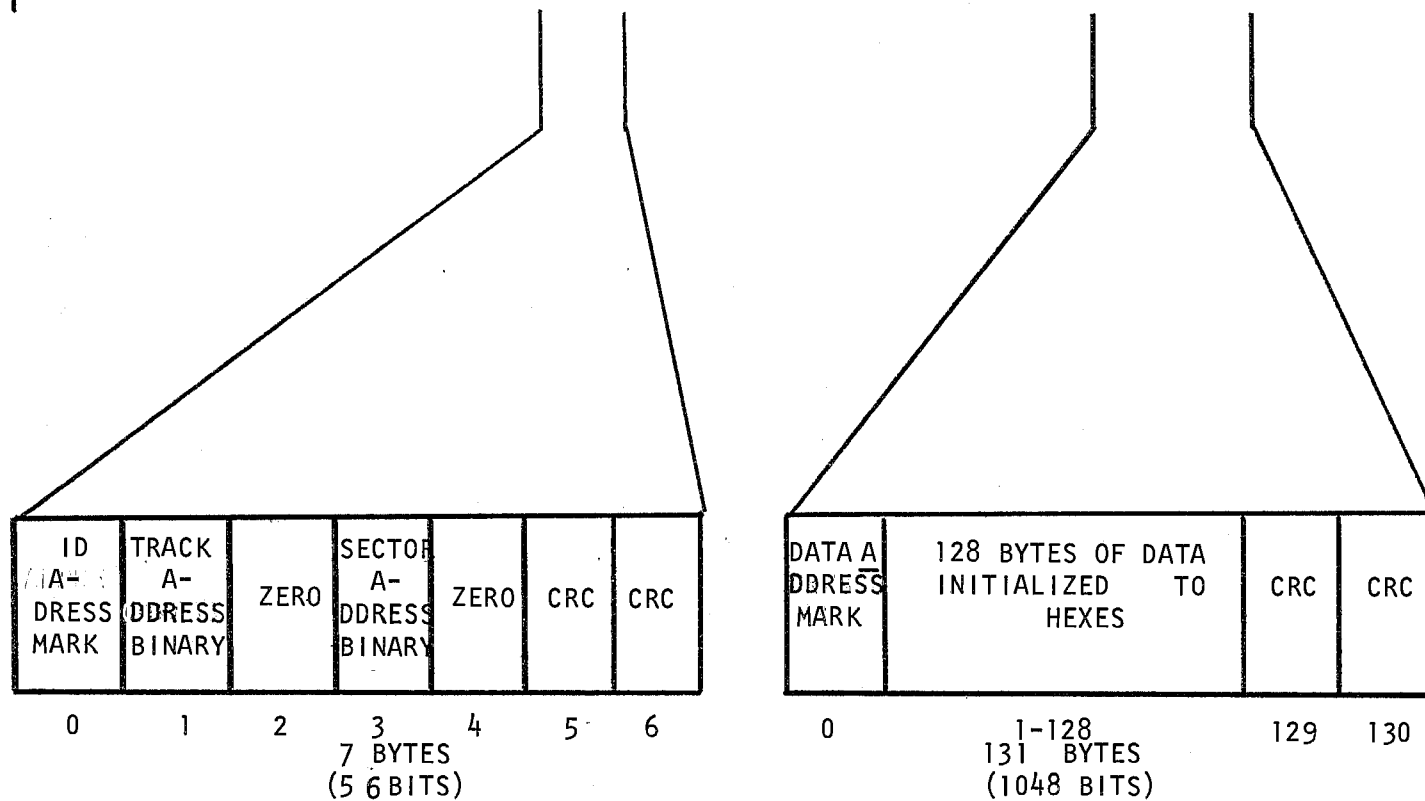


FIG. 2 - FORMATO DO SETOR IBM.

3 - A UNIDADE CALCOMP-140

3.1 - Características Técnicas

De todos os "drives" encontrados no mercado OEM internacional, o CALCOMP-140 era, na época de sua aquisição, o que de melhor havia em termos de performance dentre os compatíveis IBM. O custo unitário em 1975 era de \$670 em pequena escala.

As suas especificações técnicas são as abaixo:

NÚMERO DE CABEÇAS:	1
NÚMERO DE TRILHAS:	77
MODO DE GRAVAÇÃO:	Dupla Frequência (FM)
DENSIDADE DE GRAVAÇÃO:	3200 BPI (na trilha mais interna)
CAPACIDADE NÃO-FORMATADO	
Por DISCO FLEXÍVEL:	3.208.128 bits/401.016 bytes
Por TRILHA:	41.664 bits/5.208 bytes
FORMATO IBM	
Por DISCO FLEXÍVEL:	1.943.552 bits/242.944 bytes
Por TRILHA:	26.624 bits/3.328 bytes
Por SETOR:	1.024 bits/128 bytes
TRILHAS POR POLEGADA:	48
VELOCIDADE DE TRANSFERÊNCIA:	250.000 bits/s, nominal
MECANISMO/POSICIONAMENTO DA CABEÇA:	STEPPER MOTOR

TEMPO DE POSICIONAMENTO:	6 ms/trilha
TEMPO DE ESTABILIZAÇÃO DA CA BEÇA:	10 ms.
TEMPO DE ABAIXAMENTO DA CABE ÇA:	16 ms.
MEIO:	IBM equivalente.
VELOCIDADE DE ROTAÇÃO:	360 RPM \pm 25%
TEMPO DE ESTABILIZAÇÃO DO MO TOR:	2 s máximo.
DISSIPACÃO:	540 BTU/hora.
CONFIABILIDADE:	
Erros de LEITURA:	
Erros recuperáveis	Menos que $\frac{1}{1}$ em 10^9 bits
Erros irrecuperáveis	Menos que $\frac{1}{1}$ em 10^{12} bits
Erros de POSICIONAMENTO:	Menos que $\frac{1}{1}$ em 10^6 aces sos.
MTBF:	Excede 5.000 h.
MTR:	Menos que 30 minutos.

Dentre estas características convém ressaltar o ótimo tempo de posicionamento trilha a trilha (6 ms.) e o tempo relativamente curto de estabilização da cabeça. Definindo:

TEMPO DE ACESSO (T_a) = multiplica-se 1/3 do número de trilhas pelo tempo de acesso entre trilhas, somando-se o tempo de estabilização da cabeça e a latência média (metade do tempo de uma revolução).

Temos: $T_a = 249$ ms.

3.2 - Sinais da Interface

Os sinais da interface foram projetados para serem usados em uma configuração em barra comum e são validados, internamente na unidade, pela linha de seleção (SELECT), exceção feita ao READY que, como o SELECT, é um sinal privativo (não-comum).

Todos os sinais da interface são ativos no nível lógico \emptyset . Esta indicação é dada pela letra b após o nome do sinal.

3.2.1 - SINAIS DE SAÍDA

3.2.1.1 - RAW DATA B - Esta linha fornece os pulsos correspondentes à informação gravada não separada (CLOCK e DADO) . Cada transição de fluxo é lida do disco como um pulso positivo de 50 ns.

3.2.1.2 - INDEX B - O início do pulso negativo de ~ 4 us nesta linha representa o início da trilha.

3.2.1.3 - TRACK $\emptyset\emptyset$ B - Esta linha é ativa (Nível \emptyset) quando a cabeça está sobre a trilha zero. Esta linha é utilizada como referência. Os circuitos do stepper motor são inibidos do movimento para fora quando ela está ativa.

3.2.1.4 - READY B - Um nível \emptyset indica que o disco flexível está em sua rotação ideal. É um sinal privativo.

3.2.1.5 - WRITE ENABLED B - Um nível \emptyset nesta linha indica que o disco flexível não está protegido contra a escrita.

3.2.2 - SINAIS DE ENTRADA

3.2.2.1 - SELECT B - Esta linha permite a comunicação entre a unidade e o controlador, liberando as linhas da interface.

3.2.2.2 - WRITE ENABLE B - Quando esta linha está ativa (\emptyset) é permitido a escrita no disco. Se ela é mantida em 1 então só podemos ler.

3.2.2.3 - WRITE DATA B - Esta linha carrega pulsos negativos que representam a informação a ser gravada. A corrente de gravação reverte o fluxo na descida de cada pulso. Eles devem ter uma largura variando entre 0.2 e 1.5 us.

3.2.2.4 - ABOVE TRACK 43 B - Esta linha é utilizada para controlar a amplitude da corrente de gravação. Ela deve ser 1 quando gravando nas trilhas de \emptyset a 43, e \emptyset quando gravando nas trilhas

44 a 76. ABOVE TRACK 43 deve ser estável 10 us antes de WRITE ENABLE ser ativada.

3.2.2.5 - STEP B - Esta linha é utilizada em combinação com DIRECTION e serve para mover a cabeça de uma trilha para outra. Um pulso negativo de 0.2 us a 2 ms faz com que a cabeça mova de uma trilha na direção indicada por DIRECTION. A máxima frequência é 167 pulsos/segundo (6 ms entre pulsos).

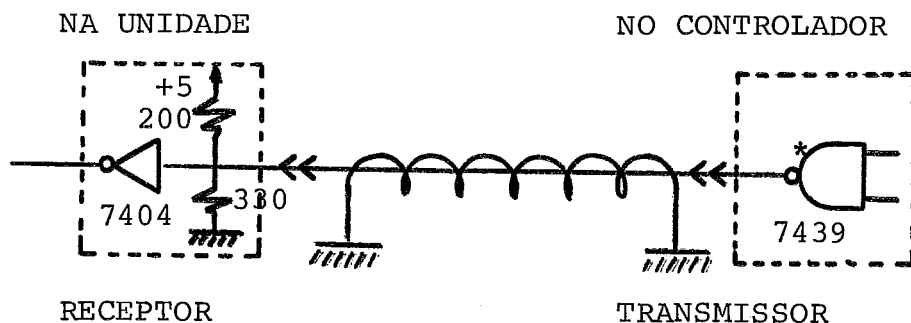
3.2.2.6 - DIRECTION - Esta linha é usada em conjunto com STEP. Quando o nível é alto a direção é PARA DENTRO (números crescentes da trilha). Quando a linha é baixa a direção é PARA FORA. Esta linha deve estar estável 100 ns mínimos antes de serem ativados os pulsos de STEP e permanecer neste estado durante todo o período do pulso.

3.2.2.7 - HEAD LOAD B - Esta linha é utilizada para forçar o disco flexível contra a cabeça. É necessário um atraso de 16 ms após a ativação da linha antes que uma transferência seja possível (leitura ou escrita).

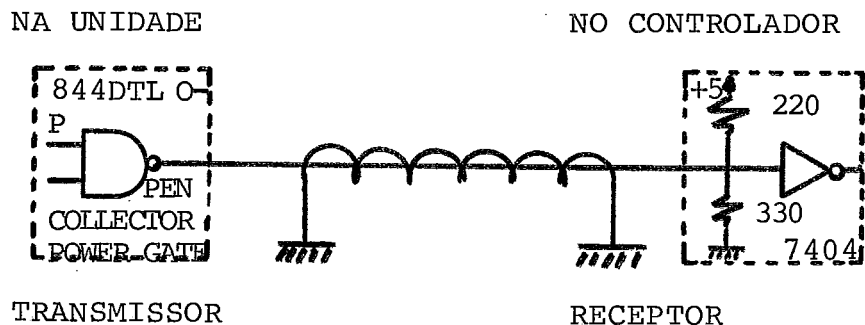
3.2.3 - TERMINAÇÃO DOS SINAIS

Eletricamente, as linhas de entrada e saída devem ser terminadas convenientemente para evitar reflexões e indução de ruídos nos cabos.

SINAIS DE ENTRADA



SINAIS DE SAÍDA



Apresento a seguir um conjunto de três (3) gráficos (FIG. 3; FIG. 4; FIG. 5) que exemplificam a utilização das linhas da interface para leitura, escrita e posicionamento. É importante observar as tolerâncias e restrições particulares impostas a cada sinal.

Para maiores informações recomenda-se a Bibliografia nº

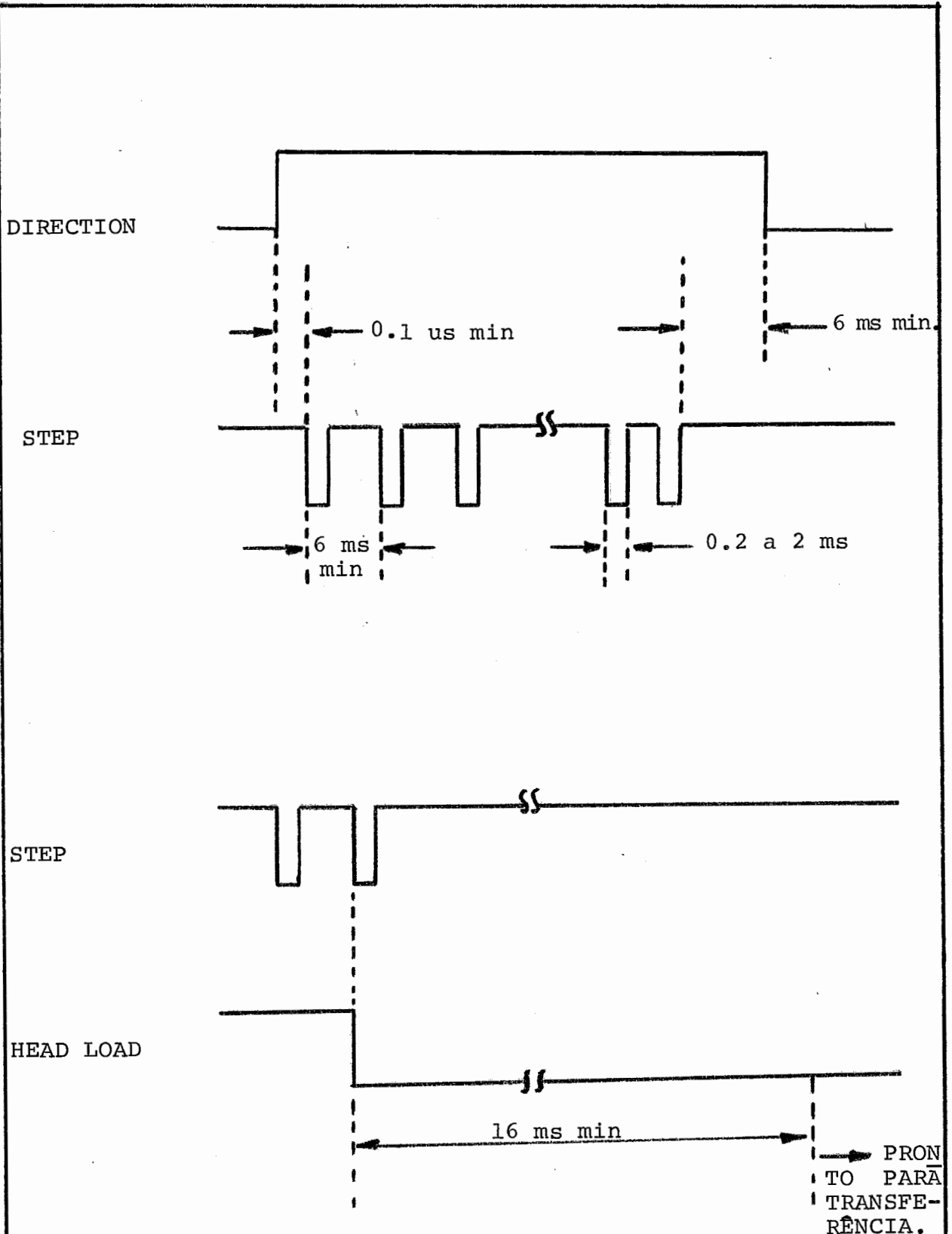


FIG. 3 - DIAGRAMA DE TEMPO POS/TRANSF.

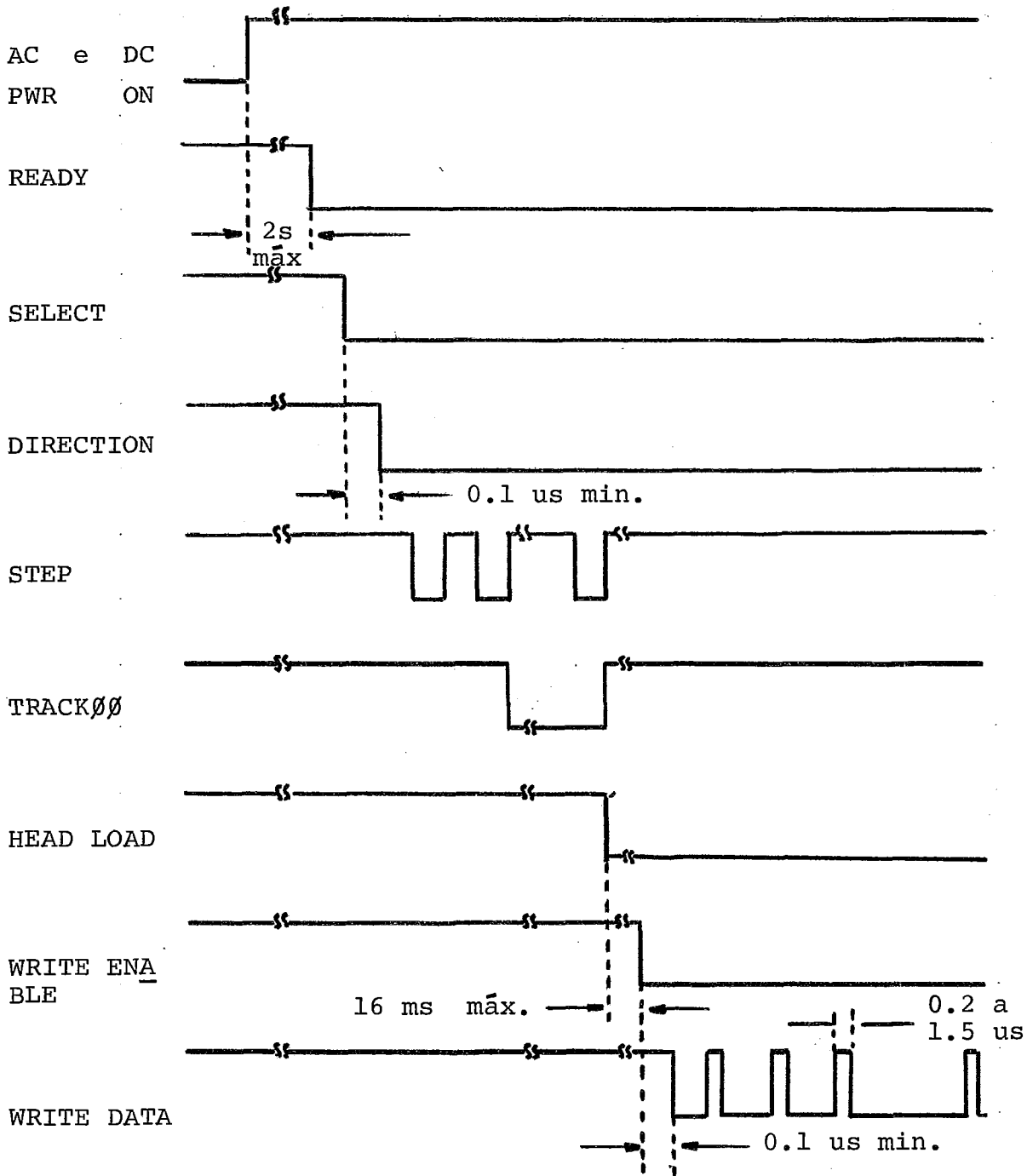


FIG. 4 - DIAGRAMA DE TEMPO PARA ESCRITA

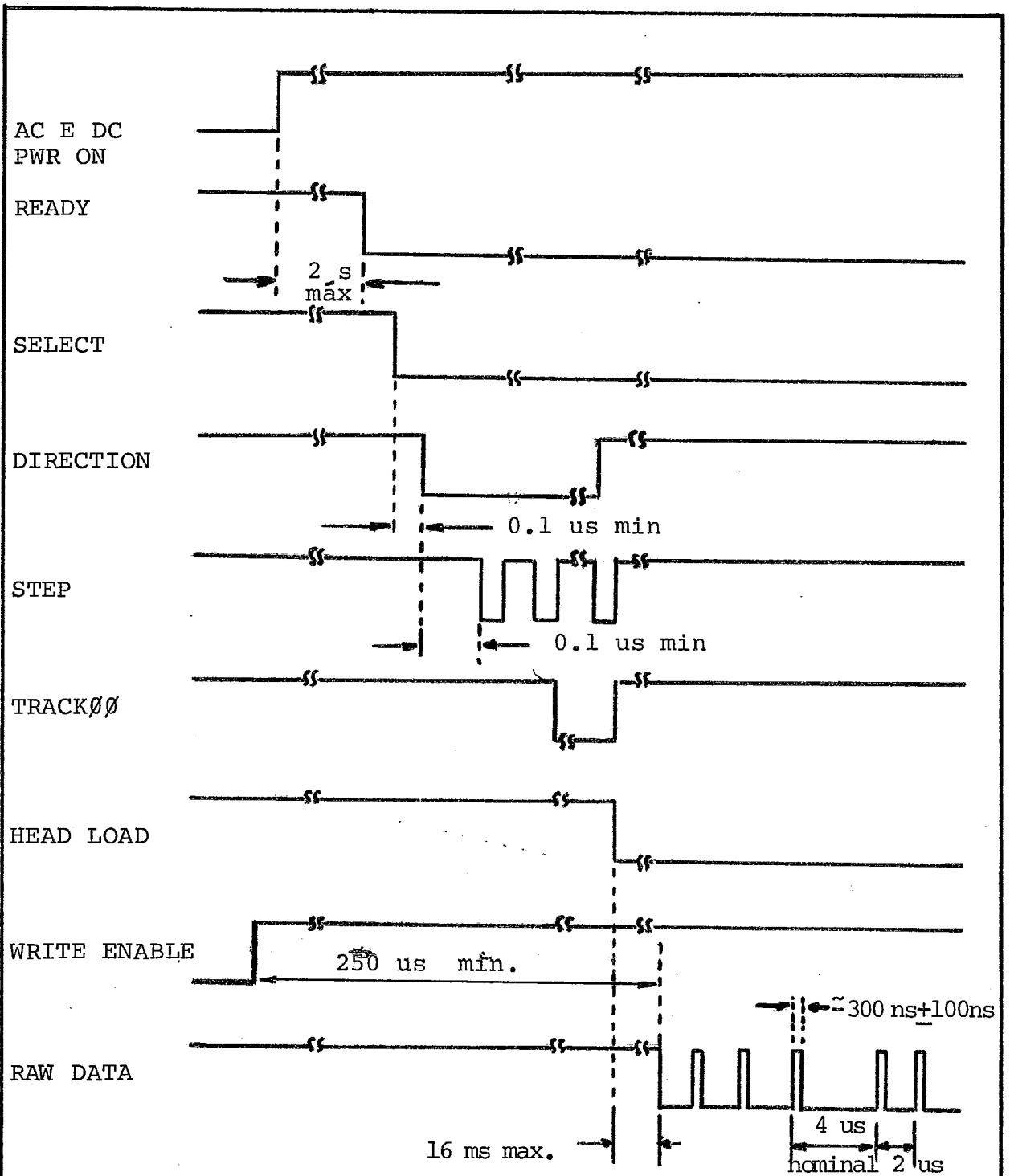


FIG. 5 - DIAGRAMA DE TEMPO PARA LEITURA

4 - ENTRADA/SAÍDA DO TERMINAL INTELIGENTE

O Terminal Inteligente (TI), um microcomputador de aplicação geral, tem como elemento básico principal uma Unidade Central de Processamento (UCP) projetada com um microprocessador-8008 da INTEL. Tendo o repertório de instruções fixo e definido, restou a concepção de uma estrutura de Entrada/Saída (E/S) que fosse versátil e simples o suficiente para o interfaceamento de qualquer tipo de periférico. A versatilidade da estrutura de E/S pode ser medida pela complementação da memória principal, endereçável até 16K bytes, por uma Pilha (STACK), tipo FIFO, de 1 K bytes, dotada de instruções de PUSH e POP, que nada mais são do que instruções de E/S reservadas.

Um diagrama da arquitetura do TI pode ser visto na FIG. 6 . Todos os sinais da barra são ativos em nível \emptyset .

4.1 - Controle e Seleção de Periféricos

O TI possui duas (2) barras de dados gerais que, com referência a UCP, se denominam:

- BARRA DE SAÍDA ou SAIBUS <0:7>
- BARRA DE ENTRADA ou ENTBUS <0:7>

O fluxo de informação entre a UCP e os periféricos se faz por meio destas duas barras e, com exceção da Memória, todos os periféricos colhem os dados do SAIBUS e fornecem dados pelo ENTBUS. A Memória tem uma implementação especial e ela possui vias bidirecionais acessando ambas as barras.

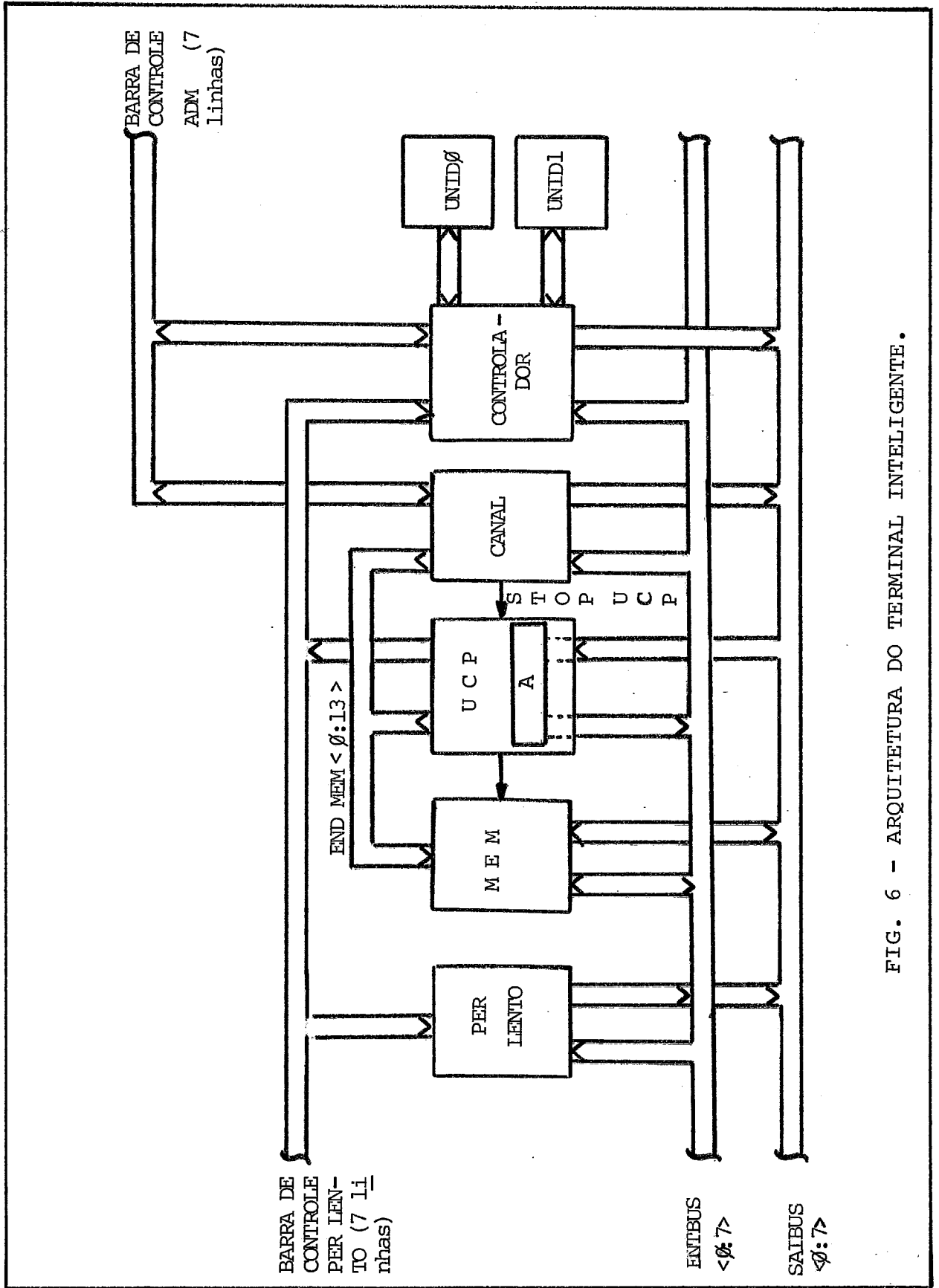


FIG. 6 - ARQUITETURA DO TERMINAL INTELIGENTE.

A barra de controle dos periféricos é composta de sete (7) sinais:

- SELECT (pulso de 4 us)
- CONTROL (pulso de 4 us)
- INPUT (pulso de 4 us)
- RESET INICIAL
- ENDBUS <Ø:2>

O ENDBUS <0:2>, definido pelo próprio código da instrução de E/S, permite decodificar 8 tipos diferentes de CONTROL, SELECT e INPUT. Além do mais, o Acumulador (A), que pode ser previamente carregado com um LOAD ACCUMULATOR IMMEDIATE (LAI), pode ser usado com qualquer uma das possibilidades acima e permite um total de $256 \times 8 \times 2 = 4096$ instruções de Saída e $256 \times 8 = 2048$ instruções de Entrada.

Algumas destas instruções são reservadas para utilização especial e não podem ser usadas. Para se evitar um uso indiscriminado das instruções de E/S, foi estabelecida uma filosofia de E/S que, salvo em casos especiais, é seguida por todos os periféricos do TI, inclusive o Controlador.

Os três (3) tipos de instruções de E/S são definidos da seguinte forma:

a - SELEÇÃO

LAI / xx (0 ≤ /xx ≤ 255)

SELECT

O limite máximo de interfaces é 256. A instrução SELECT provoca um pulso na linha SELECT da barra de controle.

b - INSTRUÇÕES

LAI / xx (0 ≤ xx ≤ 255)

CTL x (0 ≤ x ≤ 7)

São possíveis 2048 instruções diferentes identificadas por um pulso na linha CONTROL.

c - DADOS E ESTADOS

INPUT x (0 ≤ x ≤ 6, x ≠ 3)

Estas instruções são identificadas pelo pulso na linha INPUT e são usadas para a leitura de dados e estados de um periférico pela UCP. INPUT-3 e INPUT-7 são instruções reservadas para STACK.

Periféricos lentos como impressora, leitora e painel que não necessitam de acesso direto à memória (ADM) utilizam-se somente da barra de controle acima e seu protocolo. Para o controle destas, a fim de padronizar a implementação em hardware, foi de-

finida uma interface padrão para qualquer periférico do TI, na qual se acham decodificados: o endereço, os controles e os input. A implementação da interface padrão é vista na FIG.7.

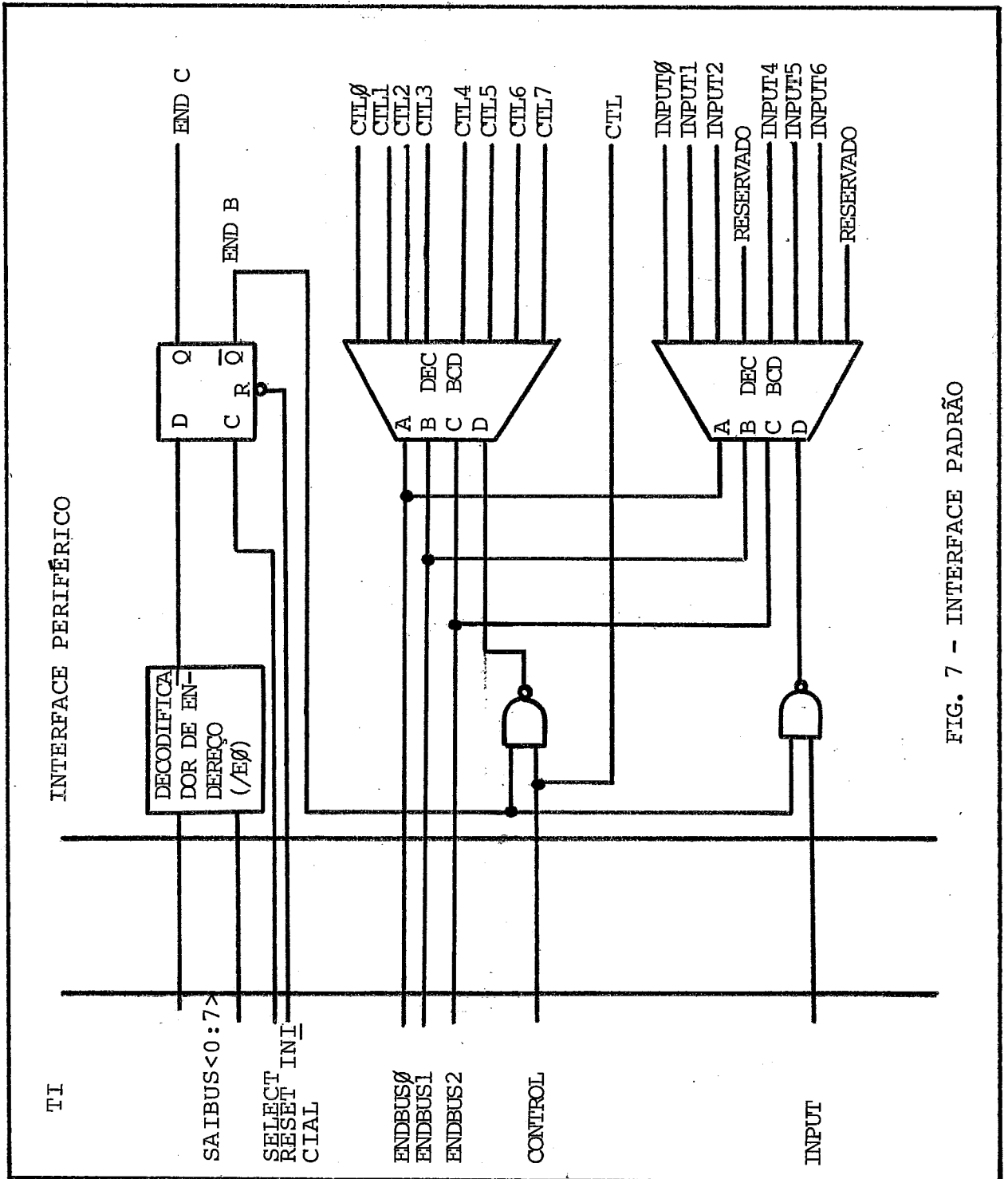


FIG. 7 - INTERFACE PADRÃO

4.2 - Canal

Sabendo que o tempo médio da instrução é da ordem de 20 us, a transferência com periféricos rápidos é impossível na barra descrita acima.

O TI possui um canal que, através de uma barra de controle própria, permite que periféricos acessem diretamente a memória.

Inicialmente o CANAL, como um periférico comum, é armazenado com o TAMANHO DO BLOCO a ser transferido, seu ENDEREÇO INICIAL e o tipo de transferência, se LEITURA ou ESCRITA. Um CTLØ para o periférico CANAL inicia a operação de transferência: a UCP é mantida em HALT pela linha STOP UCP, e o canal executa ciclos sob o controle do periférico selecionado, manipulando automaticamente o endereçamento da memória.

Existe um tempo máximo entre ciclos do Canal fixado em 700 ms, e denominado TIME-OUT. Findo este tempo o Canal sai de operação e libera a UCP. A palavra de estado do Canal possui indicação de time-out, periférico presente e canal selecionado.

O endereço do Canal é /CØ e todo periférico deste possui um endereço compreendido entre /C1 e /FF, inclusive.

Um exemplo de programação do Canal é dado pela sequência a seguir.

LAI /CØ	
SELECT	* SELECIONA CANAL
LAI /xx	
CTL2	* CARREGA PARTE BAIXA DO TAMANHO DE BLOCO E ZERA PARTE ALTA
LAI /xx	
CTL1	* CARREGA PARTE ALTA DO TAMANHO DE BLOCO
LAI /xx	
CTL3	* CARREGA PARTE ALTA DO ENDEREÇO INICIAL
LAI /xx	
CTL4	* CARREGA PARTE BAIXA DO ENDEREÇO INICIAL
CTL5 ou CTL6	* CONFORME A TRANSFERÊNCIA SEJA ESCRITA OU LEITURA (Com relação ao Periférico)
LAI /xx	(/C1 ≤ /xx ≤ /FF)
SELECT	* SELECIONA PERIFÉRICO DE CANAL
⋮	
CTLØ	* INICIA OPERAÇÃO
⋮	

A Barra de Controle para ADM consiste de 6 sinais:

- ABORT CAN
- OP CANAL
- SENT TRANSF
- PED TRANSF
- +1
- DADO PRESENTE

Um pulso negativo em ABORT CAN provoca o fim imediato da transferência em andamento.

A linha OP CANAL fica ativa durante todo o tempo em que a UCP é mantida em HALT.

O SENT TRANSF indica ao periférico se a operação é de leitura ou escrita. Dessa forma a diferença de codificação em um caso e outro pode ficar implícita na preparação do Canal.

O protocolo de leitura é o seguinte:

- a - O periférico ativa PED TRANSF quando o dado já está estável no ENTBUS, iniciando um ciclo do canal;
- b - O Canal inicia o ciclo e responde ao periférico com +1 indicando que o dado foi colhido e o endereço de memória incrementado;
- c - Se uma nova transferência consecutiva não é desejada o PED TRANSF deve ser desativado antes do fim do ciclo do canal.

O protocolo de escrita é o seguinte:

- a - O periférico ativa PED TRANSF, indicando que necessita de um byte, iniciando um ciclo do canal;
- b - O canal fornece DADO PRESENTE, durante o qual o dado está estável no SAIBUS;
- c - O canal fornece +1 indicando que uma nova posição já foi endereçada;

d - Se uma transferência consecutiva nos é desejada, o PED TRANSF deve ser retirado antes do fim do ciclo de Canal.

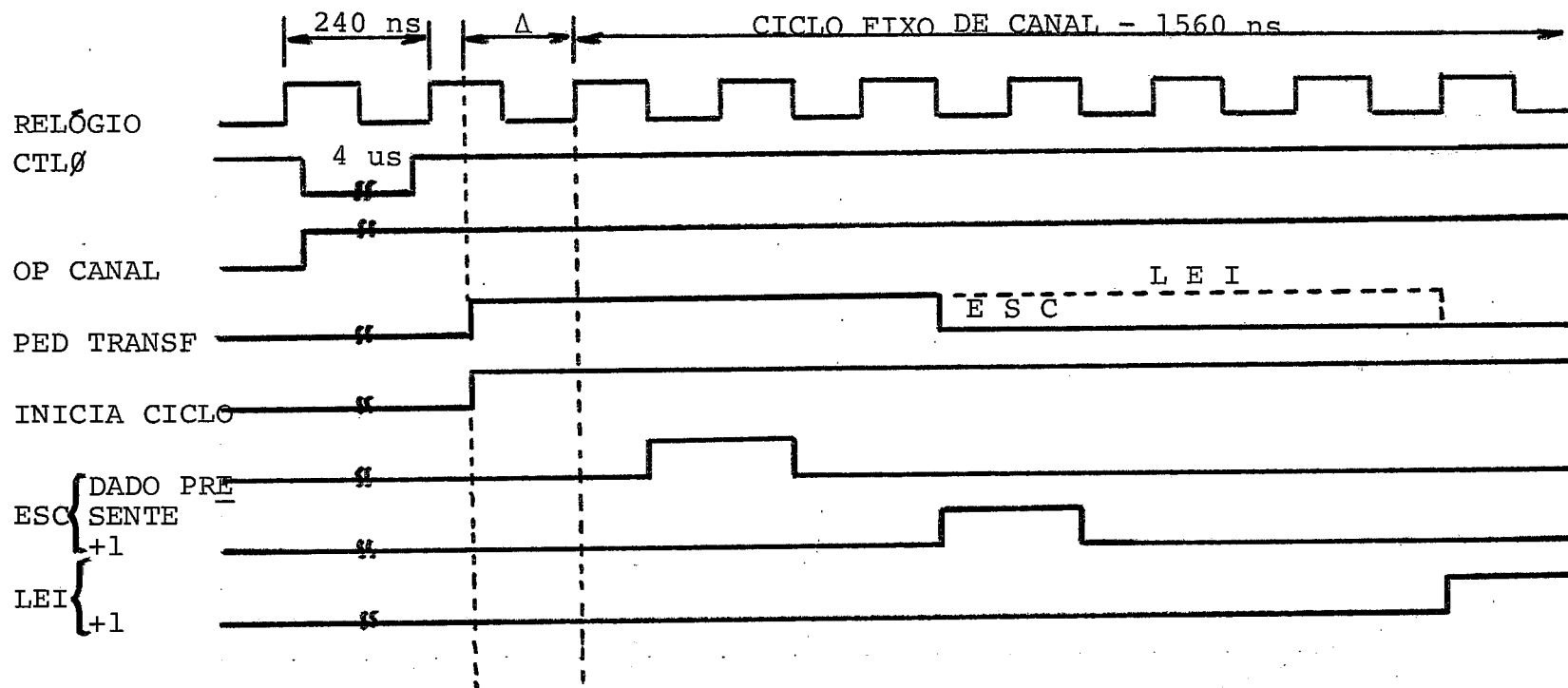
É encontrado na FIG. 8, um Diagrama de Tempo detalhado.

OBS:

1 - O primeiro ciclo do Canal pode ter uma variação Δ sobre o tempo fixo de ciclos (1560 ns) dependendo do instante da ocorrência de PED TRANSF em relação ao relógio interno:

$$0 < \Delta < 240 \text{ ns}$$

2 - Se o PED TRANSF permanecer ativado até o fim de T6, então é encadeado um novo ciclo de 1560 ns. Se o próximo pedido de transferência não é encadeado com o anterior então para este vale a observação 1.



OBS /:

- (1) - O início do ciclo do canal acontece na primeira transição positiva do relógio após PED TRANSF. Desse modo, ciclos não encadeados possuem uma duração $T = 1560 ns$ onde $0 \leq \Delta \leq 240 ns$. Para ciclos encadeados $\Delta=0$ e $T=1560 ns$.
- (2) - O PED TRANSF pode ser desativado a qualquer tempo antes do fim do ciclo de canal, caso não se queira encadear o ciclo seguinte. A forma de onda mostrada é a gerada pelo Controlador na Leitura e Escrita, quando o FF de transferência é zerado pelo +1.

FIG. 8 - DIAGRAMA DE TEMPO DO CANAL

5 - DEFINIÇÃO DO SOFTWARE

Para que o sistema tivesse um bom desempenho procurou-se moldar o hardware a uma estrutura de software pré-definida e que lançasse as diretrizes para a definição da estrutura interna da máquina. Obviamente a implementação final do hardware veio completar a definição do software, determinando a performance final para o Controlador. A compreensão do software é portanto essencial para que se entenda a implementação em hardware e, seguindo este caminho natural, apresentaremos inicialmente uma definição dos estados e instruções disponíveis.

5.1 - Estados

Para que se pudesse ter um controle completo da máquina, do seu desempenho, criou-se 15 bits de estados subdivididos em 3 grupos:

- ESTADOS DO CONTROLADOR;
- ESTADOS DA UNIDADE; e
- ESTADOS RELACIONADOS COM OPERAÇÕES.

Os bits de estado permanecem inalterados desde o fim da execução de uma instrução até o envio da próxima. Isto implica em que a leitura da palavra de estado não altera o seu conteúdo.

5.1.1 - ESTADOS DO CONTROLADOR

5.1.1.1 - PERIFÉRICO PRESENTE - Indica ao TI que o Controlador foi selecionado. Após sua seleção este bit só não estará ligado se as placas não estiverem conectadas ou se a fonte do sistema estiver com uma tensão abaixo da requerida para operação normal.

5.1.1.2 - OCUPADO - Este estado indica que o Controlador está executando uma operação. A ativação ou não deste estado classifica as instruções respectivamente em NÃO IMEDIATAS e IMEDIATAS. Durante o tempo em que o Controlador está ocupado, qualquer instrução enviada é ignorada, com exceção de um RESET. Portanto o funcionamento normal do sistema exige que o estado NÃO-OCUPADO seja confirmado antes do envio da instrução.

5.1.2 - ESTADOS DA UNIDADE

Estes estados são relacionados diretamente com a unidade selecionada e indicam condições a ela pertinentes.

5.1.2.1 - UNIDADE SELECIONADA - Um nível \emptyset indica estar a unidade \emptyset selecionada, e, similarmente, um nível lógico 1 indica a seleção da unidade 1.

5.1.2.2 - UNIDADE NÃO-PRONTA - Um nível 1 indica que a unidade selecionada não está pronta, devido a um dos fatores abaixo:

- A unidade está com a tampa aberta;
- O diskette não está inserido, ou não foi inserido corretamente;
- A velocidade de rotação de regime ainda não foi atingida.

Em qualquer um dos casos acima uma instrução NÃO-IMEDIATA não será executada e o envio desta provocará a sua rejeição. De igual forma, se durante a execução de uma instrução NÃO-IMEDIATA, a unidade apresentar um estado NÃO-PRONTA, a execução é imediatamente interrompida e o estado INSTRUÇÃO REJEITADA ativado. Se a instrução for de posicionamento, o Controlador terminará logicamente a operação embora fisicamente a unidade não esteja posicionada corretamente.

Sempre que uma unidade selecionada passa do estado NÃO-PRONTA para PRONTA ou uma unidade PRONTA é inicialmente selecionada, um posicionamento automático na trilha 0 ocorre levando um tempo constante de aproximadamente 528 ms, durante o qual o Controlador é mantido ocupado.

5.1.2.3 - TRILHA ZERO - Indica que a unidade selecionada está sobre a trilha ZERO.

5.1.2.4 - ESCRITA PROTEGIDA - Indica se o disquete inserido na unidade selecionada está protegido contra a escrita. Esta proteção é controlada por um sensor fotoelétrico existente na unidade o qual deteta a passagem de luz por um pequeno orifício existente no invólucro. Para que se possa escrever é necessário obstruí-lo com um pequeno pedaço de fita adesiva opaca.

5.1.3 - ESTADOS RELACIONADOS

São estados ativados como consequência da execução ou tentativa de execução de uma instrução.

São os a seguir:

5.1.3.1 - INSTRUÇÃO REJEITADA - Quando este estado é ativado, o Controlador é desocupado e o início de execução inibido. É ocasionado pelas condições:

- Tenta-se posicionar uma trilha maior que 76;
- Tenta-se escrever num diskette protegido;
- Tenta-se uma operação NÃO-IMEDIATA em uma unidade NÃO-PRONTA.

5.1.3.2 - ERRO DE DIMENSÃO - Este tipo de erro sempre ocorrerá quando, numa operação de transferência, o tamanho do bloco (TB) especificado no Canal é menor do que o necessário para que o Controlador realize a operação. Neste caso informação será perdida e o Controlador continuará a execução, mesmo que o Canal tenha saído de operação.

5.1.3.3 - ERRO DE CRC NA IDENTIFICAÇÃO DO SETOR (ERI) - Ocorre quando ao final do campo de identificação o cheque do CRC acusou erro. O cheque é feito do primeiro bit de MEID até o último bit do CRC. A ocorrência de erro provoca o fim de operação para uma instrução de escrita.

5.1.3.4 - ERRO DE CRC NA INFORMAÇÃO DO SETOR (ERS) - Ocorre quando ao final da leitura da informação, o cheque do CRC acusar erro. O cheque é feito do primeiro bit de MEIV ou MEIE até o último bit do CRC gravado.

5.1.3.5 - TRILHA INCORRETA (TRI) - Durante a passagem da cabeça por sobre a identificação do setor procurado, o byte referente ao número da trilha é automaticamente checado com o valor da trilha armazenado com a última operação de POS. Caso o cheque falhe, o Controlador interrompe a operação que estiver sendo executada acusando este estado. Isto ocorrerá caso o byte da trilha tenha sido danificado ou a trilha física não corresponda à trilha lógica. A detecção deste estado é inibida pelas instruções LETI e TCI, descritas a frente. No caso da execução de uma instrução LEI, a detecção é válida caso a primeira identificação encontrada coincida com a do setor armazenado.

5.1.3.6 - FALTA MARCA ESPECIAL (FME) - Este estado é ativado caso, ao final de uma rotação completa do diskette, nenhuma marca especial tiver sido detetada, com exceção da MEIX que não é considerada.

A ocorrência deste estado pode indicar uma trilha não formatada ou destruição da mesma, supondo o hardware funcionando corretamente. FME é ativado sempre ao início de qualquer instrução.

5.1.3.7 - SETOR NÃO-ENCONTRADO (SNE) - Indica que o Controlador não conseguiu encontrar nenhuma identificação de setor com o número do setor armazenado após uma revolução completa. SNE é sempre a ativado no início de qualquer instrução.

5.1.3.8 - INFORMAÇÃO ELIMINADA (IEL) - Ocorre quando se lê um setor cuja informação é precedida de uma MEIE.

5.1.3.9 - OPERAÇÃO INCORRETA (OPI) - Ocorre sempre que algum dos estados seguintes ocorrer: ERRO DE DIMENSÃO, ERI, ERS, TRI ou SNE.

5.2 - Instruções

As instruções do Controlador foram criadas visando dar a maior flexibilidade possível ao software, tanto na manipulação normal da informação como na recuperação de erros. Como veremos adiante, a potencialidade do repertório vai além do que se encontra normalmente em Controladores de Disco Flexível.

Podemos classificá-las em quatro grandes grupos:

- IMEDIATAS;
- POSICIONAMENTO;
- LEITURA/ESCRITA; e
- FORMATAÇÃO.

O envio de uma instrução ao Controlador, se ele não estiver ocupado, provoca a desativação de todos os Estados Relacionados com exceção de FME, SNE e OPI que são inicialmente ativados. Qualquer operação do Controlador é cancelada após uma revolução completa a partir do início da execução.

5.2.1 - IMEDIATAS

São na realidade pseudo-instruções caracterizados por sua ação imediata, não ocupando o Controlador. São as a seguir:

5.2.1.1 - RESET Cod: CTL6

É a única instrução não inibida pelo estado OCUPADO e interrompe a execução de qualquer outra, desocupando o Controlador e colocando os Estados Relacionados na condição inicial.

5.2.1.2 - TAMANHO DO SETOR (TAS) Cod: LAI TAS
CTL1

Esta instrução armazena o tamanho do setor a ser utilizado que deverá, invariavelmente, ser igual ao TAS dos setores da trilha usada. Caso esta igualdade não seja verificada, ocorrerão erros imprevisíveis. Necessariamente uma trilha deve ter todos os seus setores do mesmo tamanho, restrição esta que não se estende a todo o diskette porque TAS possibilita esta versatilidade. O tamanho pode variar entre 1 e 256.

5.2.1.3 - SETOR (SET) Cod: LAI SET
CTL2

Armazena a identificação do novo setor a ser considerado nas próximas operações. O maior número de setores

possível é 32, e as identificações podem variar no intervalo $[\emptyset, 31]$.

5.2.1.4 - SELECIONADA (SEL) Cod: LAI b
CTL3

Esta instrução permite a escolha de uma das unidades disponíveis. Sempre que ocorre um selecionamento diferente do anterior, o Controlador fica internamente inibido de executar qualquer instrução de transferência ou posicionamento por 16 ms. Este atraso, transparente ao programador, garante a estabilidade da cabeça.

5.2.2 - POSICIONAMENTO (POS) Cod: LAI TR
CTL7

Esta instrução determina o posicionamento da cabeka sobre a trilha escolhida. Chamando de TR o valor da trilha associado a POS, existem 3 possibilidades:

$$(1) - \emptyset \leq TR \leq 76$$

TR corresponde realmente a uma trilha física e o posicionamento ocorre.

$$(2) - 76 < TR < 128$$

Neste caso a trilha correspondente é inexistente e a instrução não é executada, o

correndo o estado Instrução Rejeitada.

(3) - $128 \leq TR$

Neste caso ocorre um posicionamento força do na trilha ZERO. Esta instrução garante a resincronização do circuito de posicionamento diretamente pelo software caso tenha sido detetado um estado TRI. Independentemente da posição inicial da cabeça, o tempo para este posicionamento é constante e aproximadamente igual a 528 ms.

Após uma instrução de POS a cabeça é automaticamente abaixada. Embora o estado Ocupado seja desativado, a interface só estará internamente liberada para a execução de uma instrução de transferência ou formatação após um atraso de 16 ms necessários ao abaixamento e estabilização da cabeça. Este controle é transparente ao programador e nada o impede de imediatamente após uma instrução POS, enviar, por exemplo, uma instrução de transferência. Neste caso a instrução ficará armazenada aguardando o fim da estabilização, mas mantendo o estado OCUPADO no Controlador.

Se no início da execução de um POS que envolverá o movimento da cabeça, ela está abaixada,

é dado um atraso de 10 ms antes do início da operação propriamente dita, durante o qual ela é levantada.

Considerando-se as seguintes condições em um POS:

A cabeça está inicialmente abaixada - CA
A cabeça está inicialmente levantada - CL
A trilha atual já é a desejada - TD

E sendo:

$K = 6,6$ ms (constante de posicionamento)

$E = 10$ ms (tempo de elevação da cabeça)

$t = 8$ us

N = diferença entre a trilha atual e a desejada em valor absoluto,

podemos obter a tabela abaixo para o tempo total de posicionamento (durante o qual o Controlador fica ocupado) para as diversas possibilidades:

CABEÇA	TRILHA	TEMPO DE POS
CL	TD	t
CA	TD	t
CL	--	$N \times K + t$
CA	--	$N \times K + E + t$

Um posicionamento automático (POS AUT) forçado na trilha ZERO ocorre sempre que uma unidade selecionada passa de um estado NÃO-PRONTA para PRONTA ou uma unidade PRONTA é inicialmente selecionada. O tempo deste posicionamento é constante, aproximadamente 528 ms, ficando o estado OCUPADO ativado durante a operação.

Os registros da trilha são independentes para cada unidade. Desse modo após um posicionamento, o valor da trilha atual permanece inalterado, independentemente de ser a unidade desselecionada ou não, até outro POS ou POS AUT.

5.2.3 - LEITURA/ESCRITA

São ao todo sete:

- LÊ SETOR (LES);
- LÊ IDENTIFICAÇÃO (LEI);
- LÊ TODAS IDENTIFICAÇÕES (LETI);
- TRANSFERE CAMPO DE INFORMAÇÃO (TCI);
- ESCREVE SETOR (ESC);
- ELIMINE INFORMAÇÃO ZERANDO (ELIZ); e
- ELIMINE INFORMAÇÃO (ELI).

Todas as instruções acima, exceção de ELIZ, utilizam o Canal, dispositivo de acesso direto à memória, para as suas transferências. O Canal deve então ser previamente armazenado com o En-

dereço, Tamanho do Bloco (TB) e Sentido da Transferência. As operações de transferência são disparadas por uma pseudo-instrução denominada INIC (CTLØ) que nada mais é do que um pulso de liberação.

As operações de escrita são inibidas e canceladas pela detecção de um erro de CRC na identificação do setor procurado. Elas são rejeitadas caso a unidade esteja protegida contra a escrita, ativando INSTRUÇÃO REJEITADA.

No caso da leitura, a transferência é completada apesar da detecção de ERI.

5.2.3.1 - LES - a informação do setor previamente selecionado é transferida para a memória. Se o tamanho do setor (TAS) é maior que o tamanho do bloco (TB), o Canal sairá de operação antes que todo o setor seja transferido e ocorrerá um Erro de Dimensão (ERD). Se $TAS < TB$ então o Canal não completará sua transferência e sairá por Time Out. Para $TAS = TB$ a operação é normal.

5.2.3.2 - LEI - esta instrução permite que se transfira para a memória os bytes da

trilha e do setor da primeira identificação encontrada. É essencial para algoritmos de otimização de acessos. Durante esta operação SNE não é ativado.

5.2.3.3 - LETI - transfere para a memória os bytes de trilha e setor de todas as identificações de setor existentes na ordem física de gravação a partir do início da trilha. O TB deve ser dimensionado com o dobro do número de setores existentes na trilha. O estado ERI é ativado no primeiro cheque de CRC de identificação que falhe. A detecção de TRI é inibida durante a execução.

5.2.3.4 - TCI - esta instrução transfere o primeiro campo de informação encontrado, independentemente do seu campo de identificação. O cheque do CRC para o campo de identificação continua válido mas a detecção de TRI é inibida. Esta instrução permite a recuperação do conteúdo de um setor cujo campo de identificação tenha sido alterado nos bytes do setor ou da trilha. Durante

esta operação SNE não é ativado.

5.2.3.5 - ESC - grava no setor selecionado tantos bytes quanto for o TAS, antecedidos de uma MEIV. Caso o TB < TAS então o setor é completado com bytes zero e ocorrerá ERD. Inversamente, se TB > TAS então os TAS-ésimos primeiros bytes serão gravados e o Canal sairá por Time-out. Para uma operação normal TAS = TB.

5.2.3.6 - ELIZ - esta instrução zera todo o campo de informação do setor, antecedendo-o por uma MEIE.

5.2.3.7 - ELI - grava no setor tantos bytes quanto for o TAS antecedidos de uma MEIE. Caso TB < TAS então o setor é completado com bytes zero, ocorrendo ERD. No caso oposto, os primeiros TAS-ésimos bytes serão transferidos e o Canal sairá de operação por Time-out. Para uma operação normal: TAS = TB.

5.2.4 - FORMATAÇÃO (FOR)

Formatar uma trilha é gravar uma seqüência de pulsos capazes de definir campos determinados, conformando-a em setores. A instrução de FOR possibilita a formatação da trilha através de um conjunto de microinstruções armazenadas na memória e transferidas pelo Canal durante a execução. Este conjunto de microinstruções forma o Programa de Formatação, cujo endereço inicial e o tamanho são armazenados no Canal antes do envio da pseudo-instrução INIC que libera a execução.

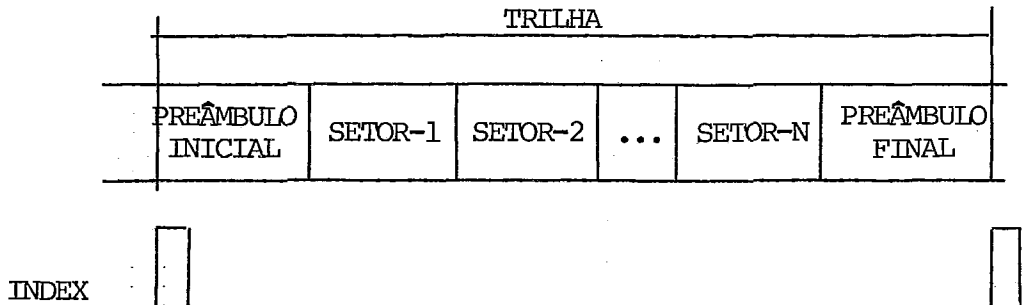
Devido a variações na velocidade de rotação e na frequência de gravação, existem requisitos que devem ser observados na definição de um programa de formatação, com relação aos preâmbulos existentes na trilha.

Recapitulando as estruturas definidas no Capítulo 2, temos:

FORMATO DOS SETORES:

SETOR				
CAMPO DE IDENTIFICAÇÃO	PREÂM-BULO-1	CAMPO DE INFORMAÇÃO	PREÂM-BULO-2	
7 bytes		TAS+3 bytes		

FORMATO DA TRILHA:



5.2.4.1 - RESTRIÇÃO AOS PREÂMBULOS

Sabendo que $\Delta v \underline{\Delta}$ variação devida a rotação e frequência de gravação $\cong \pm 2.5\%$

podemos fazer as seguintes considerações:

- (1) - PREÂMBULO FINAL MÍNIMO (PFM)

Chamando

Bmin = tempo de byte mínimo

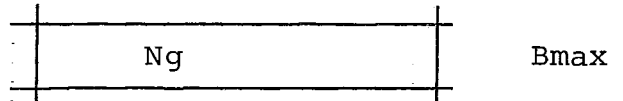
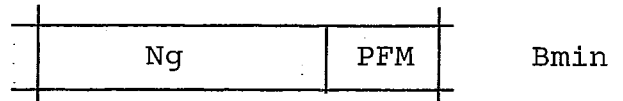
Bmax = tempo de byte máximo

Temos:

$$Bmin = 32 - 32 \times 0.025 = 31.2 \text{ us}$$

$$Bmax = 32 + 32 \times 0.025 = 32.8 \text{ us}$$

Sendo Ng o número de bytes úteis gravados, podemos estabelecer a relação a seguir:



$$(Ng + PFM) Bmin = Ng Bmax$$

$$\therefore PFM = \frac{Ng (Bmax - Bmin)}{Bmin}$$

Como $Ng = \frac{166.67 \text{ ms}}{Bmax}$ temos:

$$PFM = \frac{166.67 \text{ ms} (Bmax - Bmin)}{Bmax \cdot Bmin}$$

$$\therefore PFM = 261 \text{ bytes}$$

Adotaremos $\boxed{PFM = 265 \text{ bytes}}$;

e $\boxed{Ng = 5076 \text{ bytes}}$

A restrição acima é válida independente do número de setores da trilha.

(2) - PREÂMBULO INICIAL MÍNIMO (PIM)

O PIM depende apenas do número de bytes necessários para a sincronização e detecção de uma marca especial e que nominalmente é de 3 bytes. Adotaremos:

$$\boxed{\text{PIM} = 10 \text{ bytes}}$$

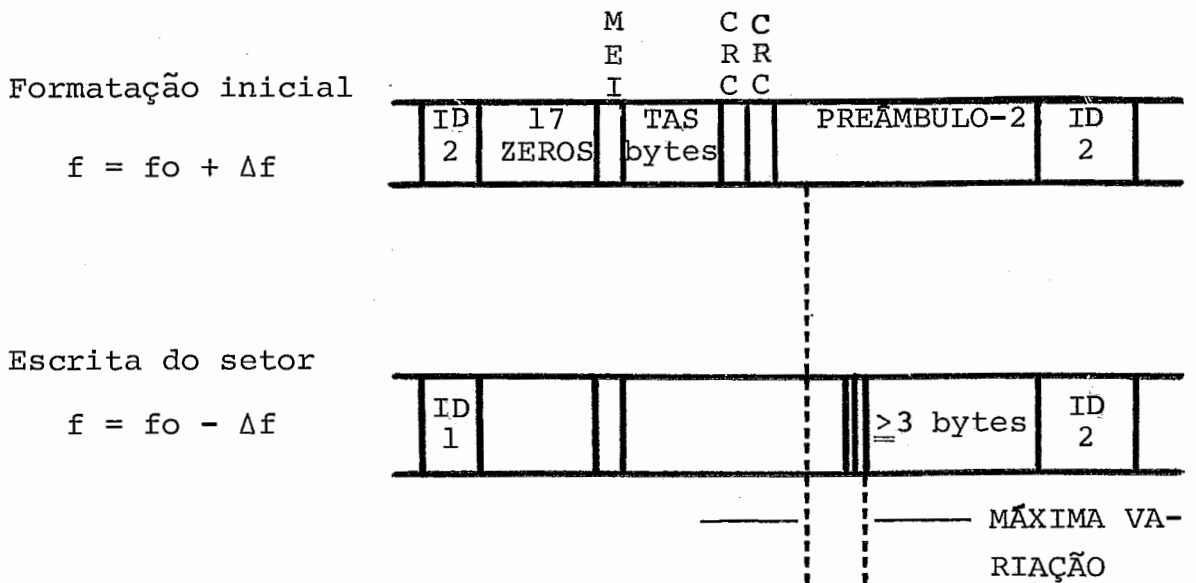
(3) - PREÂMBULO-1

Tem o tamanho fixo de 17 bytes ZERO.

(4) - PREÂMBULO-2

Deve levar em conta a variação do campo de informação em escritos posteriores à formatação e os 3 bytes mínimos necessários à detecção de uma marca especial. Observando que a escrita é iniciada logo após o último byte de CRC do campo de identificação e que estes permanecem inalterados até a próxima formatação, o pior caso ocorre quando a formatação da trilha é feita na frequência máxima, re

sultando na menor distância possível entre campos de identificação. Se o setor é escrito com a frequência mínima, então para o mesmo número de bytes um maior espaço de gravação é ocupado. Veja o esquema abaixo:



$$\text{PREÂMBULO-2} \geq 3 + (17+1+\text{TAS}+2) \cdot 0,05$$

$$\text{PREÂMBULO-2} \geq 4 + 0.05 \text{ TAS}$$

Para não trabalharmos nos limites, adotaremos como norma:

$\text{PREÂMBULO-2} \geq \lfloor 10+0.05 \text{ TAS} \rfloor$

5.2.4.2 - RELAÇÃO ENTRE O NÚMERO DE SETORES E O TAMANHO DO SETOR PARA APROVEITAMENTO MÁXIMO

Sendo:

N_g Δ número de bytes da trilha-pFM = 5076 bytes,

N_S = número de setores,

TAS = tamanho do setor,

Preâmbulo Inicial = 10 bytes,

Preâmbulo Final = 265 bytes, e

B = número de bytes do setor + preâmbulo-2 = 37 + 1.05 TAS

Então:

$$N_g = PI + N_S \times B - 9 \text{ bytes}$$

OBS: - A parcela 9 bytes acima, se refere aos bytes do Preâmbulo-2 do último setor que seriam necessários a detecção de uma próxima marca especial.

Substituindo pelos valores numéricos temos:

$$5076 = 10 + N_S(37+1.05 \text{ TAS}) - 9$$

$$\therefore 5076 = 37 N_S + 1.05 \text{ TAS} \cdot N_S$$

$$\therefore \boxed{\text{TAS} = \frac{5075 - 37 N_S}{1.05 N_S}}$$

A relação anterior fornece o máximo TAS para um número de setores determinado. Considerando que o número máximo de setores é 32 e que o tamanho máximo do setor é 256 bytes, deduzimos da relação obtida a tabela abaixo:

NÚMERO DE SETORES	TASmáximo	APROVEITAMENTO (%)
32	115	71
31	120	71
30	125	72
29	131	73
28	137	74
27	143	74
26	150	75
25	158	76
24	166	76
23	174	77
22	184	78
21	194	78
20	206	79
19	219	80
18	233	81
17	249	81
16	256	79

OBS :

$$\text{APROVEITAMENTO} = \frac{\text{NS} \times \text{TASmax} \times 100\%}{5208 \text{ (número de bytes nominal/trilha)}}$$

5208 (número de bytes nominal/trilha)

Para 16 setores a limitação no tamanho do setor provoca a queda no aproveitamento. O maior aproveitamento teórico ocorre para um número de setores igual a 17.

5.2.4.3 - MICROINSTRUÇÕES

Um programa de Formatação é um conjunto de bytes armazenados na memória e que são transferidos para o Controlador durante uma operação de FOR, cada byte representando uma microinstrução ou um dado. O programa atua em tempo real, ou seja, a cada transferência de uma microinstrução corresponde a gravação de um ou mais bytes. Ao final da gravação dos bytes correspondentes a uma microinstrução, uma nova é buscada na memória pelo CANAL.

As microinstruções disponíveis para a formatação são:

(OBS): - "b" indica 0 ou 1. B é o valor decimal correspondente aos b's.

(1) - LPZ B+1 | 1 0 b b b b b b |

Grava B+1 bytes ZERO.

(2) - WDD (C0) | 1 1 0 0 0 0 0 0 |

Grava uma MEIE.

(3) - WDAM (C3) | 1 1 0 0 0 0 1 1 |

Grava uma MEIV.

(4) - WEX (E4) | 1 1 1 0 0 1 0 0 |

Grava uma MEIX.

(5) - WID (C6) | 1 1 0 0 0 1 1 0 |

Grava uma MEID.

(6) - WTR (02) | 0 0 0 0 0 0 1 0 |

Grava um byte com o número da trilha sendo formatada.

(7) - BYTE (01) | 0 0 0 0 0 0 0 1 |

Grava o byte que o segue na memória.

(8) - CRC (08) | 0 0 0 0 1 0 0 0 |

Grava um byte do CRC correspondente a última marca especial gravada e aos bytes gravados subsequentemente. Duas micro CRC consecutivas gravam os dois bytes do CRC.

(9) - LPZS (30) | 0 0 1 1 0 0 0 0 |

Grava um número de bytes ZERO igual ao tamanho do setor definido pelo TAS.

(10) - FIM (04) | 0 0 0 0 0 1 0 0 |

Indica o fim de uma operação de FOR. Normalmente não é utilizada porque a formatação é feita dimensionando o preâmbulo final com um número de bytes maior do que o calculado. Neste caso a própria detecção do fim de trilha provoca o fim de execução da instrução FOR sem a necessidade do FIM. A utilização da microinstrução FIM só deve ser feita para casos especiais quando não se quer formatar toda a trilha.

(11) - DADOS (11) | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

Grava os TAS-ésimos bytes subseqüentes a ele.

d - PROGRAMAS DE FORMATAÇÃO

Em alguns casos, como no sistema IBM, os setores são inicializados com um determinado conteúdo. Se tomarmos um disco virgem, além da formatação das trilhas, é necessário inicializar cada setor com sua informação respectiva para que este diskette seja a ceito pelo sistema IBM. A microinstrução DADOS permite que inicializemos um setor com qualquer informação. Entretanto, este conjunto de informações para cada setor individualizará o programa de formatação em partes distintas, diferenciados entre si por um número muito grande de bytes. A criação de um programa que possa gerar o programa de formatação passa a ser uma tarefa difícil e dispendiosa, em termos de memória. O melhor é formatar inicialmente todos os setores com um conteúdo de ZEROS, o que se faz com uma única instrução LPZS. Posteriormente pode-se inicializar cada setor com sua informação conveniente através de um processo simples de cópia de um diskette já inicializado. Desse modo, os blocos correspondentes à inicialização de um setor só diferirão pelo byte do setor, o que tornará fácil a geração do programa de Formatação. Consulte o Apêndice.

Vou dar a seguir dois (2) exemplos de FORMATAÇÃO.

(1) - FORMATAÇÃO IBM (Sem Inicialização)

HEXA	MICROINSTRUÇÃO	DESCRIÇÃO	Nº DE BYTES GRAVADOS
AD	LPZ 46	PREÂMBULO	46
E4	WEX		1
9F	LPZ 32	PREÂMBULO	32
C6	WID	ID SETOR 1	1
02	WTR	ID SETOR 1	1
80	LPZ 1	ID SETOR 1	1
01	BYTE	ID SETOR 1	
HH	Nº DO SETOR	ID SETOR 1	1
80	LPZ 1	ID SETOR 1	1
08	CRC	ID SETOR 1	1
08	CRC	ID SETOR 1	1
90	LPZ 17	PREÂMBULO	17
C3	WDAM		1
30	LPZS	INFOR SETOR 1	128
08	CRC		1
08	CRC		1
A0	LPZ 33	Este trecho do programa é repetido do 25 vezes, trocando-se apenas o valor do setor a cada repetição.	25 x 33
C6	WID		25 x 1
02	WTR		25 x 1
80	LPZ 1		25 x 1
01	BYTE		
HH	Nº DO SETOR		25 x 1
80	LPZ 1		25 x 1
08	CRC		25 x 1
08	CRC		25 x 1
90	LPZ 17		25 x 17
C3	WDAM		25 x 1
30	LPZS		25 x 128
08	CRC		25 x 1
08	CRC		25 x 1
30	LPZS	PREÂMBULO FINAL	128

30	LPZS	PREÂMBULO FINAL	128	
30	LPZS	PREÂMBULO FINAL	128	
			5318	T O T A L

NÚMERO DE BYTES DO PROGRAMA = 369 bytes

(2) - 17 SETORES

Preâmbulo inicial: 10 bytes ZERO
 Preâmbulo-1: 17 bytes ZERO
 Preâmbulo-2: $\lfloor 10 + 0.05TAS \rfloor = \lfloor 22.45 \rfloor = 22$ bytes ZERO
 TAS: 249 bytes
 Preâmbulo Final Mínimo: 265 bytes ZERO

HEXA	MICRO	DESCRIÇÃO	Nº DE BYTES GRAVADOS
89	LPZ1Ø	PREÂMBULO	1Ø
C6	WID	ID1	1
Ø2	WTR	ID1	1
8Ø	LPZ1	ID1	1
Ø1	BYTE	ID1	
HH	Nº DO SETOR		1
8Ø	LPZ1	ID1	1
Ø8	CRC	ID1	1
Ø8	CRC	ID1	1
9Ø	LPZ17		17
C3	WDAM		1
3Ø	LPZS	INFO1	249
Ø8	CRC		1
Ø8	CRC		1

95	LPZ22	Este trecho do programa é repetido 16 vezes alterando-se apenas o valor do setor a cada repetição.	16 x 22
C6	WID		16 x 1
02	WTR		16 x 1
80	LPZ1		16 x 1
01	BYTE		
HH	Nº DO SETOR		16 x 1
80	LPZ1		16 x 1
08	CRC		16 x 1
08	CRC		16 x 1
90	LPZ17		16 x 17
C3	WDAM		16 x 1
30	LPZS		16 x 249
08	CRC		16 x 1
08	CRC		16 x 1
30	LPZS	PREÂMBULO	249
30	LPZS	PREÂMBULO	249
			5552
			T O T A L

TOTAL DE BYTES DO PROGRAMA = 240 bytes

6 - IMPLEMENTAÇÃO DO HARDWARE

A organização da lógica é influenciada diretamente pela existência de dois macroestados: LEITURA e ESCRITA. Esta dualidade pode ser verificada imediatamente pela existência, no fluxo geral (FIG. 9), de dois blocos separados de controle para cada um dos dois estados. Embora estas duas lógicas atuem numa mesma estrutura de fluxo, existe uma diferença marcante quanto à filosofia de projeto de cada uma: enquanto o controle de leitura é totalmente convencional, isto significando estar a lógica dispersa num conjunto amplo de elementos lógicos distribuídos, o controle de escrita é totalmente microprogramado, as instruções de escrita sendo decompostas em um conjunto de microinstruções que executam passos específicos. Esta organização da escrita veio possibilitar uma liberdade completa na formatação da trilha, pois sendo a instrução de formatação na realidade uma cadeia seqüencial de microinstruções, a simples alteração de uma delas permite um novo formato. A leitura se caracteriza pela detecção de estados a partir da informação lida no disco (RAW DATA) e portanto de atividade passiva, ao contrário da escrita que gera os seus próprios microestados e os controla.

Com exceção das instruções de posicionamento, que é assíncrona, da de Formatação, que exige sincronismo apenas com o início da trilha, e dos imediatos, que são pseudoinstruções, as demais exigem sincronismo da lógica de controle com os setores da trilha. Esta sincronização é efetuada por um conjun

to de circuitos que, estando o Controlador ocupado e em leitura, detetam as Marcas Especiais existentes na trilha as sociando-as a flip-flops e fornecendo sincronismo para os bytes subseqüentes. Desta forma, por exemplo, uma vez dete tada uma Marca de Identificação é possível verificar o by te de trilha para um posicionamento correto e determinar se estou no setor que procuro ou não.

Para ser detetada, uma marca especial deve ser precedida de um pre âmbulo de zeros, razão esta que qualquer transição Leit/Esc, e vice-versa que provocam descontinuidades no fluxo gravado devem ocorrer no início destes preâmbulos. Assim quando de sejamos gravar um setor, identificamo-lo primeiramente e, lo go após o último byte do CRC, gravamos os 17 zeros, a Mar ca de Informação (se for o caso) e a informação, retornan- do imediatamente ao estado de leitura após o último byte do CRC. Este procedimento garante a integridade do campo de informação do setor gravado e a deteção da identificação se- guinte. Uma consequência do visto acima é que qualquer ope ração se inicia e termina sob o estado de leitura.

Qualquer instrução é possível de ser executada durante uma revolução completa. Caso isto não ocorra é forçado o térmi no da operação e os bits de estado indicarão as anormalida- des.

Para que não haja um desgaste desnecessário do diskette, exis te no Controlador um contador de revoluções que, quando dete tada 2 revoluções completas sem que tenha havido uma trans-

ferência, provoca automaticamente o levantamento da cabeça. Este contador é zerado e a cabeça abaixada (se não o estiver) no início de qualquer instrução de transferência ou no fim de posicionamento (automático ou não). Este controle evita o tempo de estabilização (16 ms) para operações de transferências sucessivas. Este mesmo circuito fornece o sinal para o término forçado descrito no parágrafo anterior. A cabeça abaixada e estabilizada caracteriza o estado Estável.

Se o Controlador não está OCUPADO, ele está num estado de espera em que são mantidos os estados após a última instrução.

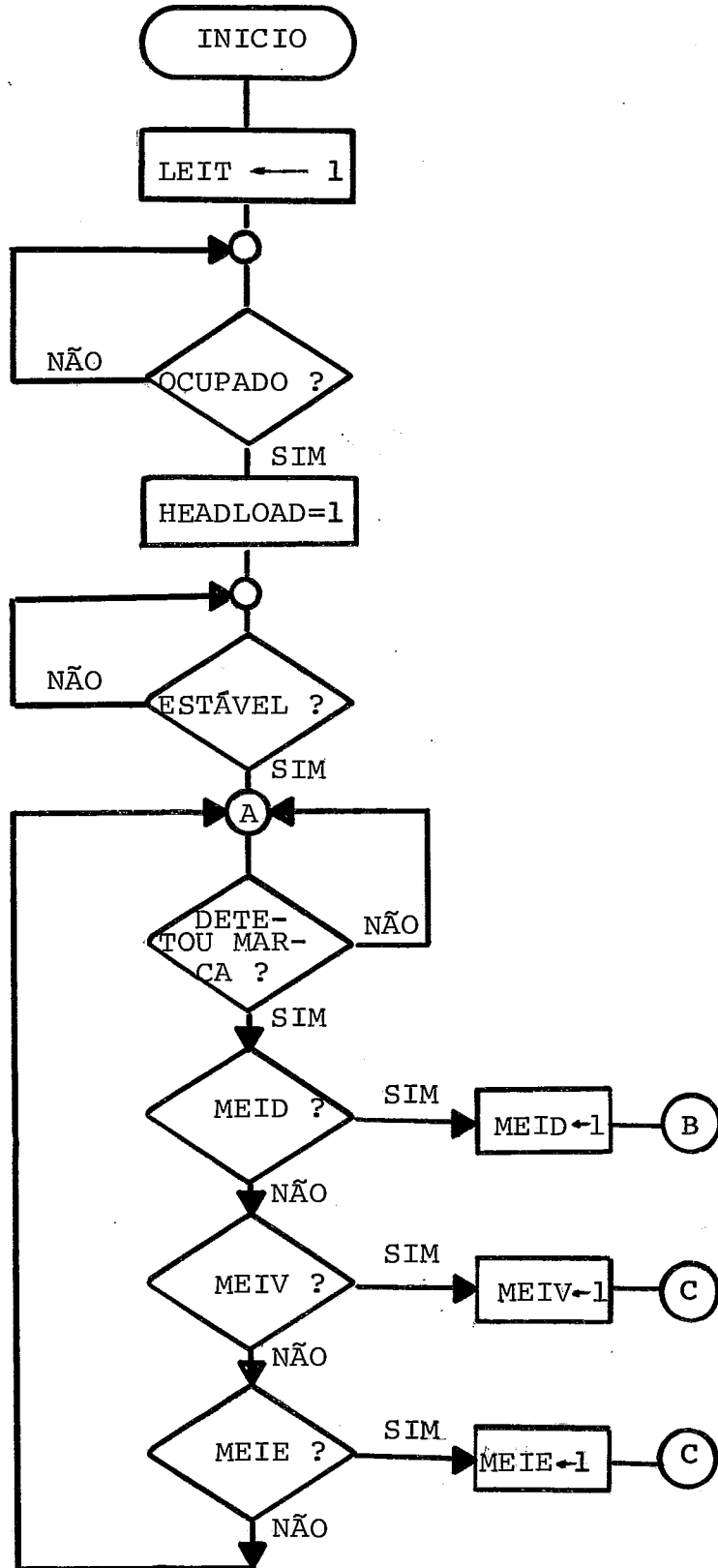
Podemos, então, organizar um fluxograma da atuação do hardware conjuntamente para as várias instruções. Algumas das variáveis utilizadas correspondem a sinais físicos detetados pelos circuitos lógicos e são descritas abaixo:

- LEIT - flip-flop que controla os macroestados Leitura e Escrita;
- MEID - flip-flop da Marca de Identificação;
- MEIV - flip-flop da Marca de Informação Válida;
- MEIE - flip-flop da Marca de Informação Eliminada;
- TRI - estado: TRILHA INCORRETA;
- SETOR ATUAL - indica se é o setor procurado;
- ERI - estado: ERRO DE CRC NA IDENTIFICAÇÃO;
- ERS - estado: ERRO DE CRC NA INFORMAÇÃO DO SETOR

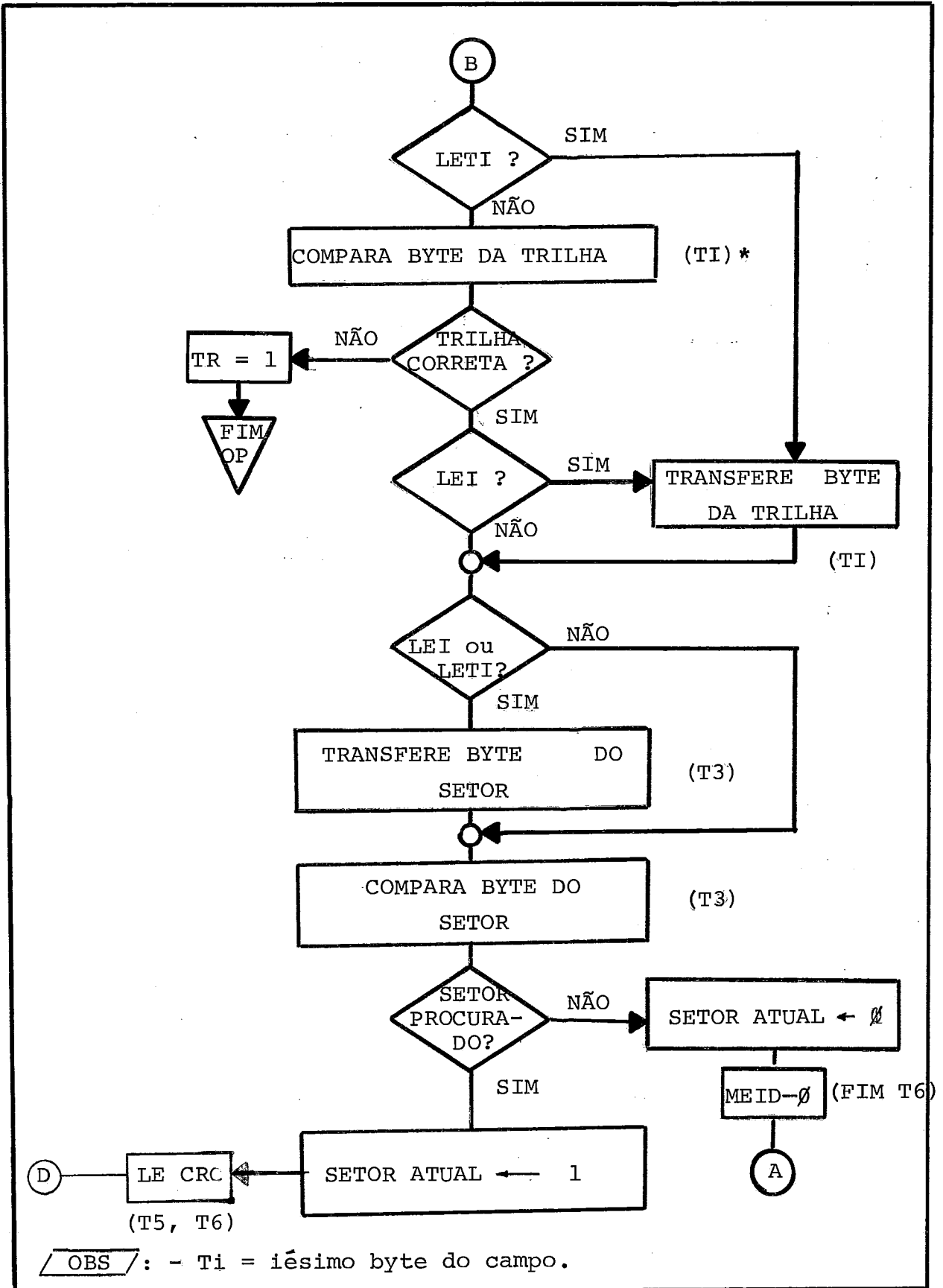
IEL - estado: INFORMAÇÃO ELIMINADA;

HEAD LOAD - quando \emptyset indica CABEÇA LEVANTADA;

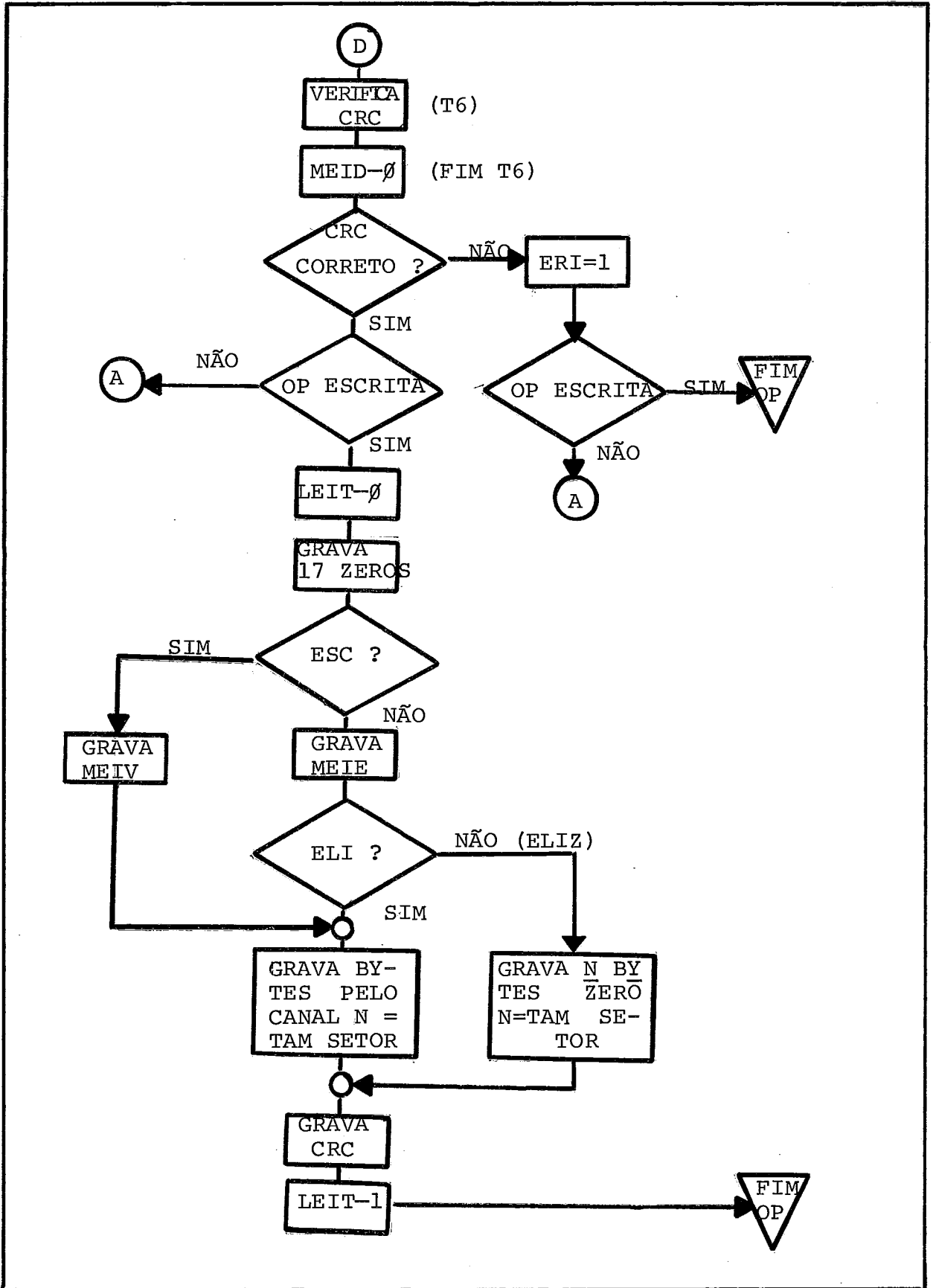
Foram considerados os aspectos principais na execução de cada instrução. As instruções de FOR e POS não foram consideradas. A decodificação da instrução LETI só é liberada após detetado o início da trilha e representamos apenas uma das transferências dentre as várias que podem ocorrer até que seja finalizada a execução com um término forçado.

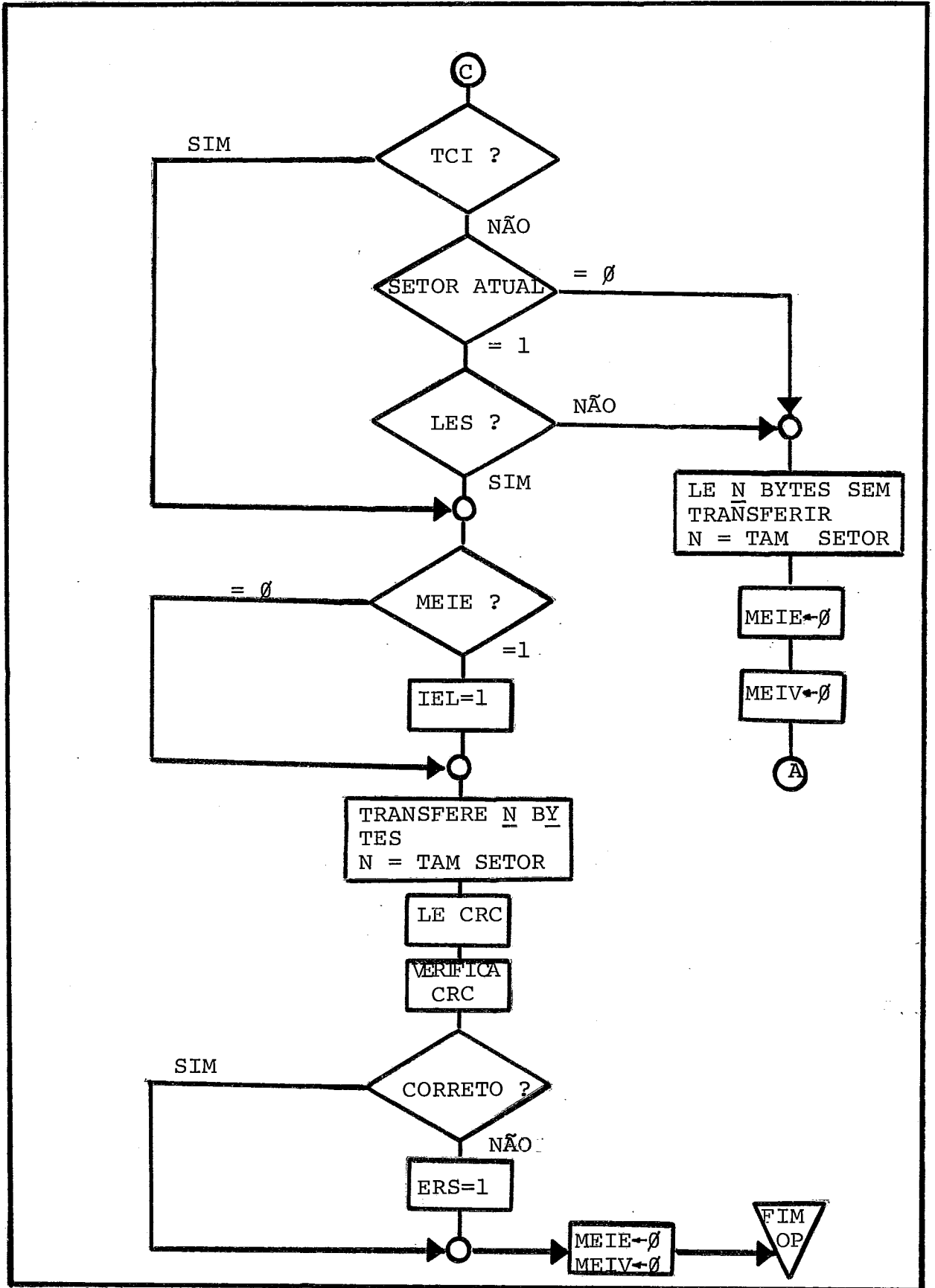


/ OBS / : - Se a marca detetada não corresponder a nenhuma das três, tento novo sin-
cronismo retornando a A. Pode ser o caso da MARCA DE INDEX.



OBS: - T_i = i ésimo byte do campo.





Descreveremos a seguir, detalhadamente, cada um dos blocos integrantes do fluxo geral da FIG. 9. Na descrição dos circuitos o número entre parênteses após o sinal indica a página onde o mesmo é gerado. Se a indicação for U ou T ele é gerado pela unidade CALCOMP ou pelo Terminal, respectivamente. Se não houver indicação o sinal é gerado dentro do próprio subsistema ou o nome é auto-explicativo.

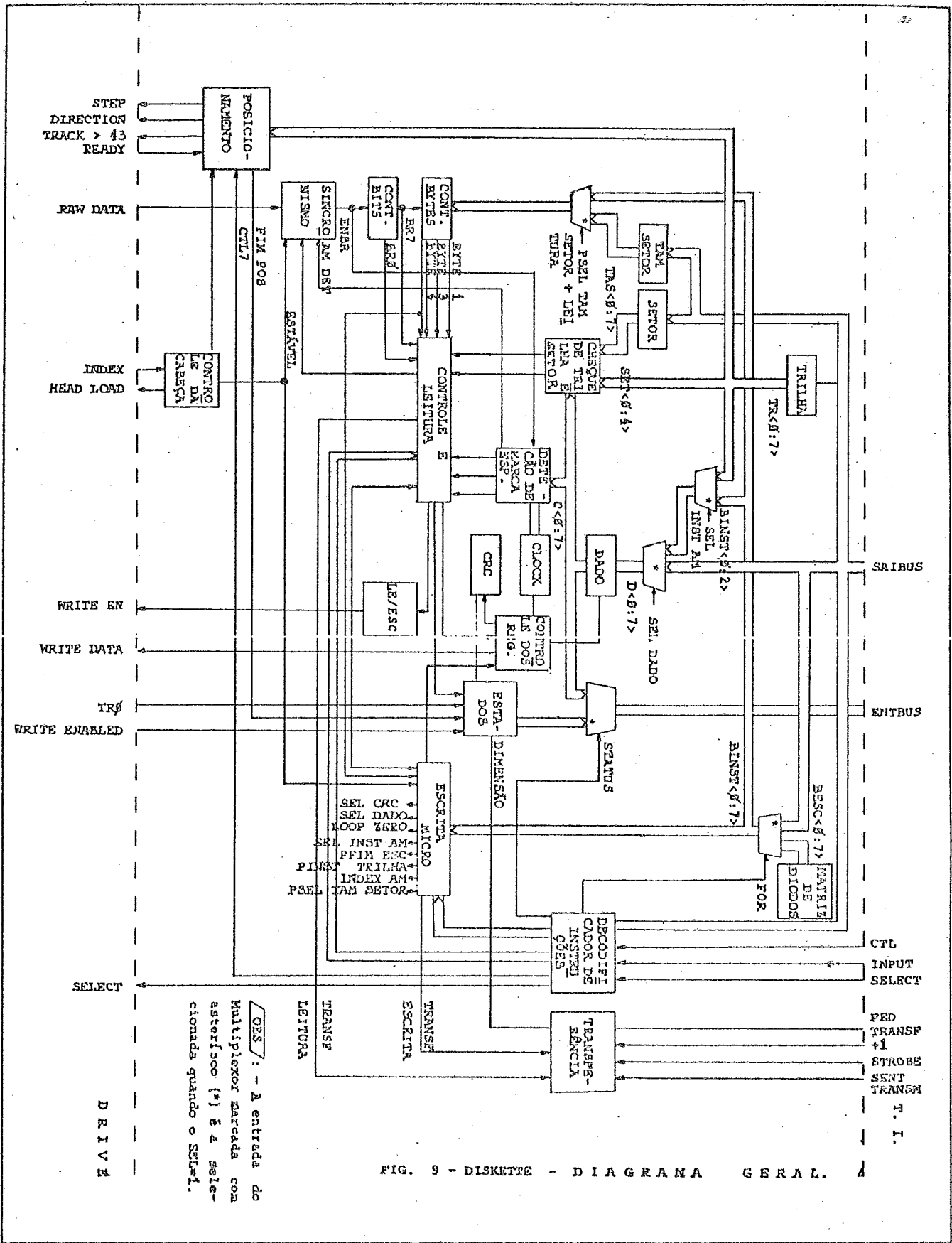


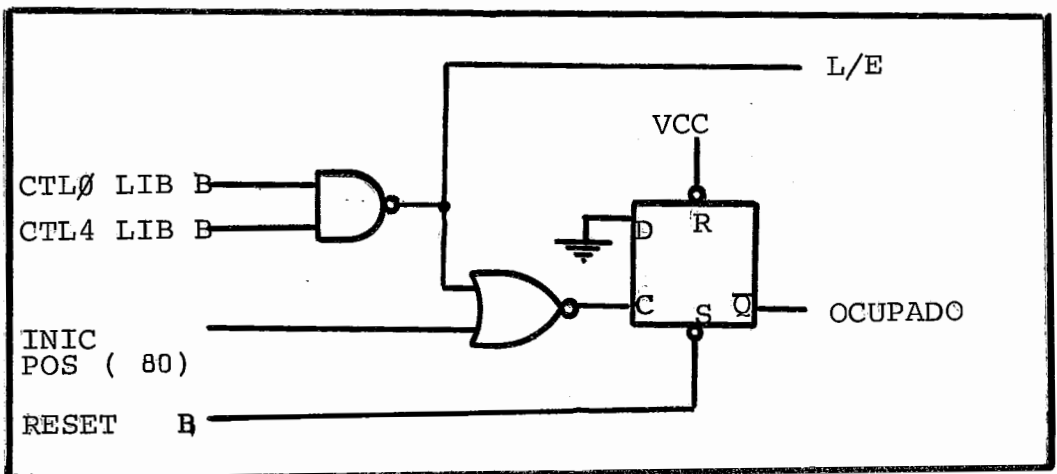
FIG. 9 - DISKETE - DIAGRAMA GERAL.

6.1 - Decodificador de Instruções

A parte de decodificação das instruções compreende a interface padrão do TI e mais uma decodificação suplementar, associando uma determinada linha a cada instrução, além do controle de Ocupado, Instrução Rejeitada e do Reset.

Inicialmente todas as linhas CTLX provenientes da Interface Padrão são validadas pela linha OCUPADO B, com exceção do CTL6, para evitar que, estando o Controlador executando uma instrução, a decodificação de outra possa levar a uma operação incorreta. A esta linha validade chamamos CTLX LIB.

A detecção do estado OCUPADO, ocorre sempre com a execução de um posicionamento ou operação de Leitura/Escreita. As operações de posicionamento são caracterizadas pela ocorrência de INIC POS e as operações de L/E pela ocorrência de CTL0 LIB e CTL4 LIB. A lógica de detecção é mostrada abaixo:



A saída do estado OCUPADO só ocorre com o pulso de RESET, Convém ressaltar que a detecção deste estado só ocorre ao fim dos pulsos, o que dará tempo, como veremos adiante, de serem detetadas condições que rejeitam instruções.

A decodificação das instruções propriamente dita pode ser vista na FIG. 10. A partir da definição do software o esquema lógico é auto-explicativo. A geração de algumas linhas é detalhada abaixo, procurando mostrar a sua atuação funcional:

TRANSF SETOR

Indica uma operação de transferência do campo de informação do setor atual, ocasionada por LES ou TCI.

SETA SETOR ATUAL

A decodificação das instruções TCI e LEI forçam o FF de SETOR ATUAL em 1. O SETOR ATUAL é ativado quando nos encontramos sobre o setor procurado.

L/E

É um pulso utilizado para zerar o Controle da Cabeça e Contador de Index no início de qualquer operação de Leitura ou Escrita.

LEITURA

Ativa durante qualquer operação de leitura, decodifica

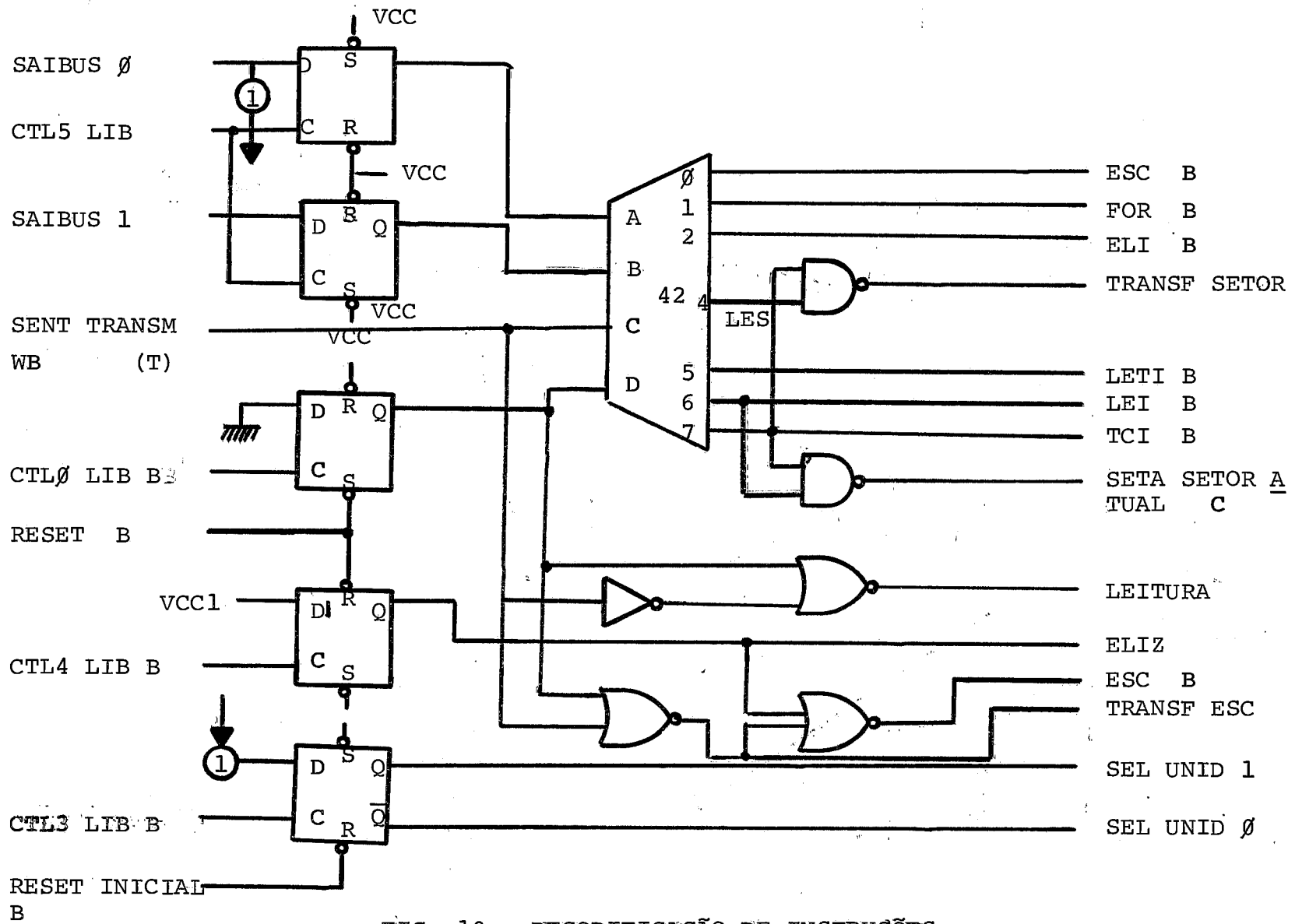


FIG. 10 - DECODIFICAÇÃO DE INSTRUÇÕES.

da pelo CTLØ LIB e SENT TRANSM, dado pelo CANAL.

TRANSF ESC

Indica qualquer operação de escrita que não seja ELIZ.

Estando o sistema estável, então estas linhas quando a tivas comandam as diversas operações de lógica interna que executam as instruções definidas no software. O fim de uma operação será sempre caracterizado por um pulso na linha RESET devido a uma das causas abaixo:

RESET INICIAL

O Controlador é inicializado quando a fonte é ligada.

PULSO FIM ESC

Ocorre sempre ao final de uma operação de escrita de vi do a microinstrução FIM.

PULSO FIM LES

Ocorre durante uma operação de LES, após ter sido de- tetado o último byte de CRC.

ESCRITA INVÁLIDA

Ocorre quando, durante uma instrução ESC, ELI ou ELIZ, o campo de identificação do setor atual apresen- tar erro de CRC. Logicamente teríamos:

ESC INV = (ESC + ELI + ELIZ) E ERI

A deteção de ERI será mostrada adiante na descrição

do circuito de CRC.

PULSO 2º INDEX

A detecção do 2º index indica que após uma revolução completa do disco a operação não foi completada por algum motivo e deve ser cancelada. Os estados deverão fornecer as condições de erro.

PULSO TRILHA INCORRETA

Ocorre quando, durante uma operação que não seja LETI, que inibe esta linha, o cheque do 1º bytes do campo de identificação com o valor da trilha armazenada não for válido. A operação é então cancelada.

PULSO FIM LEI

Durante uma operação de LEI este pulso ocorre ao final do 2º bytes do CRC.

CTL6

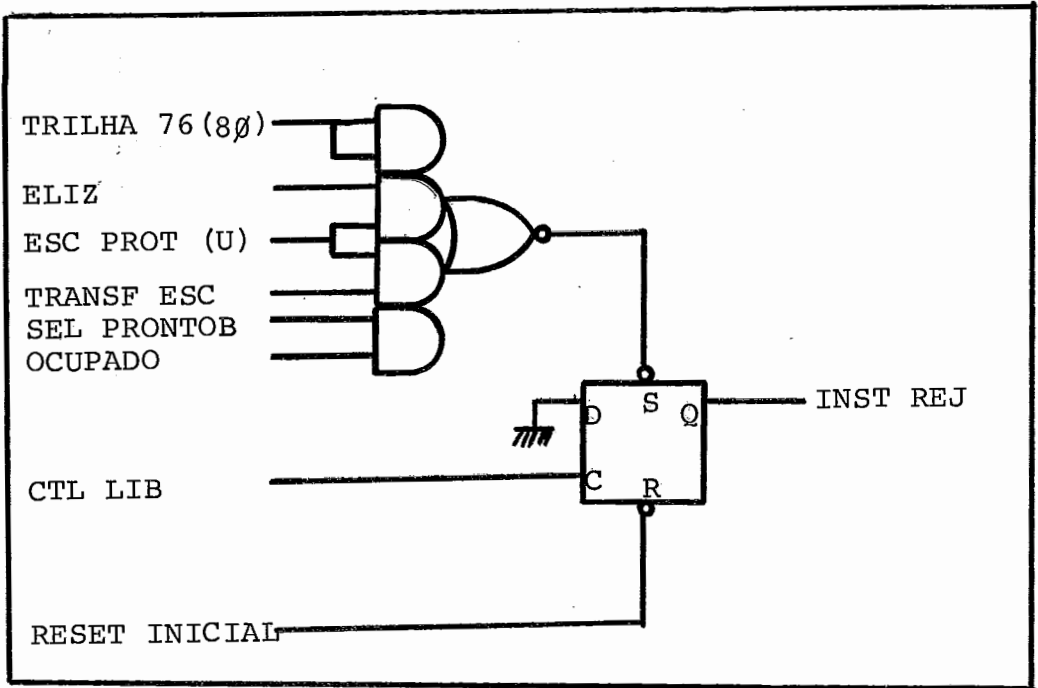
É o reset por software. Cancela a operação em andamento.

PULSO FIM POS

Vem do circuito de posicionamento e indica fim da operação.

INSTRUÇÃO REJEITADA

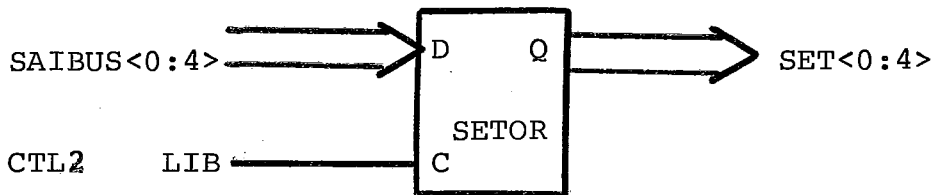
Esta linha deteta as condições de rejeição de instrução definidas por software. É gerada pela lógica abaixo:



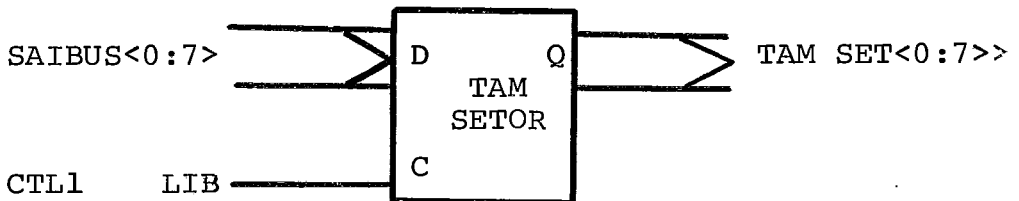
A linha TR < 76 provém do circuito de posicionamento. ESC PROT é um estado da unidade selecionada proveniente de um sensor fotoelétrico que deteta um orifício no diskette. Observe que no início do CTL LIB tenta-se desativar o estado. Se as condições de rejeição permanecem ele é mantido ativado e o Controlador não chega a entrar em OCUPADO pois o RESET fica simultaneamente ativo. Instrução Rejeitada é um estado do Controlador.

6.2 - Registros de Armazenamento: SETOR E TAMANHO DE SETOR

O registro de SETOR é armazenado com o SAIBUS diretamente pelo CTL2 LIB. O seu conteúdo é utilizado pelo circuito de cheque para reconhecimento do setor procurado. Como o máximo de setores é 32 só serão armazenadas 5 linhas do SAIBUS.



O Registro TAMANHO DE SETOR é armazenado com o SAIBUS diretamente pelo CTL1 LIB. O seu conteúdo é utilizado como valor inicial para o Contador de Bytes quando uma MEIV ou MEIE é detetada na leitura ou durante uma operação de escrita quando SEL TAM SETOR é ativada.



6.3 - Posicionamento

A função do circuito é controlar todo e qualquer posicionamento e guardar a informação da trilha posicionada independentemente para cada unidade.

O diagrama de blocos do sistema é apresentado na FIG. 11.

As linhas de entrada são:

SAIBUS Ø-7

É a barra de informação vinda diretamente do TI.

CTL7 LIB

É o pulso de controle fornecido pelo decodificador de instruções quando uma instrução POS é detetada.

HEAD LOAD

Esta linha é proveniente do circuito de controle da cabeça e detecção de Index.

SEL UNIDØ

Seleciona a unidade Ø no seu estado ativo e a unidade 1 no caso oposto. É proveniente também do circuito decodificador de instruções.

READYØ, READY1

São provenientes diretamente de cada unidade e indicam

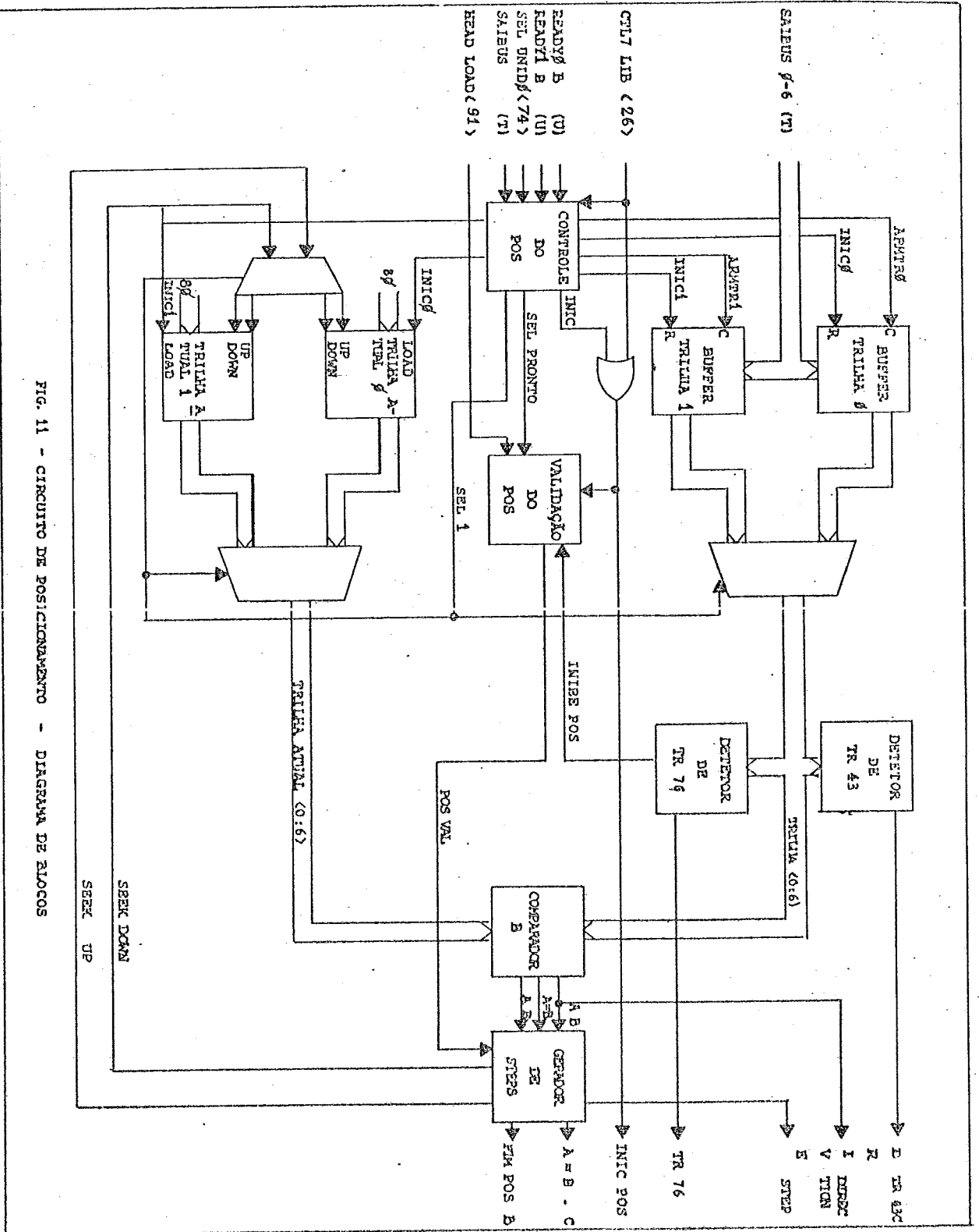


FIG. 11 - CIRCUITO DE POSICIONAMENTO - DIAGRAMA DE BLOCOS

se elas estão ou não prontas.

DESCRIÇÃO GERAL

Cada unidade possui um par de registros denominados Registro de Trilha - contém a trilha a ser posicionada e Registro de Trilha Atual - contém a trilha atualmente posicionada.

Os registros de TRILHA podem ser carregados com o conteúdo do SAIBUS pela linha ARMTR ou zerados com um pulso INIC.

Os registros de TRILHA ATUAL podem ser armazenados com o valor 80 pelo pulso INIC, incrementados ou decrementados, respectivamente, pelo SEEK UP ou SEEK DOWN, gerados pelo bloco Gerador de Steps.

Os selecionadores permitem a escolha dos registros referentes a unidade selecionada.

Normalmente os registros de TRILHA e TRILHA ATUAL contêm a mesma informação, provocando a ativação da linha A=B na saída do comparador.

Quando queremos posicionar armazenamos a nova trilha no Registro TRILHA. Sendo agora $A \neq B$ uma das saídas $A < B$ ou $A > B$ do comparador será ativada. A ocorrência do pulso INIC POS coloca o Controlador no estado OCUPADO ao mesmo tempo que desencadeia a geração de POS VAL. O pulso POS VAL dispara o circui-

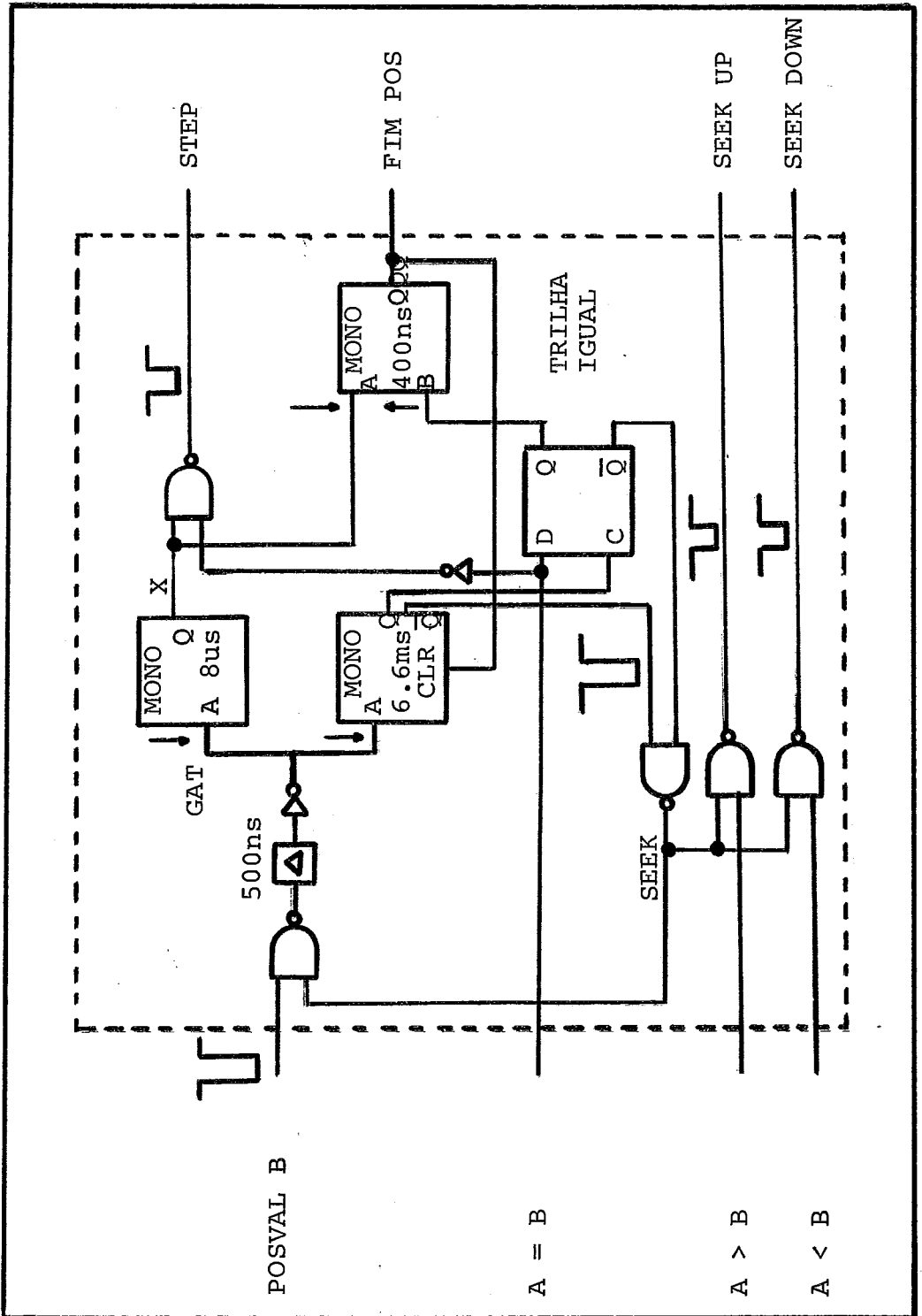
to de controle que, gerando pulsos em STEP, move a cabeça, trilha a trilha, na direção indicada por DIRECTION. Correspondentemente a cada STEP é gerado um SEEK (DOWN ou UP, conforme esteja ativado A<B ou A>B) que altera o conteúdo do registro TRILHA ATUAL. Quando o comparador deteta A=B a geração dos STEP é interrompida e o circuito alcança a estabilidade gerando o pulso FIM POS que liberará o estado OCUPADO do Controlador.

O detetor TR > 43 não influi no posicionamento e sua saída é utilizada para controlar a intensidade da corrente de gravação durante uma operação de escrita.

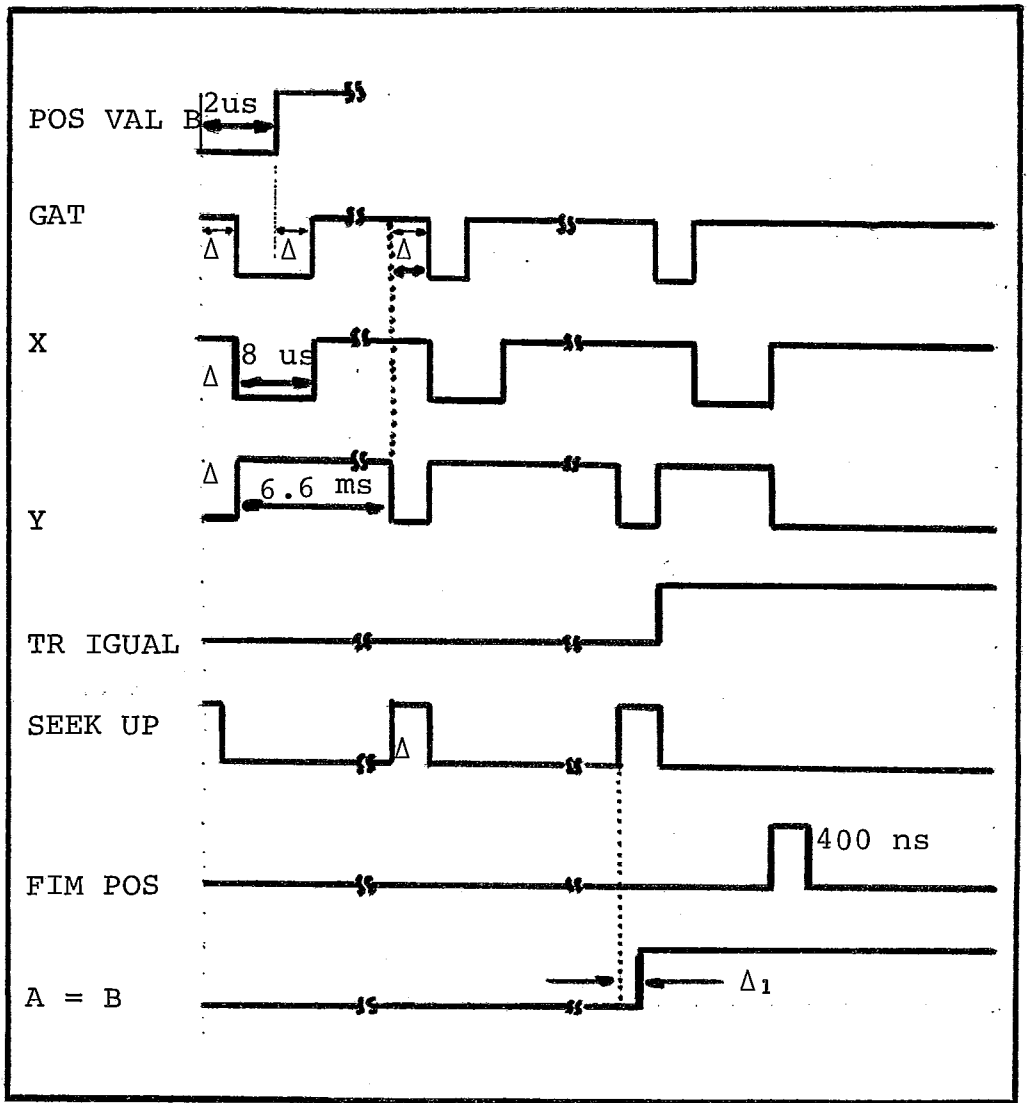
O detetor TR > 76 é necessário para se evitar um posicionamento errôneo. Ele inibe a geração do POS VAL ao mesmo tempo que ativa o estado INSTRUÇÃO REJEITADA.

GERADOR DE STEPS

Em linhas gerais o controle está implementado como a-
baixo:



O diagrama de tempo para o poscionamento de uma trilha a outra com diferença de valor de 2 é o abaixo:



A restrição aos atrasos é que devemos ter $\Delta > \Delta_1$.

Δ_1 = atraso no contador (trilha atual) + atraso dos selecionadores + atraso no comparador.

Atraso máximo nos contadores: $2 \times 47\text{ns} = 94 \text{ ns}$.

Atraso máximo nos selecionadores: 14 ns .

Atraso máximo no Comparador: $2 \times 35\text{ns} = 70 \text{ ns}$.

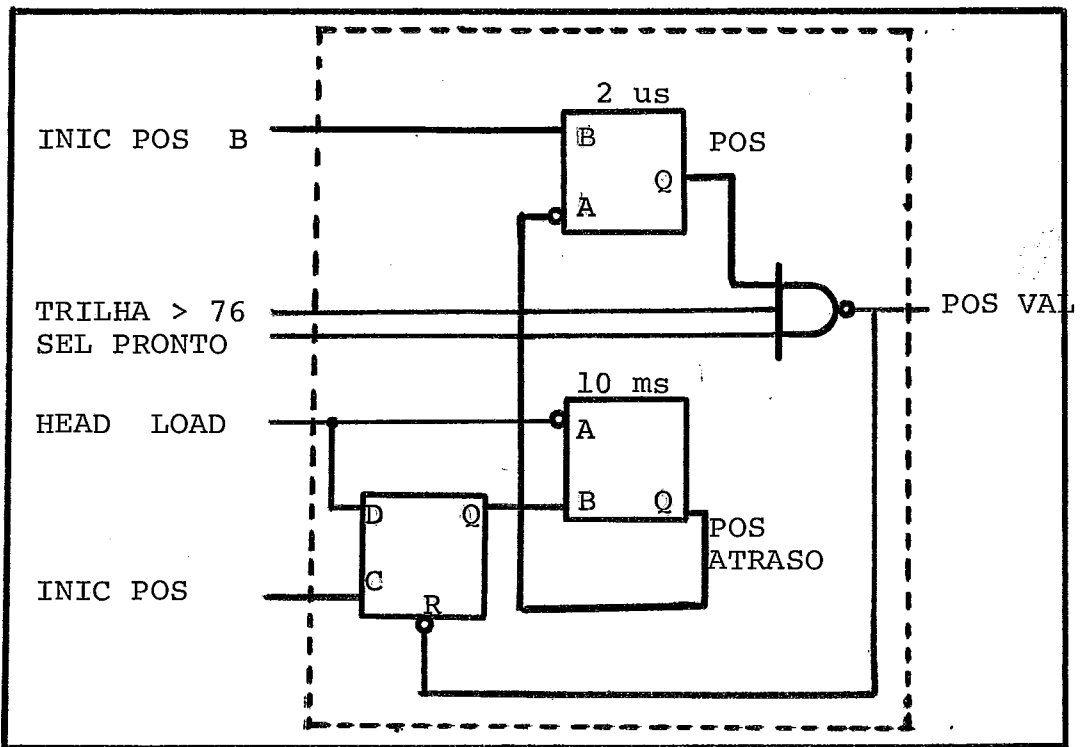
$\Delta_{\text{lmáx.}} = 288 \text{ ns}$.

Foi utilizado $\Delta = 500$ ns.

Como se pode observar, uma vez disparada, a lógica só atinge o estado estável quando $A=B$ for detetado.

Se já temos inicialmente $A=B$ então decorrerá $8\mu s$ antes da ocorrência do FIM POS.

GERAÇÃO DO POS VAL



Ocorrendo INIC POS (CTL 7 LIB ou INIC) podemos ter duas situações:

A CABEÇA ESTÁ ABAIXADA

Antes da ocorrência de INIC POS, HEAD LOAD = 1. Assim que INIC POS sobe, ao mesmo tempo em que o FF é armazenado com o valor 1, os registros da trilha são armazenados com a trilha a ser posicionada (Vide GERA-

ÇÃO DE POS AUT, a seguir) e após um atraso $\Delta t_{\text{máx}}=84\text{ns}$ a saída $A=B$ do comparador vai a \emptyset . Esta linha atua no controle da cabeça forçando $\text{HEAD LOAD} = \emptyset$. Esta transição negativa em HEAD LOAD dispara POS ATRASO , já que a entrada B (do mono) = 1.

Este pulso de 10 ms inibe o gatilho de POS pela transição positiva de INIC POS B , disparando POS na sua transição negativa. Este atraso permite que haja tempo para a cabeça se afastar do disco flexível antes do primeiro pulso de STEP .

Se a cabeça está abaixada, mas a trilha a ser posicionada já é a atual então POS ATRASO não ocorrerá e POS é gerado ao final de INIC POS B .

Como se vê, os registros de trilha são armazenados no início de INIC POS e, na melhor das hipóteses, POS VAL só é gerado ao final de INIC POS . Isto garante que a linha DIRECTION estará estabilizada antes do primeiro STEP .

POS VAL B reseta o FF para evitar que uma posterior transição negativa em HEAD LOAD venha a gatilhar o mono POS ATRASO caso o FF tivesse ficado ativado ($Q=1$).

A CABEÇA ESTÁ LEVANTADA

Neste caso POS é gerado ao final do INIC POS B.

A linha SEL PRONTO é utilizada para se evitar disparar o circuito de controle com a unidade NÃO PRONTA . Neste caso haveria um posicionamento lógico que não corresponderia à realidade física.

CONTROLE DO POSICIONAMENTO

A finalidade deste bloco é gerar os pulsos ARMTR, INIC e SEL PRONTO de acordo com o tipo de posicionamento desejado. A transição positiva de ARM TR Ø (1) carrega o registro de Trilha Ø (1) com o valor do SAIBUS <0:6>. Os pulsos INICØ ou INIC1 zeram o registro de TRILHAØ (1) e armazenam 8Ø nos registros de TRILHA ATUALØ (1). O pulso INIC é um OU de INICØ e INIC1.

Logicamente podemos ter 3 tipos diferentes de posicionamento:

- 1 - POSICIONAMENTO AUTOMÁTICO NA TRØ;
- 2 - POSICIONAMENTO FÍSICO NA TRØ;
- 3 - POSICIONAMENTO NORMAL.

POSICIONAMENTO AUTOMÁTICO NA TRØ

Neste caso o circuito de controle funciona como uma máquina sequencial que deve fornecer pulsos de 500 ns

nas linhas INIC e INICØ (1) (conforme a unidade selecionada) quando:

- a - Uma unidade selecionada passa do estado NÃO PRONTA para PRONTA;
- b - Uma unidade PRONTA, anteriormente não selecionada é selecionada. Só poderá haver um novo POS AUT se a unidade passar para o estado NÃO PRONTA.

Como temos somente 77 trilhas o valor 8Ø carregado na TRILHA ATUAL garantirá que a TRØ será detectada. À medida que os pulsos de STEP são gerados, o registro de TRILHA ATUAL é decrementado a cada SEEK DOWN e quando TRØ é alcançada, os pulsos restantes não mais afetarão a unidade pois são inibidos internamente. Portanto não precisamos ter a preocupação de inibir os STEP's desnecessários. O resultado desta filosofia é que a duração deste posicionamento é constante e igual a $80 \times 66 = 528$ ms. INIC POS é gerado por INIC e, portanto, o Controlador fica OCUPADO durante este tempo total.

POSICIONAMENTO FÍSICO NA TRØ

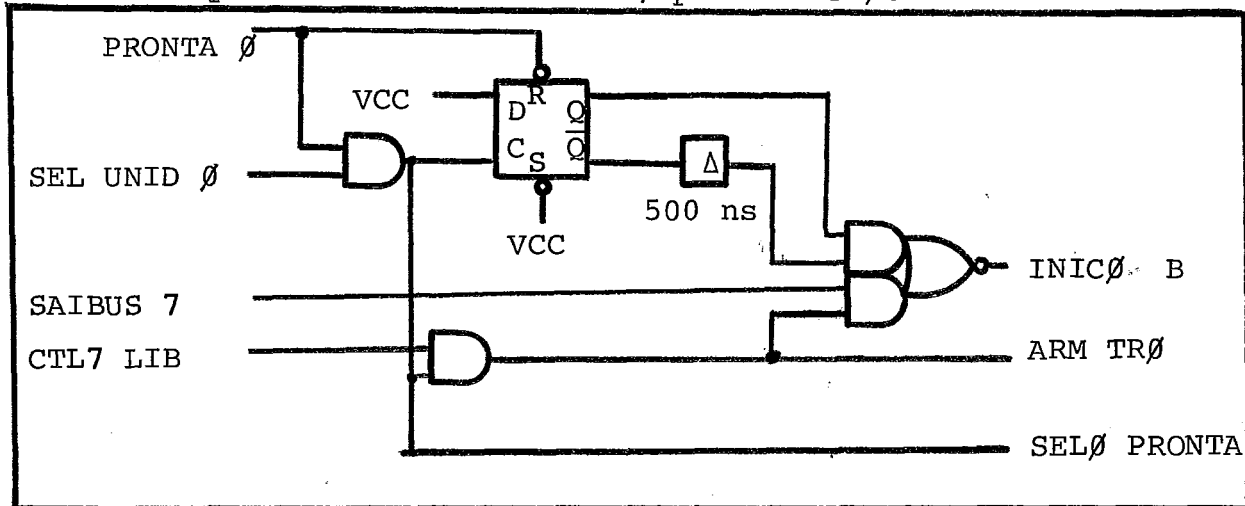
Este tipo de posicionamento é obtido sempre que deixarmos o SAIBUS7 ligado e decodificarmos CTL7. Daí a necessidade do SAIBUS7 como entrada neste bloco. A seqüência provocada por este posicionamento é semelhante ao caso anterior. A única exceção é que os

pulsos INIC são imagens do CTL7 LIB e, portanto, com uma duração aproximada de 4 us. O conteúdo dos 6 bits menos significativos do SAIBUS não interessa pois, embora ARM TR seja gerado, INIC predomina, zerando o registro de TRILHA durante toda a ocorrência deste. Esta instrução é necessária pois é a única possível de recuperar o sincronismo do posicionamento por software.

POSICIONAMENTO NORMAL

Ocorre para um conteúdo do SAIBUS ≤ 76 . Neste caso os pulsos de INIC não são gerados. O registro de TRILHA é carregado com o valor do SAIBUS <0:6> e o registro de TRILHA ATUAL mantém inicialmente o seu valor armazenado. O pulso ARM TR é imagem do CTL7 LIB.

O circuito é duplo, cada um para gerar os sinais correspondentes a sua unidade. Está, esquematicamente implementado como abaixo, para UNID \emptyset :



6.4 - Controle da Cabeça

Este pequeno subsistema possui duas finalidades principais:

- 1 - Deve detectar a ocorrência do primeiro e segundo pulsos de INDEX após o início da execução de uma instrução.

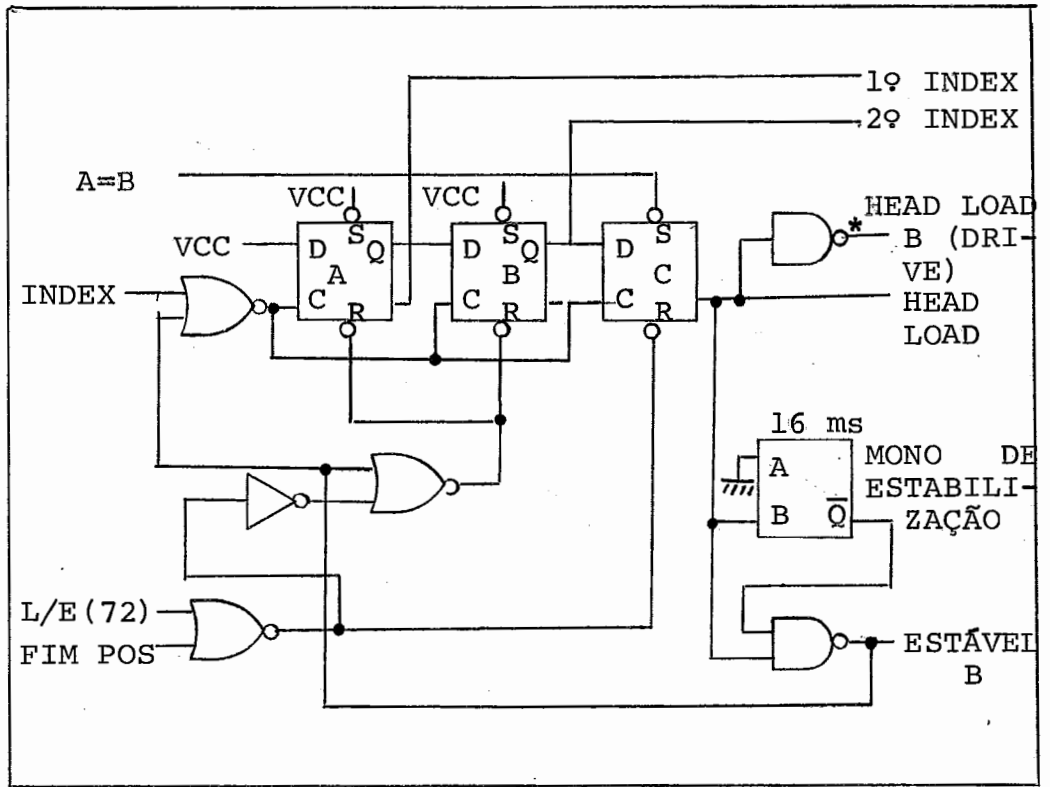
A detecção do primeiro pulso é feita pela linha 1º INDEX e sincroniza o início de uma instrução FOR ou LETI. A linha 2º INDEX é utilizada para forçar um fim de execução.

- 2 - Fazer o controle da cabeça através da linha HEAD LOAD e ESTÁVEL.

Este controle deve ser tal que a linha HEAD LOAD deve ser desativada depois de duas revoluções completas após a última instrução de transferência, evitando um desgaste desnecessário do diskette.

A linha ESTÁVEL só deve estar ativa 16 ms após o abaixamento da cabeça, tempo de estabilização fornecido pelo fabricante.

O circuito foi implementado como abaixo:



A linha A=B, proveniente do circuito de posicionamento, garante que, durante toda a duração de um posicionamento, a cabeça permaneça levantada, atuando no SET do FFC.

A linha L/E origina-se no circuito de decodificação de instrução e um pulso ocorre sempre que uma instrução de L/E é detetada. A ocorrência de L/E ou FIM POS provoca um RESET nos três FF's. O contador de INDEX (FFA e FFB) é zerado e HEAD LOAD a tivado. A transição positiva de HEAD LOAD dispara o mono de estabilização e a linha ESTÁVEL B (cabeça abaixada e estabilizada) só vai a zero, após o atraso de 16 ms. Durante este tempo os FF's A e B perman

necem zerados e o pulso de INDEX que atua no CLOCK dos três FF's é inibido.

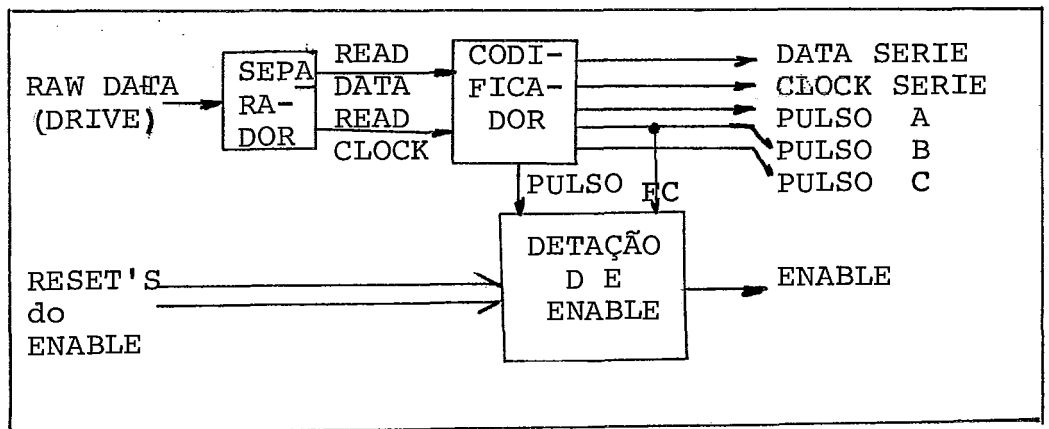
A saída ESTÁVEL também atua no circuito de sincronismo, só liberando este quando a cabeça está estabilizada.

Se uma instrução L/E ocorre e a cabeça já está abaixada, então o mono não é disparado.

Uma vez estabilizada a cabeça, os pulsos de Index são liberados e deslocam 1 sucessivamente de A a C. Quando o FFC é gatilhado com 1, HEAD LOAD é desativado e a cabeça é levantada.

6.5 - Sincronismo

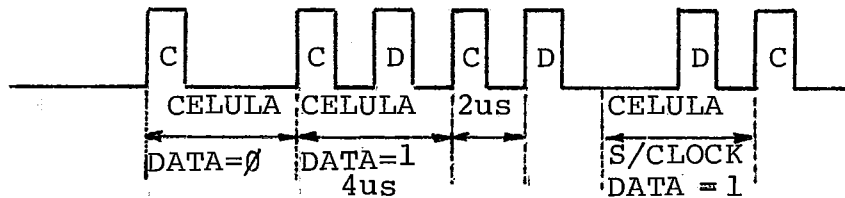
A função deste subsistema é detetar as condições que antecedem uma possível ocorrência de Marca Especial e se confirmada, gerar pulsos que fornecerão a informação e a sincronização durante o estado de Leitura. Consiste basicamente de 3 blocos distintos, de funções bem definidas, e interrelacionadas da forma abaixo:



6.5.1 - SEPARADOR PLL

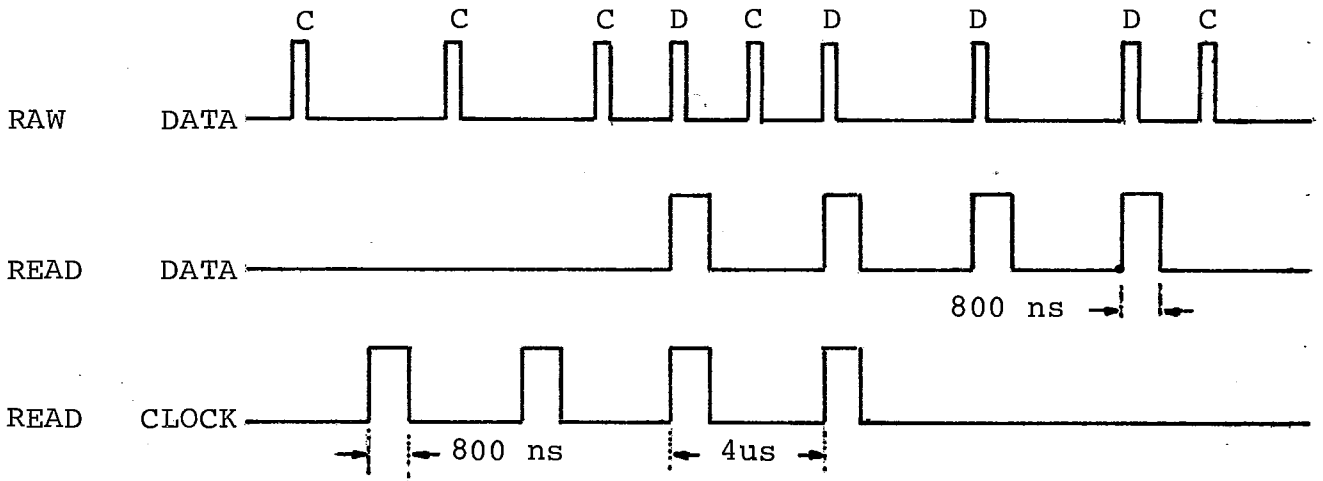
A informação gravada no disco é codificada em FM de tal forma que:

- a - existe um pulso de CLOCK no início de cada célula de informação, exceto em algumas células das Marcas Especiais.
- b - Um bit de informação 1 é representado por um pulso no meio da célula.



Sabendo que a cada pulso corresponde uma inversão do fluxo magnético, então o que temos gravado é uma alternância do fluxo correspondendo ora aos CLOCKS ora aos BITS 1 da informação. A cada uma destas inversões do fluxo corresponde um pulso de ≈ 200 us na linha RAW-DATA.

O circuito foi projetado de forma a fornecer pulsos coincidentes de 800 ns nas saídas READ DATA e READ CLOCK se existe, respectivamente, o pulso de informação 1 e o pulso de CLOCK.



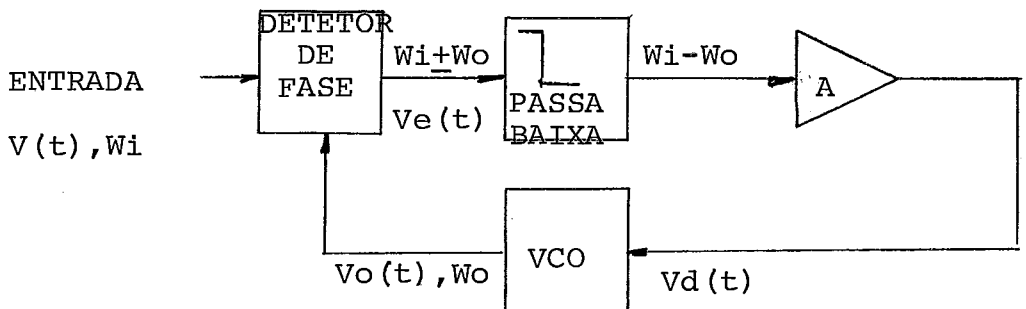
Durante a leitura de um disco, em RAW DATA observa-se distorção e variação da frequência. Uma forma de variação da frequência (lenta) é atribuída a mudanças na velocidade do disco (RPM). Esta mudança pode ser tão grande quanto 5% de unidade para unidade. Variações instantâneas de velocidade (rápidas) ou "peak shift" causadas pela interface disco/cabeça e a falta de CLOCK nas Marcas Especiais são outros problemas a serem enfrentados quando se tenta recuperar a informação.

A melhor solução encontrada foi a utilização de um "Phase Locked Loop" (PLL) na geração do sinal de amostragem. O PLL é um esquema que acompanha a variação lenta de fase e frequência e não responde a variações rápidas, predizendo a partir do tempo médio das células anteriores o centro da próxima célula de informação. Além do mais ele é tal que, durante a leitura de uma Marca Especial, permanece em síncro-

nismo a despeito da falta do pulso de CLOCK.

A TEORIA DO PLL

O PLL é um sistema realimentado cuja teoria data da década dos 30's. Consiste basicamente de um detector de fase, um filtro passa baixa e um oscilador controlando a tensão (VCO).



A aplicação desta técnica é de vital importância em circuitos que exigem uma grande imunidade a ruído, alta frequência de operação, alta seletividade e banda de passagem ajustável.

COMO FUNCIONA

O PLL é semelhante a qualquer sistema realimentado com exceção de que a grandeza tratada é a fase ao contrário da tensão ou corrente como nos circuitos convencionais.

Não se aplicando nenhum sinal na entrada o sistema tende a oscilar em uma frequência própria denominada f_o

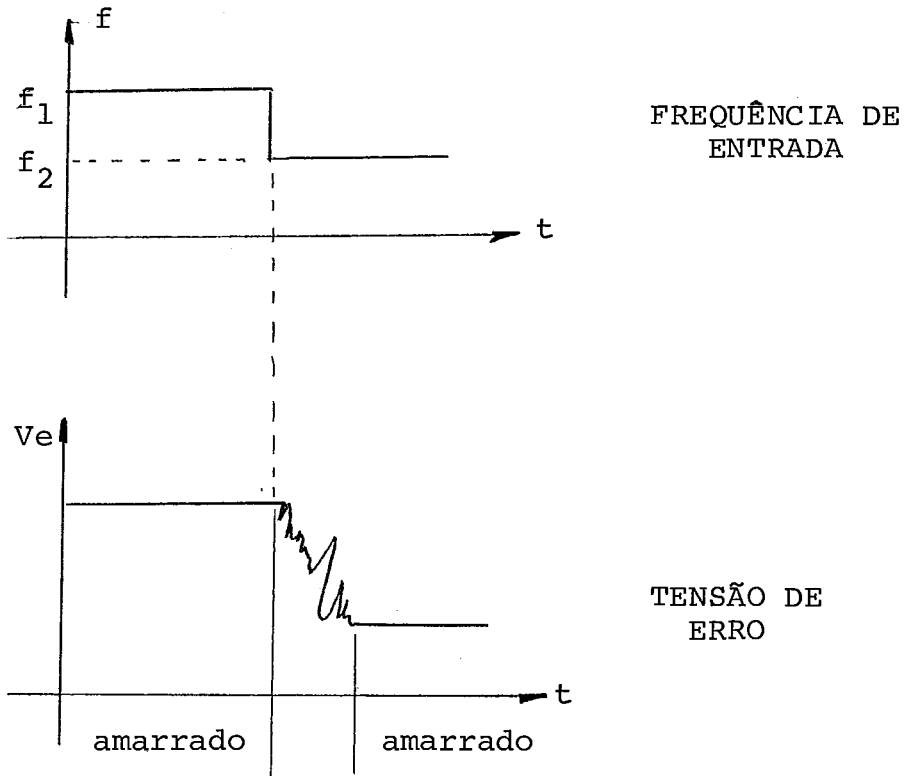
ou frequência livre. Existe um nível DC constante aplicado a entrada da VCO que mantém a oscilação.

Suponhamos agora um sinal de frequência W_i ($2\pi f_i$) aplicado a entrada do PLL. O detetor de fase pode ser visto como um misturador que gera $W_i \pm W_o$. Sendo a frequência a derivada no tempo da fase, os erros de fase e frequência podem ser relacionados como:

$$\Delta W = \frac{d\theta_o}{dt}$$

onde ΔW é a separação instantânea da frequência e θ_o é a diferença de fase instantânea. Se $W_i \gg W_o$ este batimento ΔW é atenuado e não aparece na saída do filtro. À medida que W_i se aproxima de W_o o batimento é menos atenuado e a frequência da VCO será modificada por este. Se durante este processo a frequência W_o da VCO se aproxima de W_i , então $\frac{d\theta_o}{dt}$ decresce, permitindo maior informação a ser transmitida pelo filtro. A saída do comparador passa a ser uma função variável no tempo. À medida que W_i tende para W_o , maior é a amplitude do erro e mais rapidamente a frequência do batimento tende a ZERO. Quando amarrado, ΔW é igual a zero e somente uma componente DC de erro permanece, suficiente para manter a frequência da VCO afastada de sua frequência livre f_o .

O gráfico para a voltagem de erro $V_e(t)$ é o abaixo:



A realimentação permite que a frequência ω_o acompanhe a frequência ω_i dentro de um intervalo conhecido como LOCK RANGE ou TRACKING RANGE. Sendo a realimentação do sistema DC, pode-se ver que o filtro não influi na extensão do LOCK RANGE e este só depende da voltagem de erro capaz de ser gerada e o correspondente desvio de frequência do VCO.

A faixa para a qual o PLL consegue amarrar com o sinal de entrada é denominada CAPTURE RANGE e depende diretamente do filtro passa baixa e do ganho de malha fechada. Deve-se observar que temos necessariamente $\text{CAPTURE RANGE} \leq \text{LOCK RANGE}$.

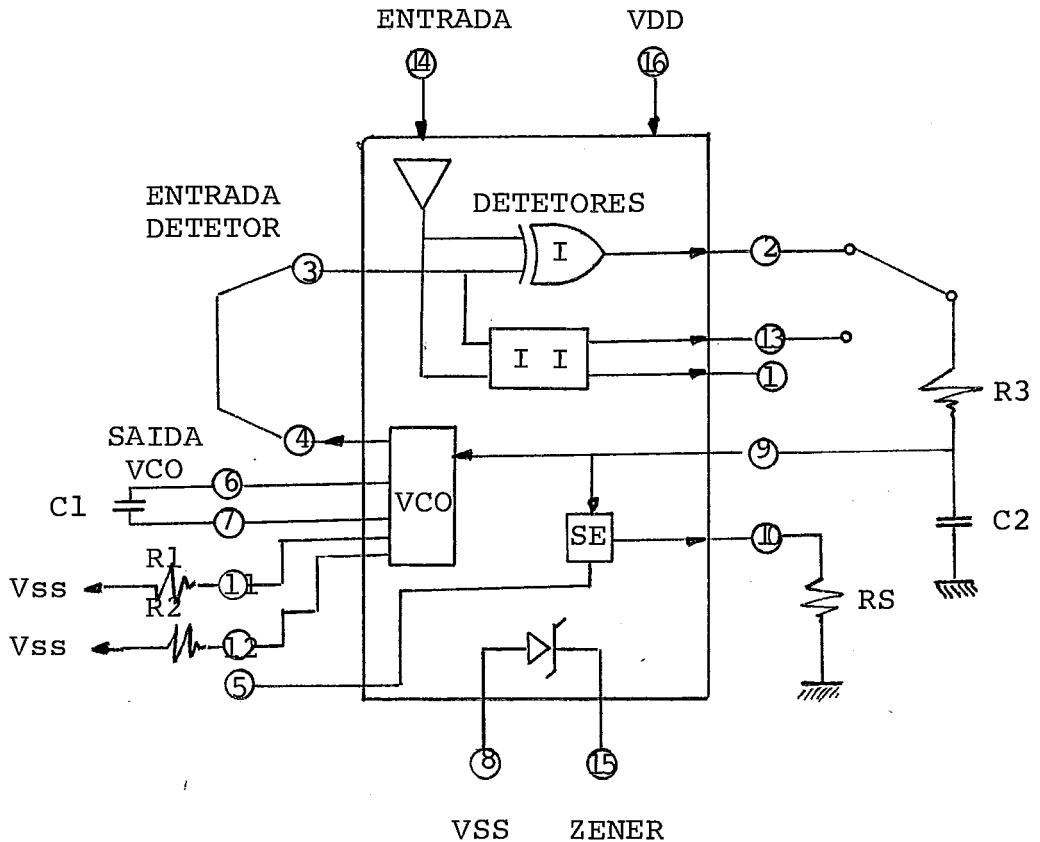
mais sensível a ruídos.

A performance de um PLL dependerá diretamente das características do seu detetor de fase, as quais podem variar bastante dependendo do uso programado.

O CD-4046A

O diagrama de blocos do Separador pode ser visto na FIG. 12.

Na implementação do PLL utilizamos a pastilha CMOS CD-4046A comercializada pela RCA e encontrada no mercado nacional. Ela engloba no seu interior um PLL completo cujo esquema pode ser visto abaixo:



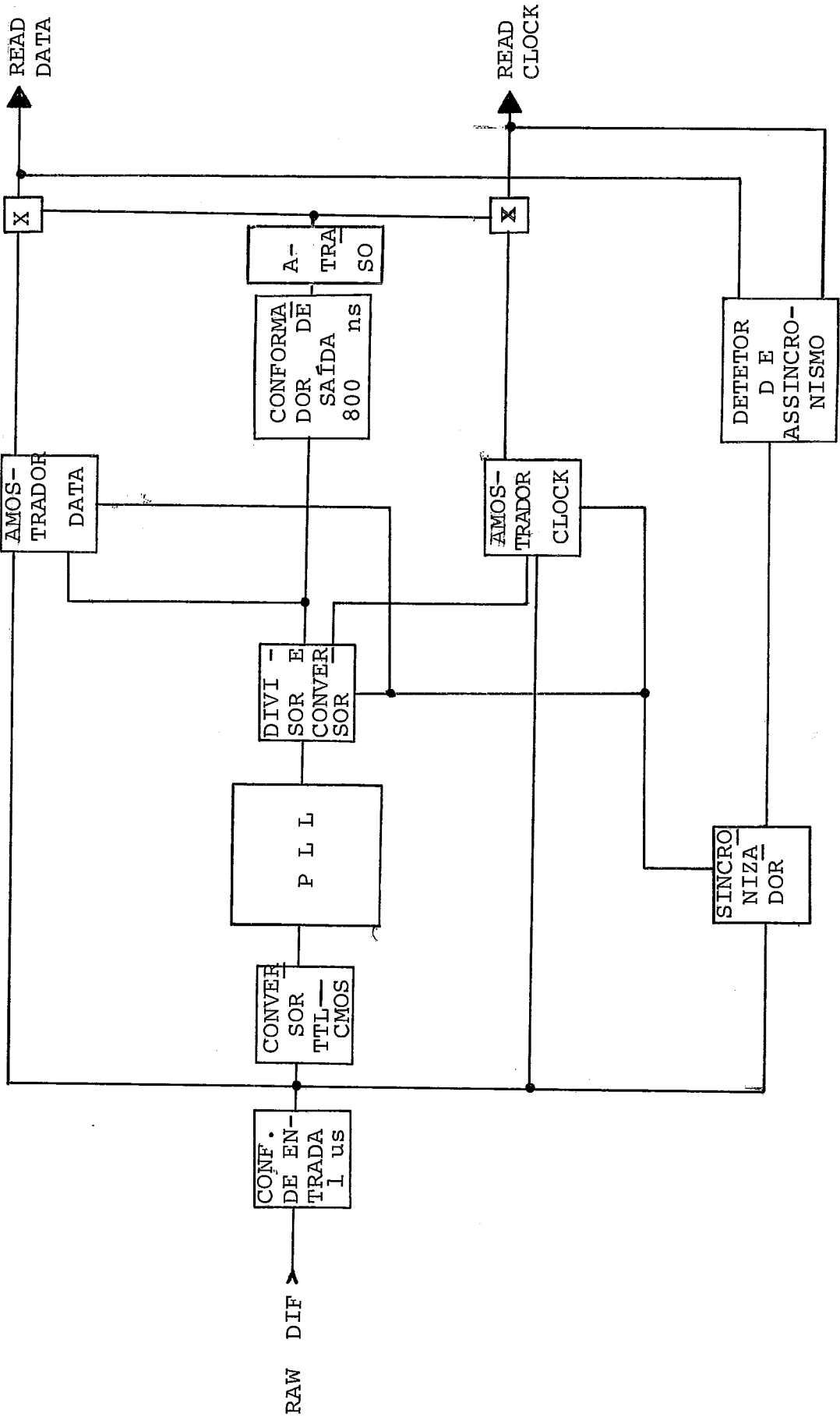


FIG. 12 - SEPARADOR - DIAGRAMA DE BLOCOS

As características elétricas da pastilha são:

CONSUMO: 600 uW a 10 KHz. 200 mW máximo.

VDD-VSS: -0,5 V a 15 V.
Recomendado: VDD-VSS: 5 a 15 V
ENTRADA: VDD a VSS

VCO: Freq. Max.: 1.2 MHz
Estabilidade: 600 ppm/°C
Linearidade: 1%
Voltagem de Saída: 10 Vpp (para VDD-
-VSS = 10 V).
Ciclo de trabalho: 50%
Tempo de subida e descida: 50 ns.

Os pinos 1, 15, 10, 8 e 5 não foram utilizados. Os componentes R1, R2, C1 definem a frequência livre fo e o LOCK RANGE. R3 e C2 formam o passa-baixa e definem o CAPTURE RANGE, sendo este inversamente proporcional a raiz quadrada da constante R3C2 do filtro.

A pastilha fornece como opção a utilização de um dos dois detetores de fase construídos internamente. Foi utilizado o detetor I (OU-exclusivo) porque ele apresenta as duas características fundamentais para o desempenho do Separador:

1 - A defasagem entre o sinal e a saída do VCO é 90° na frequência central, aproximando de 0° e 180°

nos extremos do LOCK RANGE.

2 - O PLL amarra-se em harmônicos da frequência central.

A pastilha foi projetada para apresentar as seguintes características:

$$f_0 = 500 \text{ KHz}$$

$$2f_L = \text{LOCK RANGE} = 40\% f_0$$

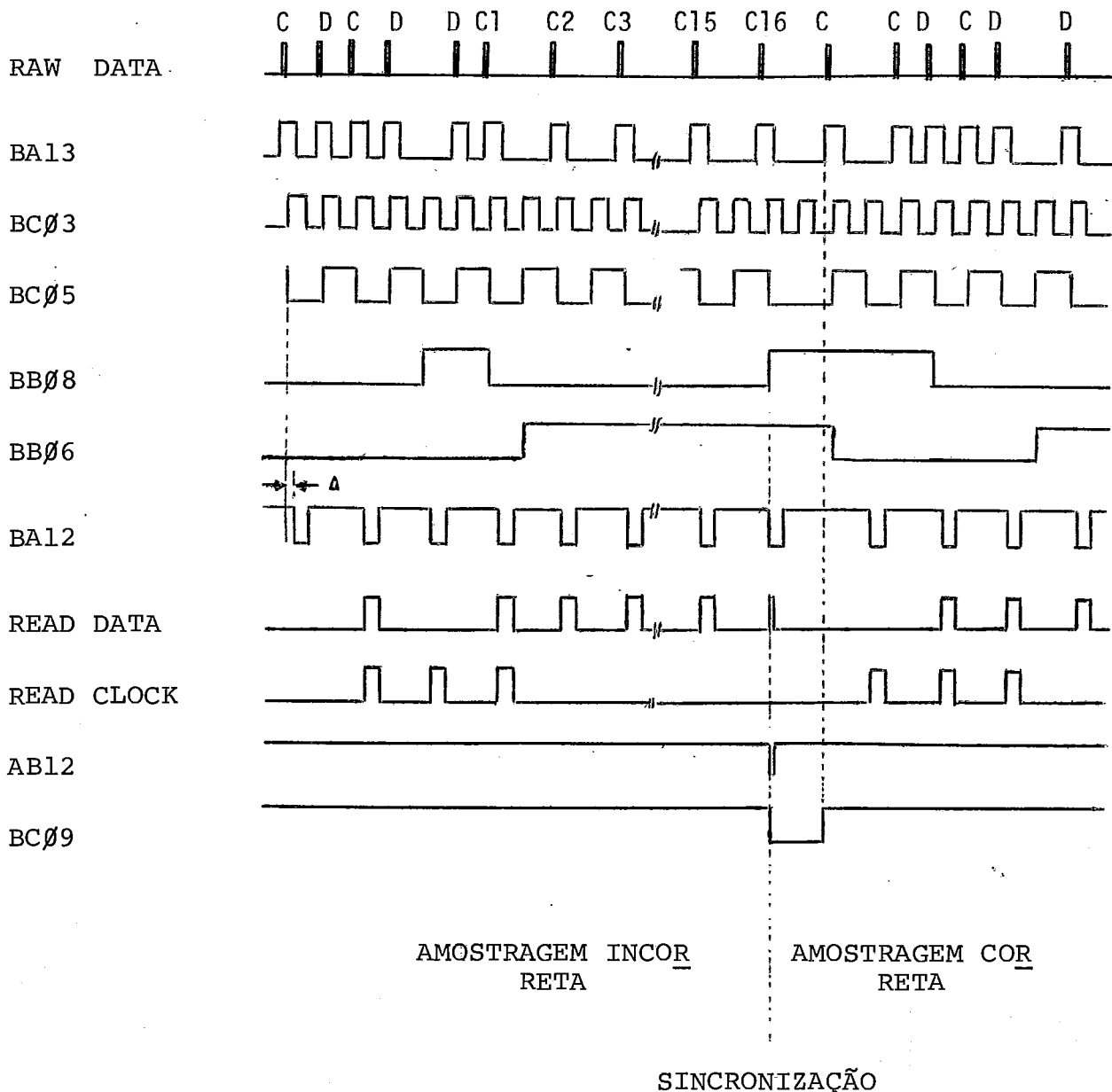
$$2f_c = \text{CAPTURE RANGE} = 20\% f_0$$

A razão destes valores é explicada a seguir.

O sinal RAW DATA, depois de conformado e convertido aos níveis do CMOS é jogado na entrada do PLL (pino 14). Devido a codificação utilizada, este sinal pode apresentar as frequências de 500 KHz ou 250 KHz e a característica (2) do detetor garante-me que para qualquer uma das condições existirá uma realimentação suficiente para manter o PLL amarrado. Esta característica explica porque o sistema mantém o sincronismo mesmo na ausência de pulsos de clocks nas Marcas Especiais.

A saída do VCO dividida por 2 é usada diretamente como sinal de amostragem, uma vez que a característica (1) do detetor garante que ele ocorrerá defasado 90° do sinal de entrada. Para que se possa sempre garantir uma amostragem próxima de 90° se fez o LOCK RAN-

GE (40% fo) >> VARIAÇÃO RPM (5%). Desse modo, mesmo para uma variação máxima de frequência de rotação, estaremos próximos do meio da célula durante a amostragem.



Δ - Atraso de CC.

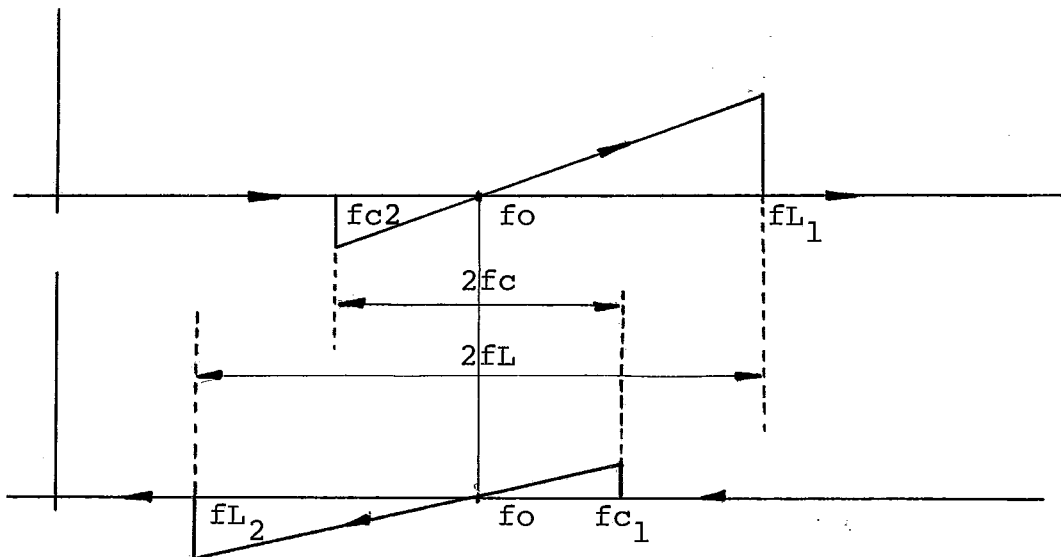
FIG. 13 - SEPARADOR - D I A G R A M A D E T E M P O

Em notação representamos:

$$\text{CAPTURE RANGE} = 2 W_c$$

$$\text{LOCK RANGE} = 2 W_L$$

Uma função de transferência típica frequência-voltagem para o PLL é a mostrada abaixo:



No dimensionamento do filtro passa baixa, existe um compromisso entre o tempo que o sistema leva para amarrar a uma dada frequência W_i contida no CAPTURE RANGE e a sensibilidade do sistema a variações rápidas (imunidade a ruídos). Se o filtro possui uma frequência de corte muito baixa, o CAPTURE RANGE é estreito e o PLL leva um tempo maior para a amarração, ficando por outro lado com menor sensibilidade a variações. Neste caso a taxa com que a frequência de entrada varia dentro do LOCK RANGE é limitada a um valor máximo, abaixo do qual o PLL ainda a acompanha.

No caso oposto, quando o CAPTURE RANGE é largo o sistema amarra mais rapidamente, tornando-se entretanto

Devemos ter $CAPTURE\ RANGE \geq 5\%$ fo para garantir a amarração sob qualquer variação. Um valor baixo para $CAPTURE\ RANGE$ pode acarretar um tempo de amarração grande o que não é bom para a performance do sistema. Como foi implementado, a detecção do sincronismo exige condições que a tornem independente do tempo de amarração, devendo este ser tratado não como uma característica crítica, mas como uma característica capaz de fornecer melhor desempenho ao Separador. O valor 20% é um resultado prático.

FUNCIONAMENTO DO CIRCUITO

O funcionamento do Separador pode ser acompanhado perfeitamente a partir do diagrama de tempo da FIG. 13 que se refere ao esquema detalhado mostrado na FIG.14.

Merece uma explicação adicional a alteração do sincronismo. Inicialmente o Separador pode ter 2 comportamentos: ele amostra os pulsos de $CLOCK$ e $DATA$ gerando $READ\ DATA$ e $READ\ CLOCK$ corretamente nas saídas correspondentes ou amostra invertido gerando $READ\ DATA$ na linha $READ\ CLOCK$ e vice-versa.

No primeiro caso, enquanto não houver uma falta de $CLOCK$, os pulsos de $READ\ CLOCK$ manterão o $CONTADOR$ (AB) zerado. Uma falta de 3 $CLOCKS$ consecutivos é o máximo que pode ocorrer durante uma Marca Especial e neste caso o Contador vai até 3, sendo zerado no

próximo READ CLOCK.

Suponhamos agora que o sistema tenha se amarrado incorretamente. Neste caso a ocorrência de um CLOCK significará um pulso em READ DATA e uma informação 1 um pulso em READ CLOCK. Se estivermos em um conjunto de ZEROS, a cada um deles significará uma falta de CLOCK e os pulsos em READ DATA incrementarão o Contador. O 16º zero consecutivo gerará um CARRY que resetará BC09, resetando o divisor e os FF's de amostragem. O próximo pulso em BA13, que corresponde a um CLOCK realmente, seta BC09 e libera o divisor e amostradores, e a amostragem agora invertida é correta.

Durante o período desincronizado, o Separador fornece pulsos em READ DATA, com o correspondente ou não em READ CLOCK. No caso específico de se estar sobre uma Marca Especial pode ser gerado até 3 zeros no máximo (correspondendo aos 3 clocks que faltam).

O circuito de ENABLE (será analisado posteriormente) coloca como restrição para a liberação da detecção das Marcas Especiais a ocorrência consecutiva dos 4 eventos abaixo:

- a - Serem detetados pelo menos 15 zeros;
- b - Ocorrem 2 bits de informação 1 com CLOCK;
- c - Ocorre um bit 1 com falta CLOCK;
- d - Os próximos 5 bits de informação e CLOCK junto com os 3 já deletados são a configuração de uma

Marca Especial.

Como pode ser verificado, uma desincronização jamais satisfará a restrição quádrupla acima. Estas condições acima garantem uma confiabilidade de 100% na sincronização.

CÁLCULO DO PLL

Os valores de R_1 , R_2 , R_3 , C_1 e C_2 foram obtidos para satisfazer a performance teórica. O fabricante fornece um conjunto de gráficos e fórmulas (vide referências) dos quais se obtém os valores teóricos. Usando valores comerciais, fixa-se R_2 e C_1 e medindo-se f_o ajusta-se o valor de R_1 com um potenciômetro. Como a diferença entre o valor calculado e o medido para R_1 foge bem além das tolerâncias, deve-se supor que os gráficos contêm uma boa dose de impressão.

VALORES TEÓRICOS

$$f_o = 500 \text{ KHz} \quad 2 f_L = 200 \text{ KHz} \quad 2 f_C = 100 \text{ KHz}$$

$$R_1 = 27 \text{ K}; \quad R_2 = 22 \text{ K}; \quad R_3 = 4\text{K}7; \quad C_1 = 220 \text{ pF}; \quad C_2 = 680 \text{ pF}$$

1a. MONTAGEM

$$R_1 = 18 \text{ K}; \quad R_2 = 22\text{K}; \quad R_3 = 4\text{K}7; \quad C_1 = 220 \text{ pF}; \quad C_2 = 680 \text{ pF}$$

MEDIDAS

$$f_o = 500 \text{ KHz (ajustado por } R_1);$$

fL1 = 606 KHz (120% fo) LOCK RANGE \cong 48% fo
fL2 = 364 KHz (72% fo)
fC1 = 588 KHz (118% fo) CAPTURE RANGE \cong 38% fo
fC2 = 400 KHz (80% fo)

2a. MONTAGEM

R1 = 16 K; R2 = 22K; R3 = 4K7; C1 = 220 pF; C2 = 680pF

MEDIDAS

fo = 500 KHz (ajustado por R1);
fL1= 620 KHz (123% fo) LOCK RANGE \cong 53% fo
fL2= 350 KHz (70% fo)
fC1= 588 KHz (118% fo) CAPTURE RANGE \cong 38% fo
fC2= 400 KHz (80% fo)

Para as duas placas acima foram feitos testes de banca da utilizando.se um Gerador de Palavras capaz de simular a entrada RAW DATA para diversas situações. Simulou-se as variações rápidas de velocidade, entrando com uma frequência modulada de até 10% no CLOCK EXTERNO do Gerador. Mesmo para este valor de variação, o circuito decodificou a saída corretamente.

6.5.2 - CODIFICADOR

A partir das entradas READ DATA e READ CLOCK são geradas quatro linhas que atuam na Leitura:

PULSO A

De uso geral.

PULSO B

A transição positiva gatilha os deserializadores de DATA, CRC e CLOCK, durante a Leitura. É importante observar que a falta de um pulso de CLOCK corre sempre numa célula com pulso de informação de modo que para este caso um \emptyset é forçado no registro de CLOCK e um 1 no registro de DATA.

PULSO C

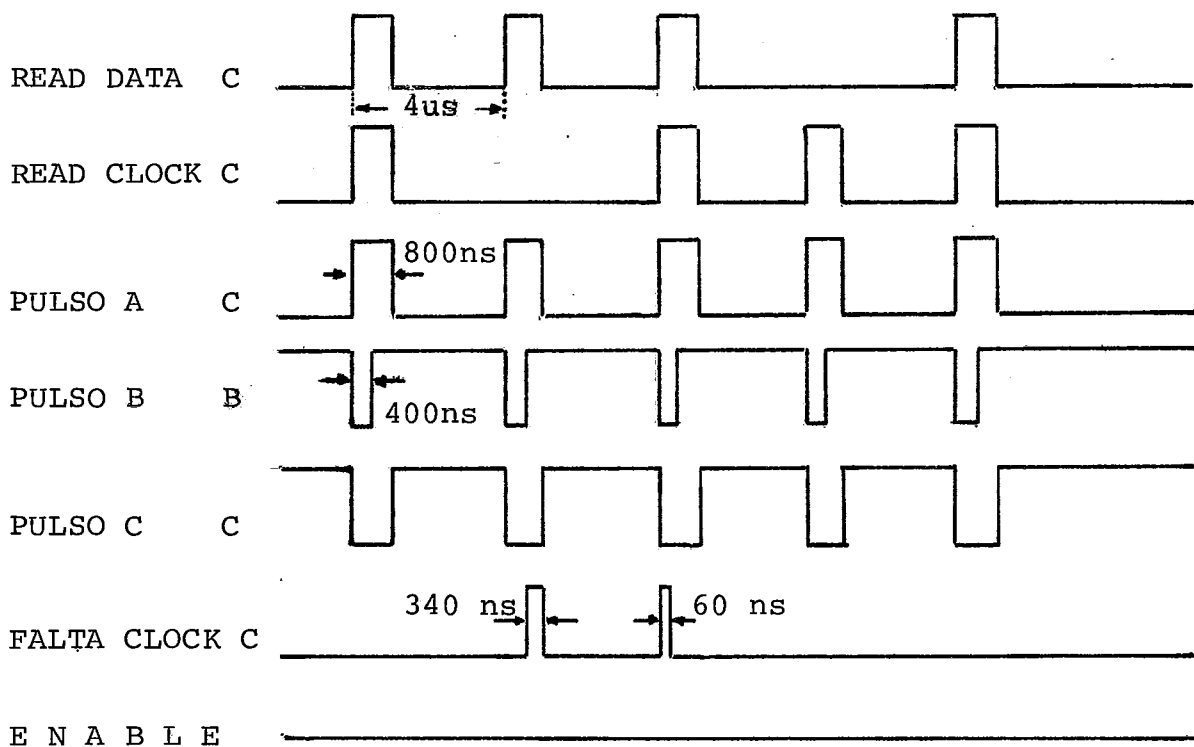
A transição positiva deste gatilha o Contador de Bits, fazendo com que este altere de estado somente ~400 ns após a informação. Este fato é importante pois permitirá atuarmos com decodificadores junto aos registros mencionados acima e obter uma resposta já estabilizada antes da alteração de estado dos contadores.

FALTA CLOCK C

Um pulso nesta linha sempre ocorrerá quando detetarmos uma ausência de CLOCK com DATA=1. A transição positiva deste pulso é utilizada para gatilhar o circuito de ENABLE, que libera o contador de bits possibili

tando a operação em Leitura. O próprio ENABLE inibe a geração de pulsos posteriores.

O diagrama de tempo para os pulsos acima é visto na FIG. 15.



O B S: - No caso do primeiro pulso FALTA CLOCK ativar ENABLE, a forma de onda correspondente para FALTA CLOCK será:

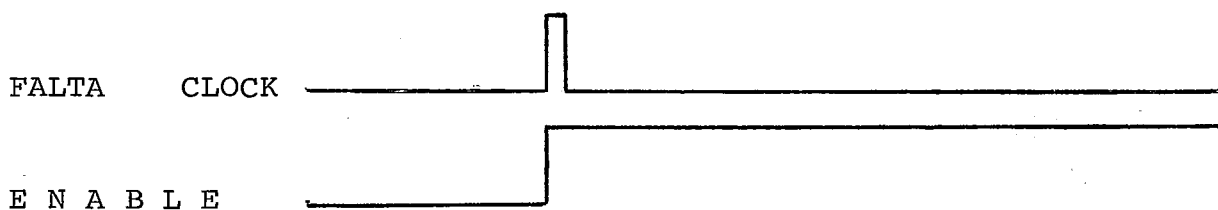


FIG. 15 - CODIFICADOR.

6.5.3 - DETEÇÃO DO ENABLE

Para que todo o processo de leitura se desendeie é necessário que a informação que chega para ser reconhecida como os bits de determinação byte e este como o byte de determinado campo. Em termos de deteção a trilha não é encarada como um conjunto de setores, mas realmente como um conjunto de campos particulares precedidos de uma Marca Especial. Esta Marca servirá tanto para a identificação como para a sincronização no campo pois a cada campo corresponde um determinado tipo de ME e cada ME possui três (3) características básicas:

- 1 - São precedidos de um preâmbulo de ZEROS;
- 2 - Possuem os 5 bits mais significativos iguais a 1;
- 3 - Falta CLOCK na terceira célula mais significativa.

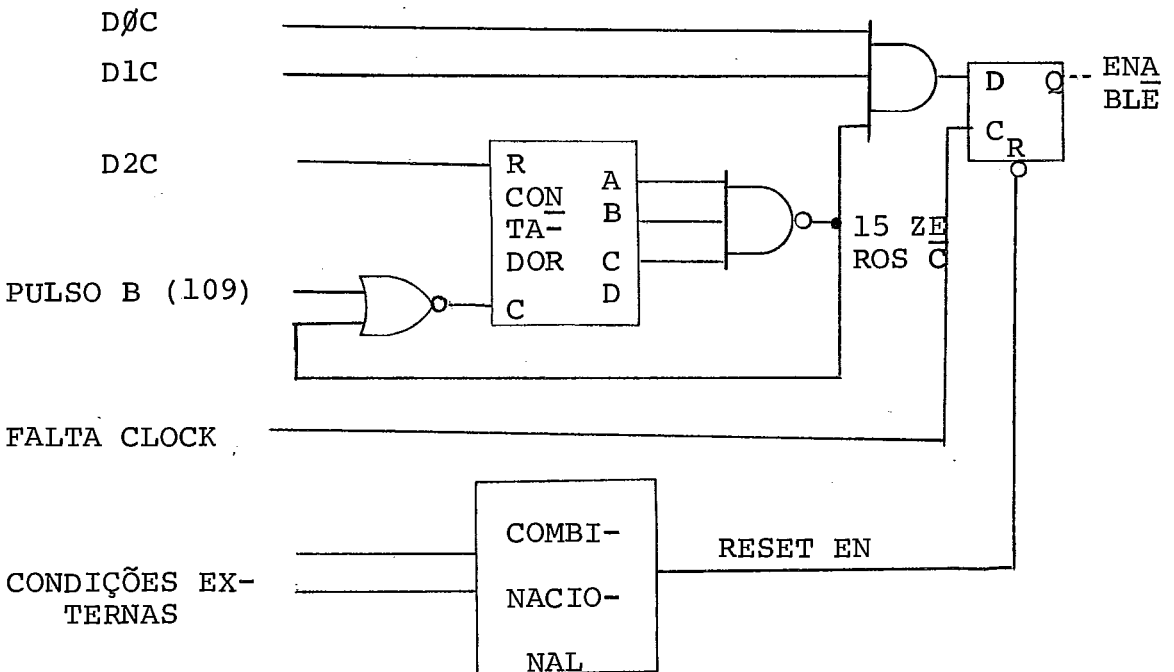
A ocorrência simultânea dos 3 fatos acima indica a possibilidade de ocorrência de uma ME que é concretizada quando ao final do byte, uma das marcas é decodificada: MEIV, MEIE ou MEID. Em qualquer um dos casos um FF é ativado e permanece neste estado durante todo o campo correspondente na trilha. Em qualquer hipótese somente uma das ME está ativa.

O ENABLE nada mais é do que uma linha que libera o Contador de Bits (registro de trabalho descrito posteriormente) da leitura em sincronismo com a

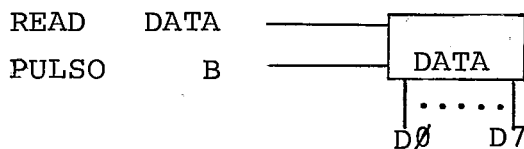
informação lida da trilha quando é ativado, e zera este contador e todos os FF's de detecção de ME quando é desativado, inibindo toda a lógica de leitura e mantendo o Controlador inalterado no estado em que se encontra. Esta condição deve sempre ocorrer quando OCUPADO = 0 ou ESTÁVEL = 0.

O funcionamento do circuito de ENABLE deve ser tal que ele é ativado na detecção de uma ME devendo no entanto ser desativado antes da ocorrência da ME seguinte pela linha RESET EN.

Levando-se em consideração as condições acima foi implementado o circuito abaixo:



As linhas Di correspondem às saídas do registro de DATA da forma mostrada abaixo:



O circuito acima ativa ENABLE se ocorrer sucessivamente:

- 1 - São detetados pelo menos 15 ZEROS;
- 2 - Ocorrem duas células de informação 1 e a célula seguinte não tem CLOCK.

A realimentação da linha 15 ZEROS permite que o Contador seja inibido até que $D2 = 1$. Estando 15 ZEROS, $DOUT0$ e $DOUT1$ em 1, a entrada D do FF também estará. Então se ocorre uma falta de CLOCK na célula seguinte a transição positiva do FALTA CLOCK coloca ENABLE em 1 antes que o pulso B desloque o registro de DATA passando D2 a 1 e zerando 15 ZEROS. Como o ENABLE em 1 inibe a operação de FALTA CLOCK, teremos este estado mantido até que ocorra um pulso em RESET EN, quando teremos $ENABLE = 0$. Este pulso ocorre para as condições abaixo:

- a - O controlador não está OCUPADO ou ESTAVEL = 0;
- b - Se após a ativação de ENABLE, ao final do byte uma Marca Especial não é detetada.

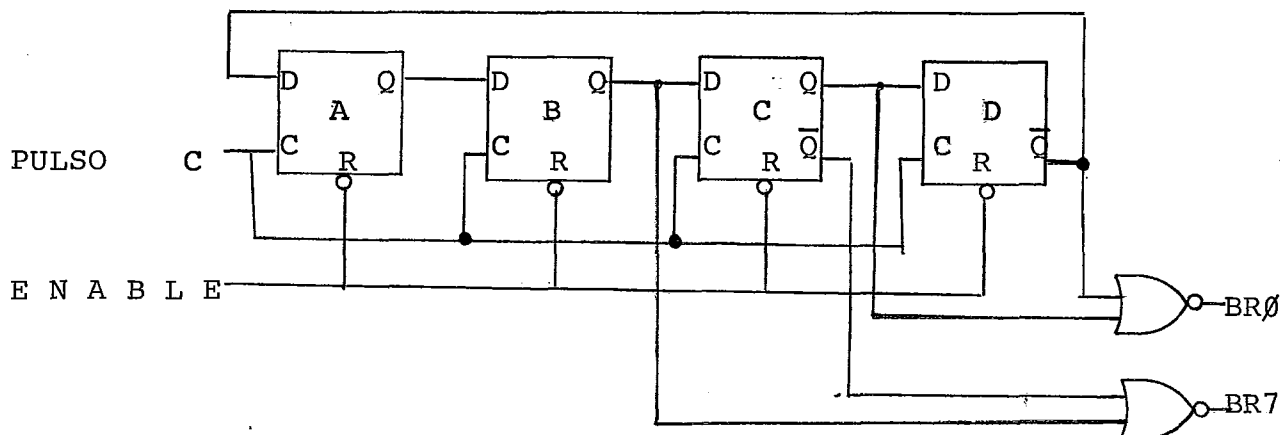
- c - Se $MEID = 1$, ao final do último byte de CRC com $\Delta FIM\ BYTE6$ (Vide Contador de Byte).
- d - Se $MEIE = 1$ ou $MEIV = 1$, e não é uma operação de LES, após o último byte de dado do campo de informação.
- e - Se $MEIE = 1$ ou $MEIV = 1$, e é uma operação de LES, após o último byte de CRC do campo de informação.

As condições acima são necessárias e suficientes para que o circuito desempenhe a sua função.

6.6 - Contador de Bits (CBITS) em Leitura

É um contador implementado em anel, liberado diretamente pelo ENABLE do circuito de Sincronismo e sua função é associar, a cada bit de um byte que chega, um determinado estado interno.

A lógica do Contador é mostrada abaixo:

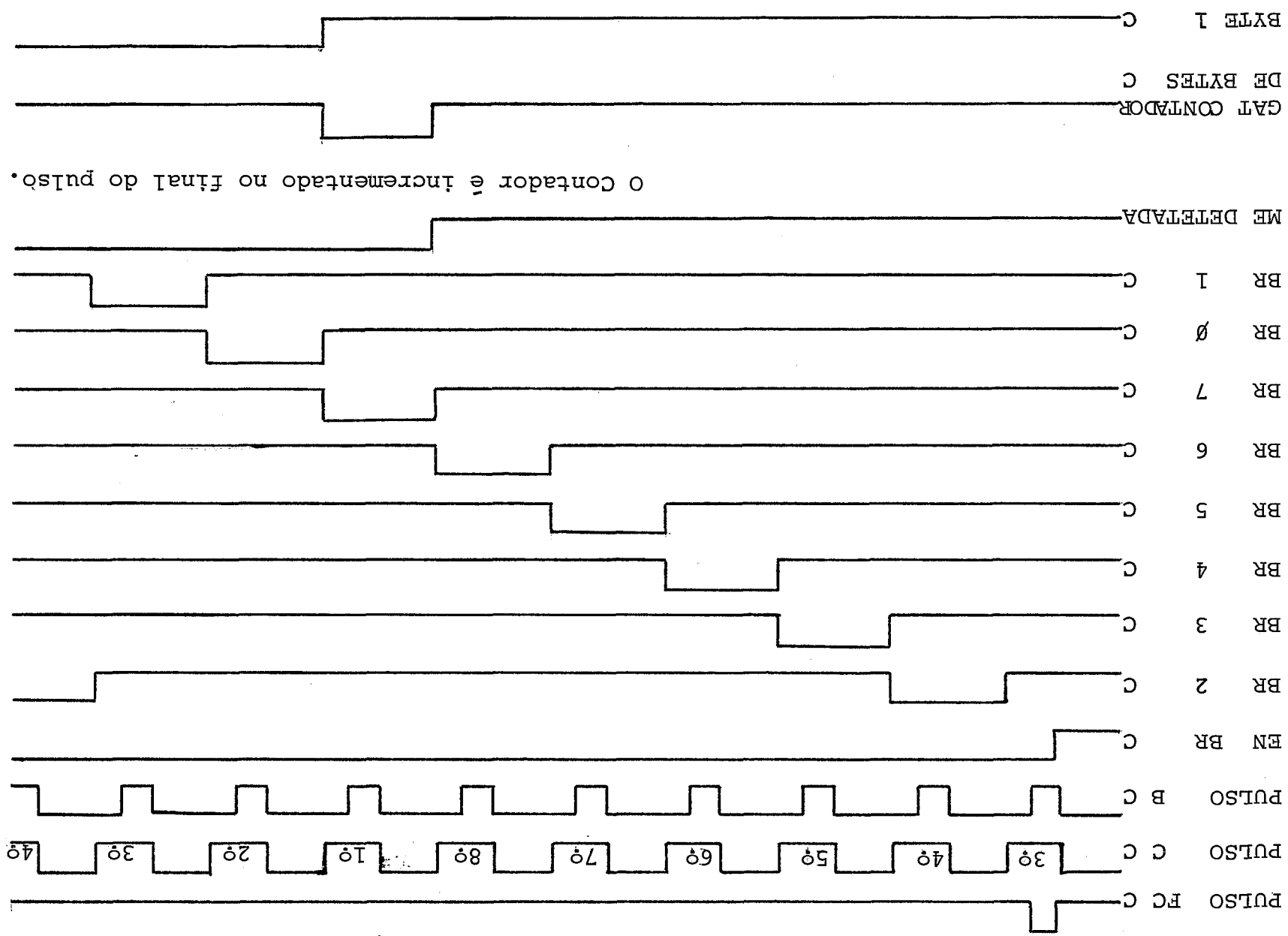


Chamando de B7 ao bit mais significativo de um byte e B0 ao bit menos significativo, D<0:7> às saídas do registro de DATA, podemos elaborar a tabela abaixo, supondo que o primeiro byte que chega é uma ME e as condições para a sua detecção foram satisfeitas. Convém lembrar que o ENABLE é detetado antes do deslocamento dos registros e portanto a liberação do Contador de Bits ocorre antes do armazenamento.

ESTADO	QA	QB	QC	QD	ENABLE	D0	D1	D2	D3	D4	D5	D6	D7
BR 1	0	0	0	0	0	0	0	0	0	0	0	0	0
BR 1	0	0	0	0	0	B7	0	0	0	0	0	0	0
BR 1	0	0	0	0	0	B6	B7	0	0	0	0	0	0
BR 2	1	0	0	0	1	B5	B6	B7	0	0	0	0	0
BR 3	1	1	0	0	1	B4	B5	B6	B7	0	0	0	0
BR 4	1	1	1	0	1	B3	B4	B5	B6	B7	0	0	0
BR 5	1	1	1	1	1	B2	B3	B4	B5	B6	B7	0	0
BR 6	0	1	1	1	1	B1	B2	B3	B4	B5	B6	B7	0
BR 7	0	0	1	1	1	B0	B1	B2	B3	B4	B5	B6	B7
BR 0	0	0	0	1	1	B7	B0	B1	B2	B3	B4	B5	B6
BR 1	0	0	0	0	1	B6	B7	B0	B1	B2	B3	B4	B5
BR 2	1	0	0	0	1	B5	B6	B7	B0	B1	B2	B3	B4

A ocorrência dos diversos estados é mostrada na FIG. 16. Como se pode observar, BR7 se torna ativo 400 ns depois da estabilização da informação nos registros de DATA e CLOCK, e sua transição positiva será utilizada como gatilho

FIG. 16 - Contador de Bits - D I A G R A M A D E T E M P O .



1º BYTE DEPOIS DA ME

ME

do Contador de Bytes durante o campo de identificação e como amostrador de informação decodificada nos circuitos de cheque e detecção da ME.

6.7 - Controle da Escrita

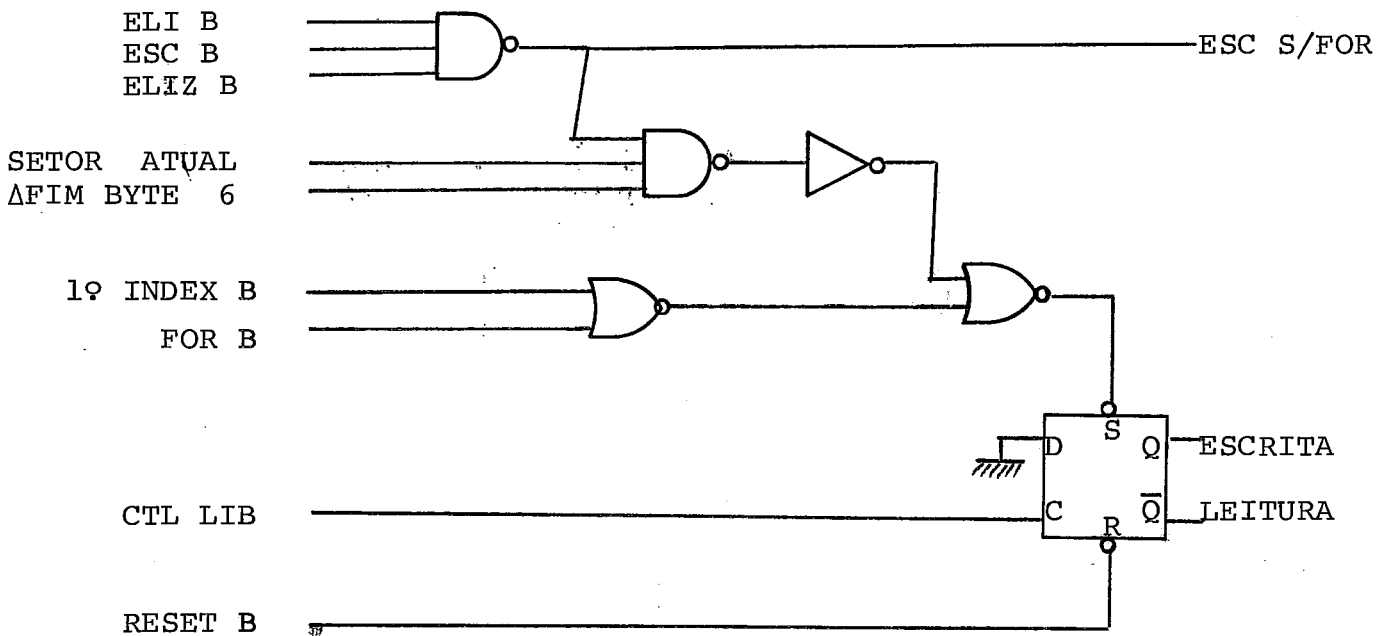
6.7.1 - ESTADO ESCRITA

Sabendo que uma transição do estado de Leitura para Escrita, e vice-versa, causa descontinuidades no campo magnético gravado, uma perda de sincronismo deve ser esperada. Para se garantir a resincronização do circuito Separador, faz-se com que a transição ocorra sempre no início de um preâmbulo de zeros. Isto implica em que uma instrução de escrita, excetuando a Formação, não se inicia simplesmente com a gravação da ME do campo de informação, mas, anteriormente, gravando, de uma só vez, inicialmente o preâmbulo de 17 ZEROS seguido da ME, informação e do CRC. Ao final do segundo byte do CRC, o Controlador retorna ao estado de Leitura, tendo, no preâmbulo de zeros que antecede o próximo campo de identificação, a garantia da detecção do setor seguinte.

As transições Leitura/Escrita para os diversos instruções são mostrados na tabela a seguir.

INST	LEIT — ESC	ESC — LEIT
ESC	SETOR ATUAL <u>E</u> Δ FIM BYTE <u>6E</u> ESC	FIM (MICROINSTRUÇÃO)
ELI	SETOR ATUAL <u>E</u> Δ FIM BYTE <u>6E</u> ELI	FIM
ELIZ	SETOR ATUAL <u>E</u> Δ FIM BYTE <u>6E</u> ELIZ	FIM
FOR	1º INDEX <u>E</u> FOR	P2º INDEX ou FIM

As condições de fim de Escrita atuam ambas no RESET e a determinação do estado do Controlador é implementada como abaixo:



É preciso salientar que, havendo sido detectado um erro de CRC na identificação do Setor Atual (ERID), a li

nha ESC INVALIDA = ESC S/FOR E ERID gera RESET e cancela a operação, colocando o Controlador no estado de Leitura;

O BORROW do Contador de Bytes atua no circuito escrita como indicador de fim de microinstrução. O CBYTE, como veremos detalhadamente a frente, é decrementado ao final de cada byte gravado. Como o estado normal do CBYTE é estar zerado ao início de qualquer microinstrução, este decremento gera um BORROW, a menos que a microinstrução, envolvendo a gravação de mais de um byte, armazene este valor durante a sua decodificação. Até a ocorrência do BORROW que desativa os FF's acionados pela microinstrução em andamento, eles permanecem ativos atuando diretamente no controle do fluxo de informação. No mínimo, um byte é gravado para cada microinstrução.

Como já visto na definição do software, qualquer operação de Escrita é um conjunto de microinstruções abaixo:

MICROINSTRUÇÃO	b7	b6	b5	b4	b3	b2	b1	b0
BYTE	0	0	0	0	0	0	0	1
WTR	0	0	0	0	0	0	1	0
FIM	0	0	0	0	0	1	0	0
CRC	0	0	0	0	1	0	0	0
LPZS	0	0	1	1	0	0	0	0
DADOS	0	0	0	1	0	0	0	1
WDD	1	1	0	0	0	0	0	0
WDAM	1	1	0	0	0	0	1	1
WID	1	1	0	0	0	1	1	0
WEX	1	1	1	0	0	1	0	0
LPZ B+1	1	0	0	0	0	0	0	0

A codificação das microinstruções acima não é casual, como se vê na explicação abaixo:

Se $b7 = \emptyset$, então:

b0 - indica que os bytes a serem gravados devem ser tratados como dado. A decodificação deste bit gera SEL DADO. SEL DADO atua no fluxo de dados selecionando o SAIBUS como entrada do registro de DATA, possibilitando aos bytes de informação buscados pelo canal, serem desserializados e gravados.

b1 - armazena no registro de DATA a trilha posicionada. Este bit gera um pulso no LOAD do registro de DATA e do fluxo vê-se que, estando SEL DADO e SELINST AM em zero, o valor armazenado será o proveniente do registro de trilha.

b2 - gera um pulso indicando fim de operação.

b3 - ativa SEL CRC. Esta linha inibe a entrada do registro gerador de CRC para que este gere o código cíclico corretamente, ao mesmo tempo que seleciona a saída deste registro de deslocamento como fonte para a gravação.

b4 - gera um pulso SEL TAS que carrega no CBYTE o tamanho do setor, gerando um loop. Veja o fluxo geral.

b5 - ativa o FF LOOP ZERO (LPZ) que acarreta duas atitudes: na sua transição positiva gera um LOAD no CBYTE e durante o tempo em que está ativo atua no CLEAR do registro de DATA gerando bytes ZEROS.

OBS: - No caso da microinstrução LPZS, os bits 4 e 5 atuam conjuntamente no LOAD do CBYTE mas o efeito é único e armazena TAS $\langle \emptyset:7 \rangle$.

Se $b7 = 1$, então:

b6 = 0 - indica uma operação de loop de zeros. Ativa o FF LPZ, só que agora o valor armazenado é $\text{BINST}\langle 0:7 \rangle$ como mostra o fluxo geral. Se a instrução é de FOR, $\text{BINST}\langle 0:7 \rangle = \text{SAIBUS}\langle 0:7 \rangle$. Caso contrário, $\text{BINST}\langle 0:7 \rangle = \text{BESC}\langle 0:7 \rangle$ (vem diretamente da matriz de diodos). Em qualquer dos casos o valor armazenado é $B = b5b4b3b2b1b0$.

b6 = 1 - esta condição ativa SEL INST AM e grava a ME através de dois efeitos: gera um LOAD no registro de DATA armazenando o valor $D\langle 7:0 \rangle = (1,1,1,1,1,b2,b1,b0)$ e gera um LOAD no registro de CLOCK (cujo valor é guardado em complemento) armazenando $C\langle 7:\emptyset \rangle = (0,0,1,b5,1,0,0,0)$. Neste caso b5 evita a necessidade de decodificação da MEIX.

Enunciamos abaixo o conjunto de microinstruções que compõem cada uma das instruções ESC, EL e ELIZ.

INSTRUÇÃO	ESC	ELI	ELIZ
	LPZ 17	LPZ 17	LPZ 17
	WDAM	WDD	WDD
	DADOS	DADOS	LPZS
	CRC	CRC	CRC
	CRC	CRC	CRC
	FIM	FIM	FIM

A tabela acima é a base da montagem da matriz de diodos.

6.7.2 - RELÓGIO E CODIFICADOR

Uma vez detetado o estado Escrita as ações sb sequentes passam a ser sincronizadas por um re relógio operando na frequência $f_0 = 4 \text{ MHz}$. Um codificador acoplado diretamente a ele permite gerar os pulsos fundamentais para o Controlador.

Esquema:

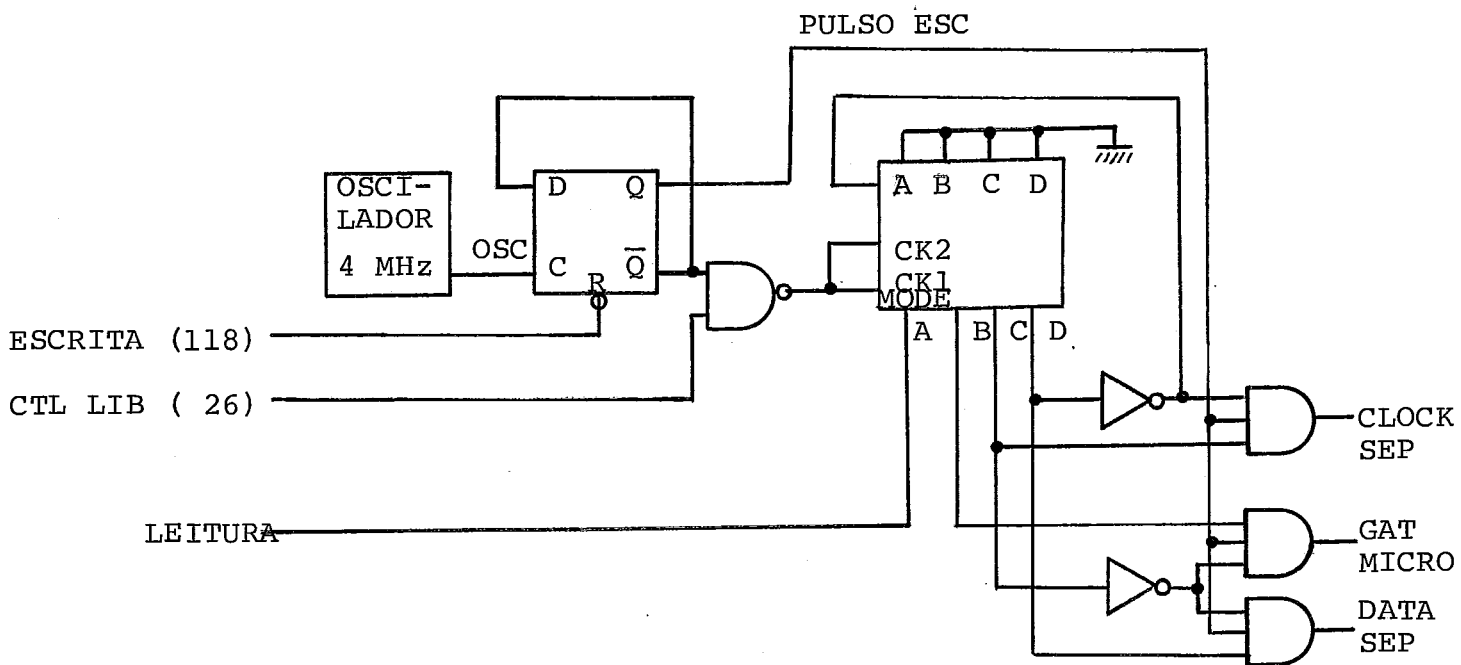
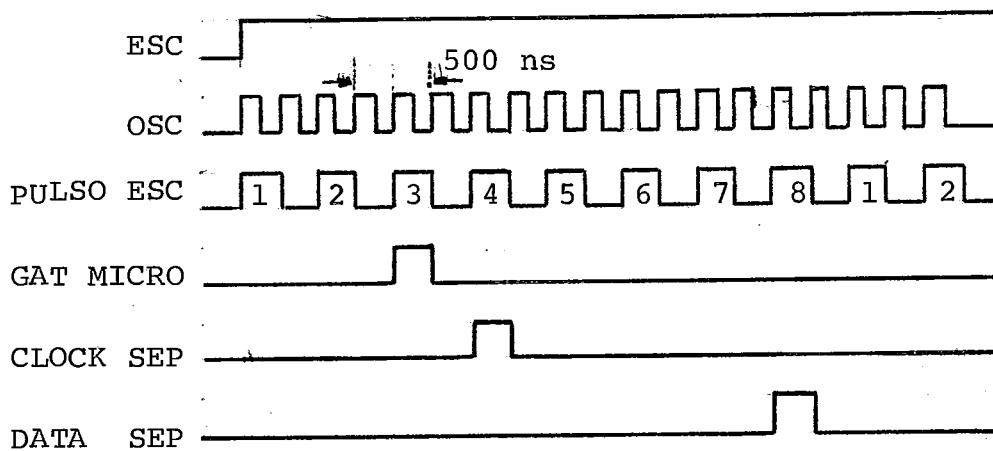


Diagrama de Tempo:



Os pulsos CLOCK SEP e DATA SEP gerarão a linha WRITE DATA. GAT MICRO será usado como sincronismo no circuito de decodificação de microinstruções.

6.7.3 - CONTADOR DE BITS NA ESCRITA

Gera PROX BYTE ESC, capaz de indicar o fim de um byte, que será a detecção da ocorrência de 8 pulsos de DATA SEP. Cabe aqui observar, como informação adicional, que tanto o registro de DATA como o de CLOCK, durante a escrita, são deslocados ao final do pulso DATA SEP. PROX BYTE ESC decrementa o CBYTE e a ocorrência de um BORROW no Contador provoca um RESET ESC que, na realimentação, zera o Contador de Bits.

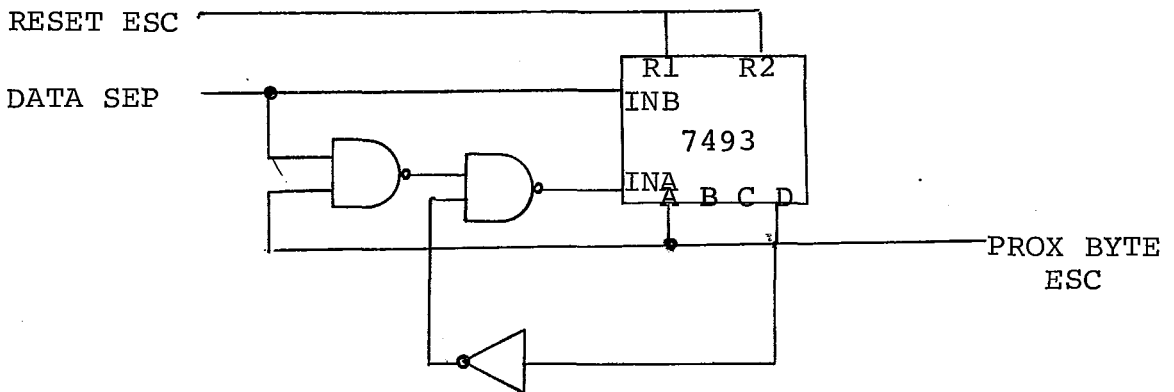
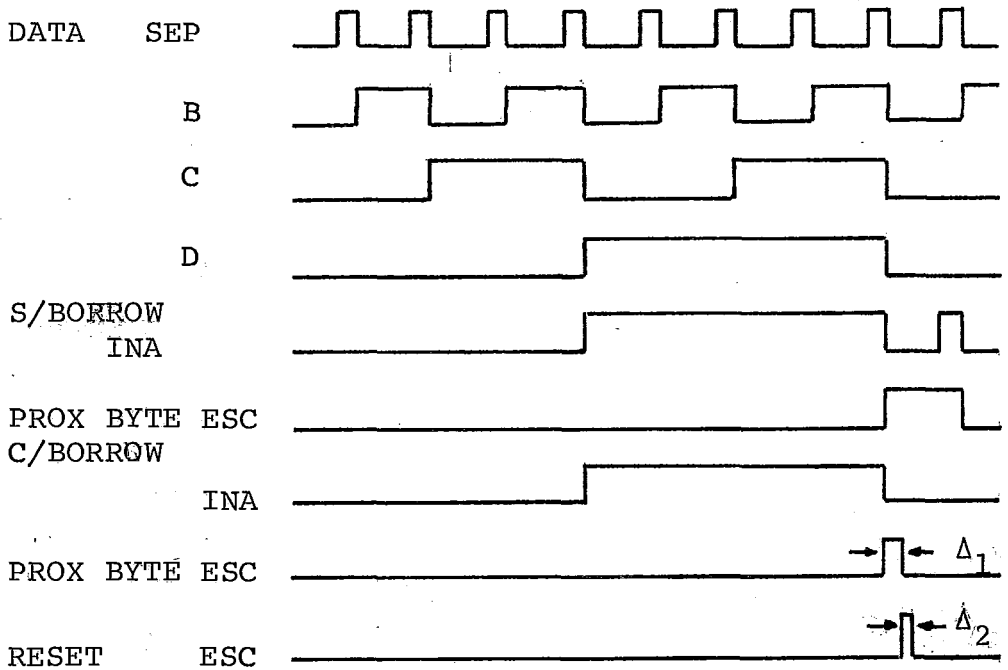
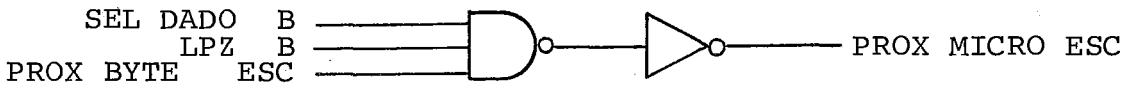


Diagrama de Tempo:



OBS: - As larguras Δ_1 e Δ_2 dependem dos atrasos da lógica interna. Certos atrasos foram inseridos propositadamente no circuito para garantir uma largura razoável a estes pulsos.

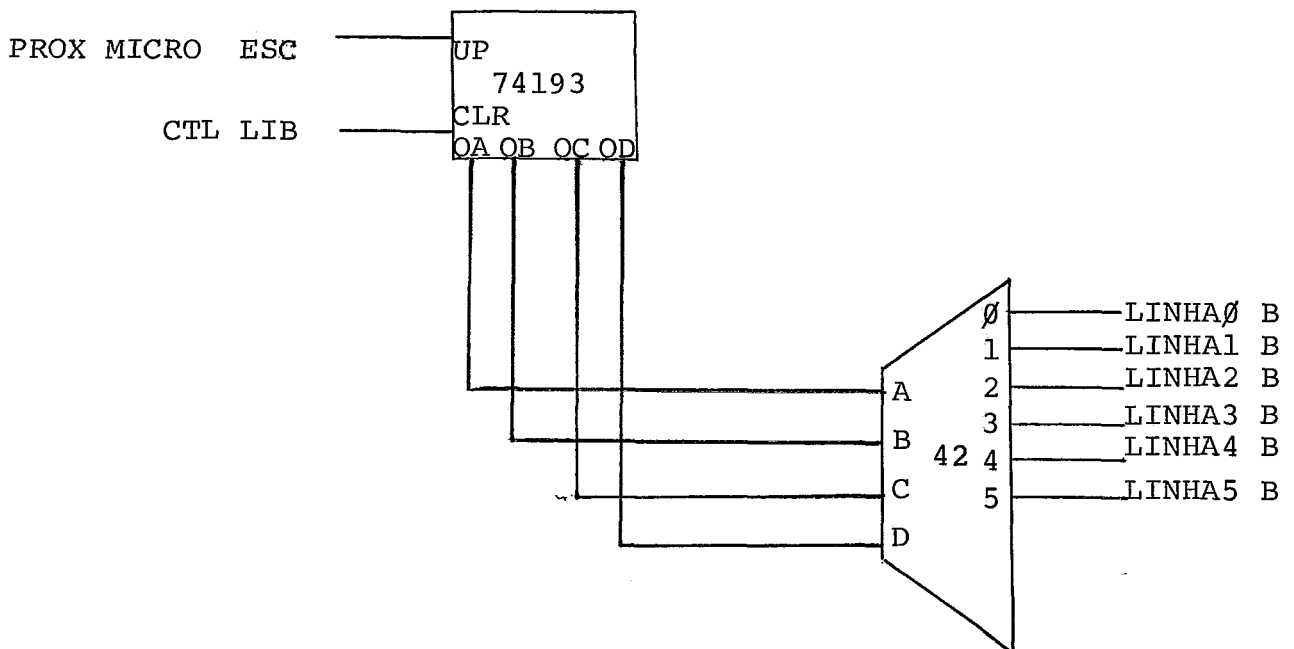
A ocorrência do PROX BYTE ESC sempre significará o pedido para uma nova microinstrução exceto quando estivermos num loop de zeros ou transferindo dados. Geramos então o pulso de indicação de busca de microinstrução da forma simples a seguir.



6.7.4 - SEQUENCIADOR E MATRIZ DE DIODOS

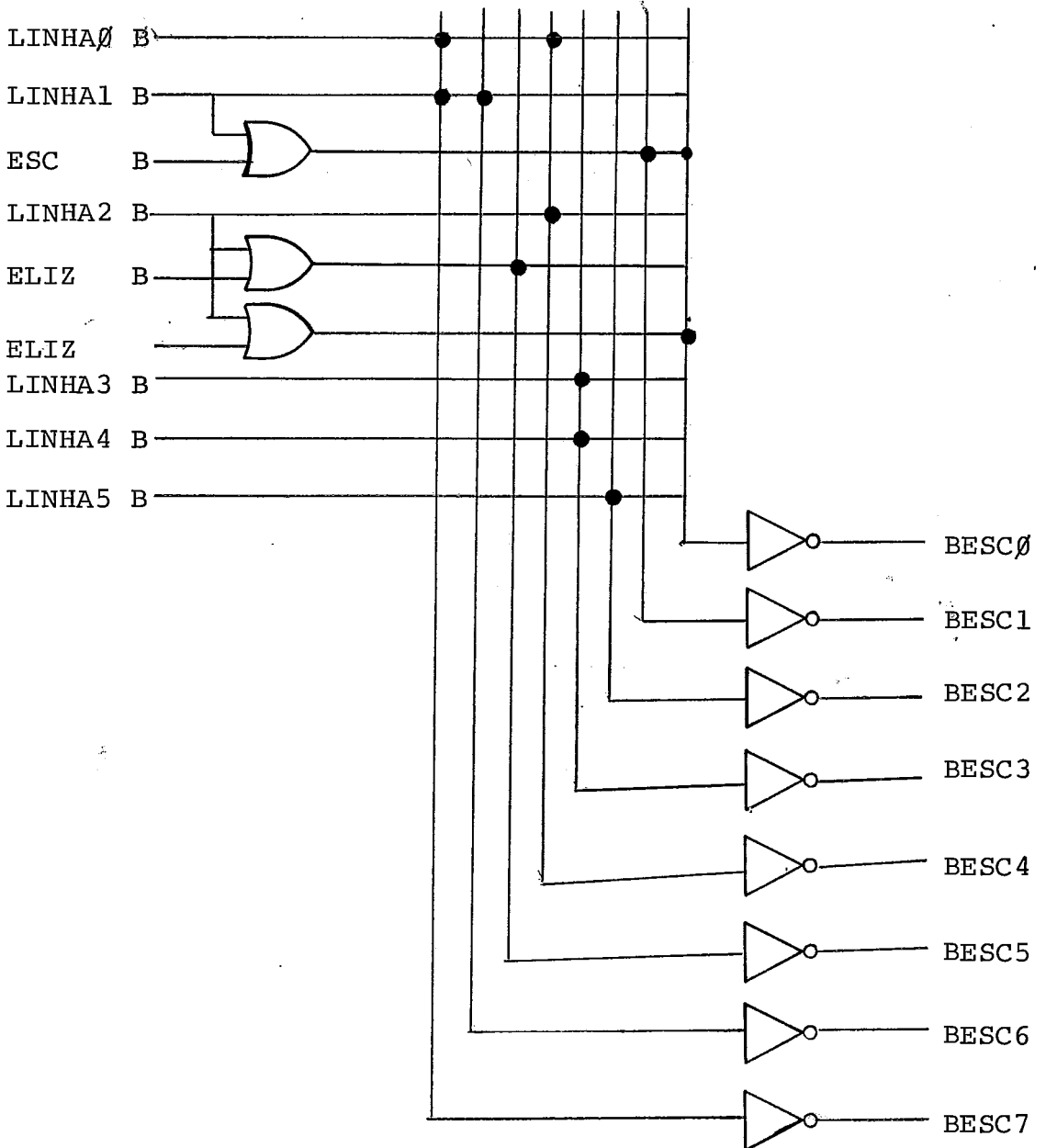
O Sequenciador nada mais faz do que endereçar sequencialmente, a cada PROX MICRO ESC, a Matriz de Diodos que gera, a cada instrução ESC, ELI ou ELIZ, a seqüência de microinstruções definida. O sequenciador sempre se inicia do endereço 0 pois é zerado a cada CTL LIB. A formação da micro correta para cada caso se deve à utilização de linhas adicionais condicionadas a decodificação das instruções.

Esquema: (Sequenciador)



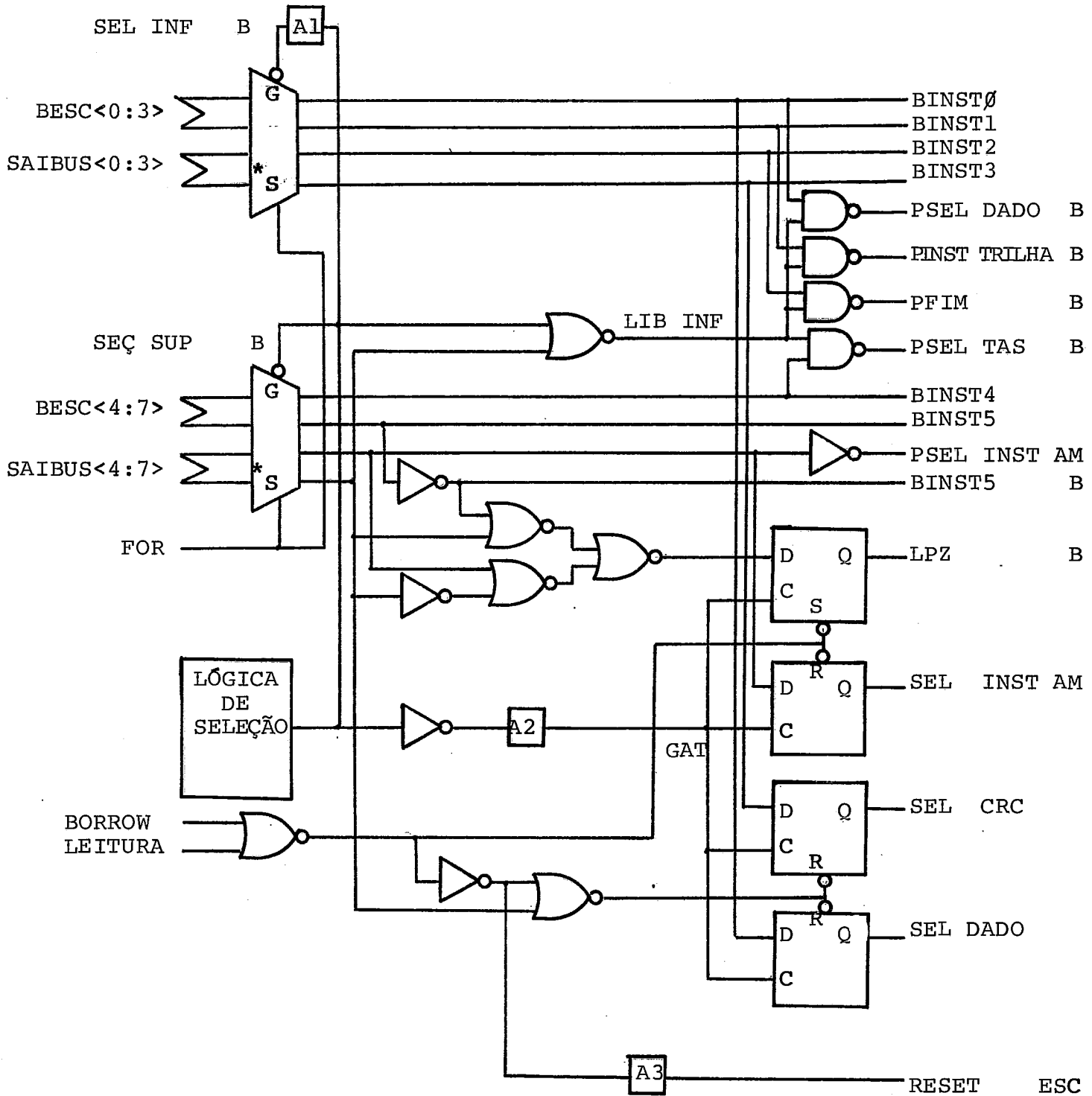
Esquema: (Matriz de Diodos)

OBS: - A cada ponto negro corresponde um diodo. Se selecionada a linha deste, ele leva a coluna correspondente a 0, caso contrário ela permanece em 1.



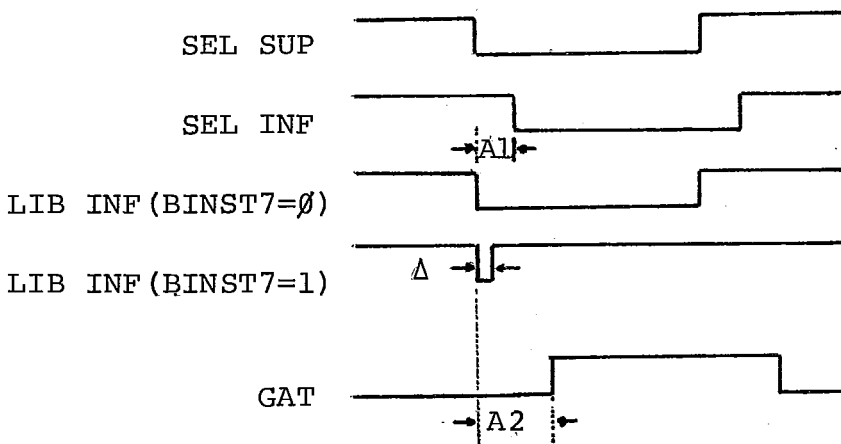
6.7.5 - DECODIFICADOR DE MICROINSTRUÇÕES

Esquema:



Pode-se observar primeiramente que, estando o Controlador em Leitura, os FF's de decodificação estão zerados. Isto garante estado inicial para Escrita.

A decodificação em si é bastante clara. Os atrasos A1 e A2 devem possibilitar as formas de onda abaixo:

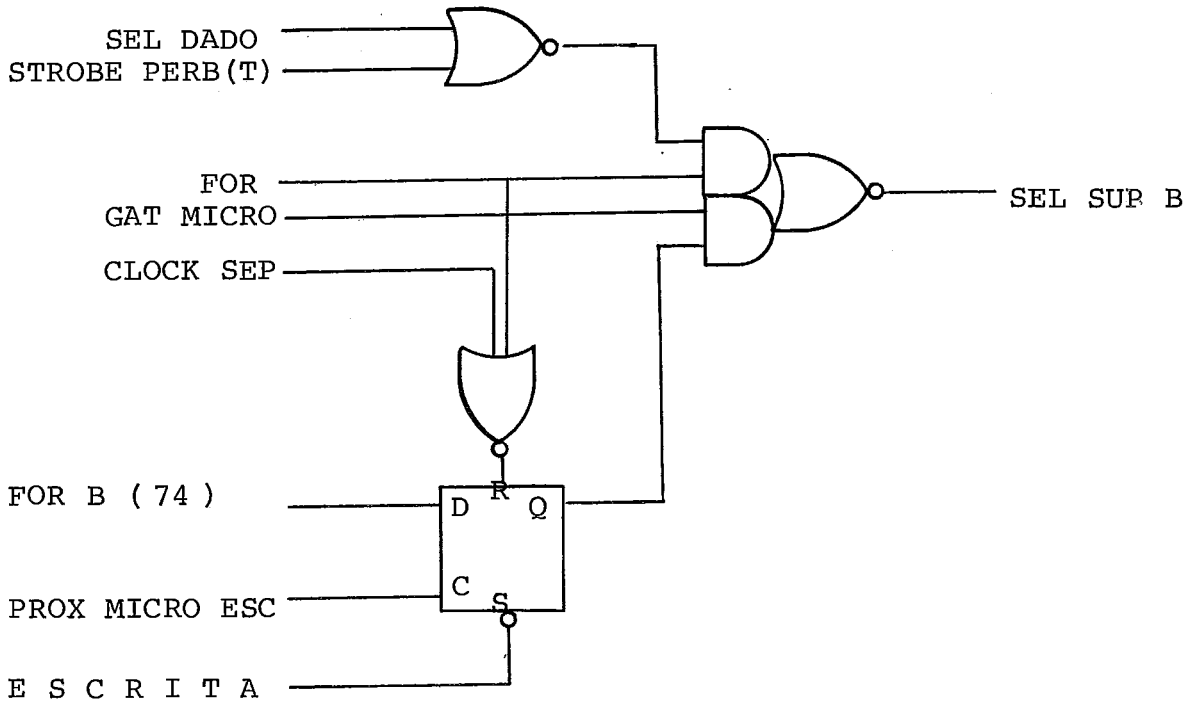


O pulso ~~no~~ em LIB INF quando BINST7 = 1 de largura Δ é proveniente do atraso do multiplexador. O atraso A1 visa exatamente liberar os bits inferiores pouco mais tarde, quando LIB INF está já estabilizada. O atraso A2 permite que as linhas inferiores sejam amostradas depois de sua estabilização. Os pulsos nas portas E possuem uma largura de SEL SUP - A1.

O atraso A3 colocado na geração do RESET ESC é exatamente o que garante uma determinada largura para o pulso PROX MICRO ESC.

A ordem de grandeza dos atrasos é: A1 = 40 ns;
A2 = 80 ns; e A3 = 40 ns.

A lógica de seleção encarregada de gerar SEL SUP é mostrada abaixo:



A função do FF é permitir que somente o primeiro pulso GAT MICRO de um novo byte decodifique a microinstrução. Note que CLOCK SEP gera o FF e ele só é liberado na ocorrência de PROX MICRO ESC, se não estivermos em Formatação. Estando em Formatação, o FF é mantido zerado impedindo a ação do GAT MICRO. Para possibilitar a decodificação da primeira microinstrução de uma operação ESC, ELI ou ELIZ, garante-se que o FF está inicialmente em 1 pela linha Escrita.

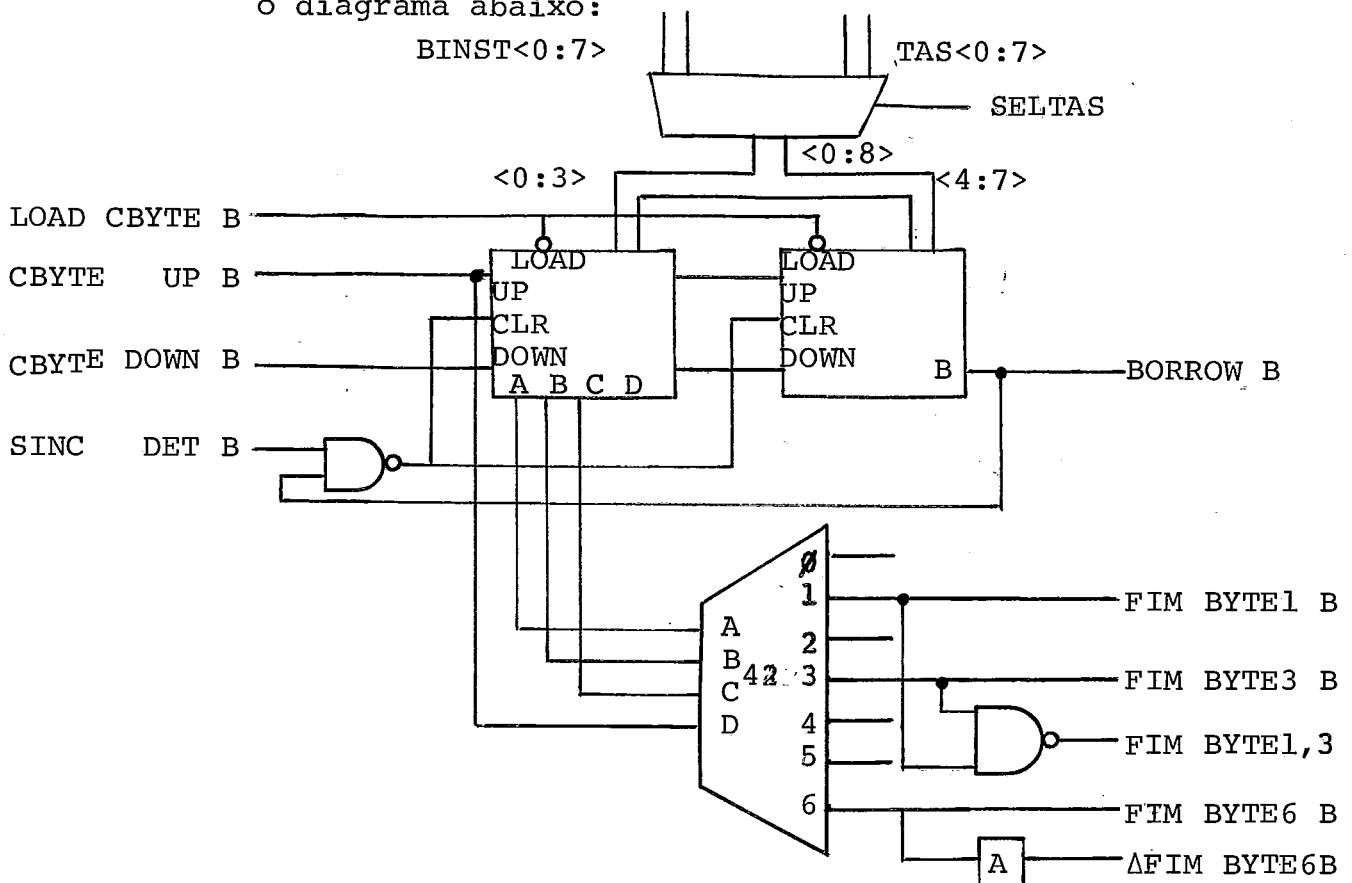
A geração do SEL SUP na Formatação é feita com o próprio protocolo de transferência através do STROBE

PER, com a única condição de não estarmos numa transferência de dados. Durante a Formatação instruções e dados são transferidos da memória, via protocolo do Canal.

A atuação das várias linhas geradas pelos circuitos de escrita é direta e a descrição se acha em cada circuito particular, como nos registros de trabalho ou no pedido de transferência.

6.8 - Contador de Bytes (CBYTES)

É o elemento fundamental para o sincronismo de Leitura e Escrita. Ele é implementado com 2 CI 74193 (BINARY COUNTER UP AND DOWN WITH LOAD AND CLEAR), como mostra o diagrama abaixo:

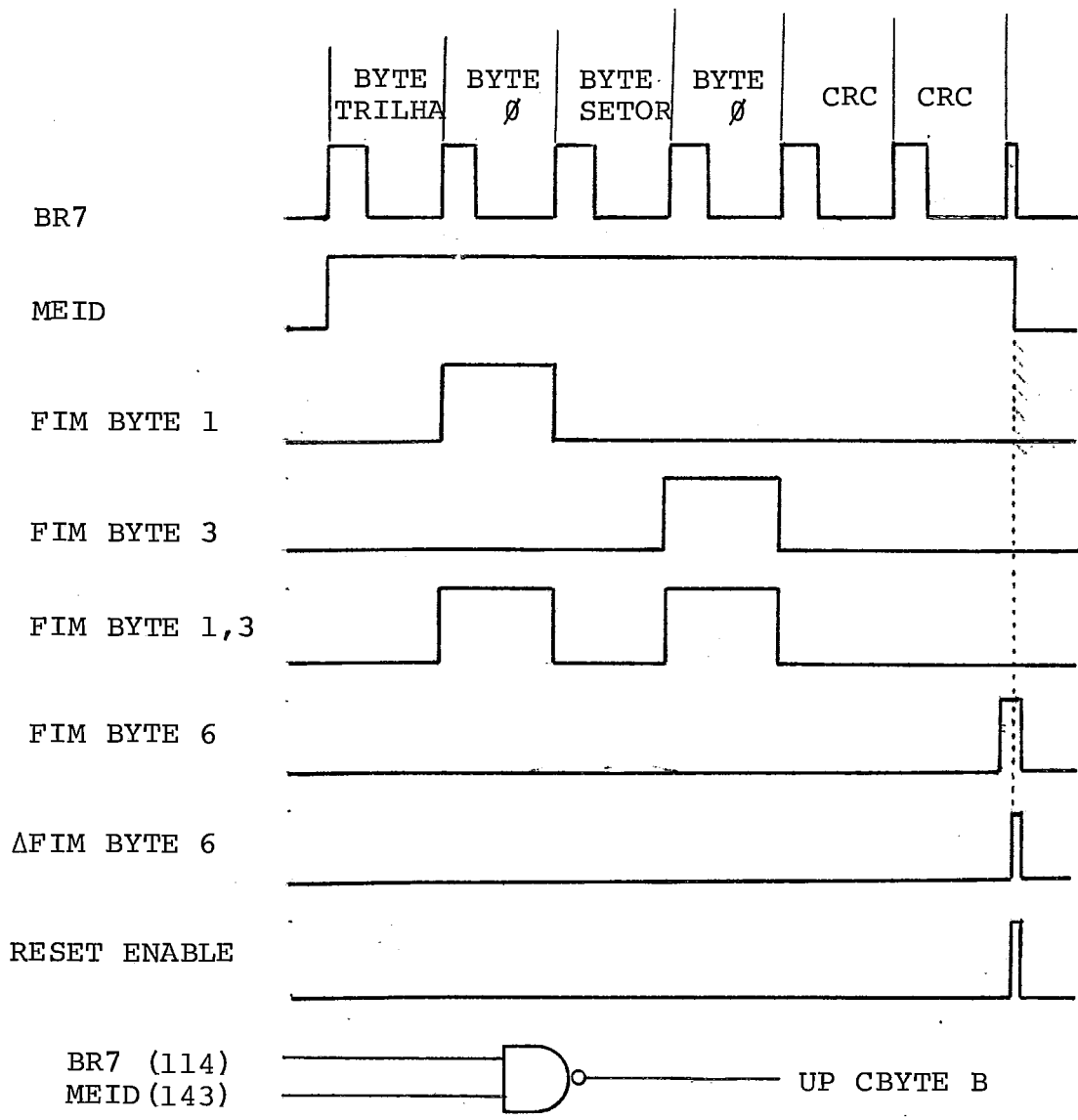


Uma análise imediata do circuito implementado mostra que, devido a realimentação do BORROW no CLR, o Contador nunca será decrementado de /00 para /FF, gerando sempre BORROW a cada pulso no CBYTE DOWN nas condições acima. Esta característica do Contador é fundamental para o seu desempenho como elemento de controle na escrita e na leitura.

Se o CBYTE é usado incrementado ou decrementado, depende do tipo de campo em que se opere. Descreveremos abaixo cada uma das situações.

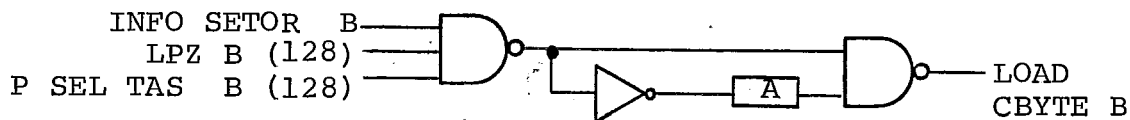
CAMPO DE IDENTIFICAÇÃO

Durante o estado de Leitura, detetada uma MEID, o CBYTE passa a ser incrementado a cada BR7 do CBITS. Nestas condições são decodificadas as linhas FIM BYTE1, FIM BYTE3, FIM BYTE1,3, FIM BYTE6 e ΔFIM BYTE6 (atrasada ~40 ns em relação a anterior) que permitem a sincronização dos circuitos de cheque e deteção de erro de CRC, como veremos adiante. Em particular, ΔFIM BYTE6 atua no reset do ENABLE (vide SINCRONISMO).

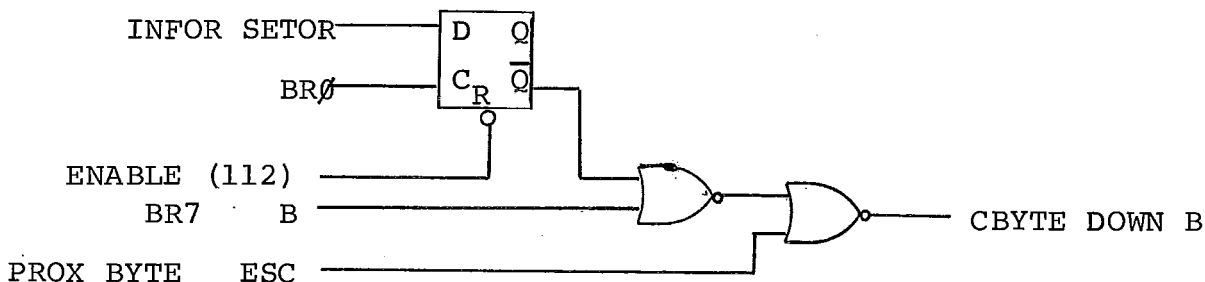


CAMPO DE INFORMAÇÃO

Em leitura, a detecção de MEIV ou MEIE ativa INFO SETOR = MEIV + MEIE e um circuito monoestável gera um LOAD no CBYTE, armazenando neste o TAS. Durante a Escrita, o LOAD é gerado com a decodificação de LPZ ou PSEL TAS. O circuito completo que gera o LOAD, é mostrado abaixo:



O CBYTE é decrementado na escrita a cada PROX BYTE ESC e na leitura a cada BR7. Veja o esquema abaixo:



O FFD no circuito garante que o Contador de Byte, logo após a detecção de Info Setor, não é gatilhado pelo BR7 correspondente ao final do byte de Marca Especial, ocorrendo o primeiro pulso em CBYTE DOWN somente ao final do primeiro byte de informação.

Em leitura, o tratamento que será dado ao BORROW depende diretamente da linha INFO PRONTA gerada da forma abaixo:

$$\text{INFO PRONTA} = \text{TRANSF SETOR} \underline{\text{E}} \text{INFO SETOR} \underline{\text{E}} \text{SETOR ATUAL}$$

$$\text{TRANSF SETOR} = \text{TCI} + \text{LES}$$

SETOR ATUAL - é ativado pelo circuito de cheque quando o byte do setor coincidir com o setor armazenado.

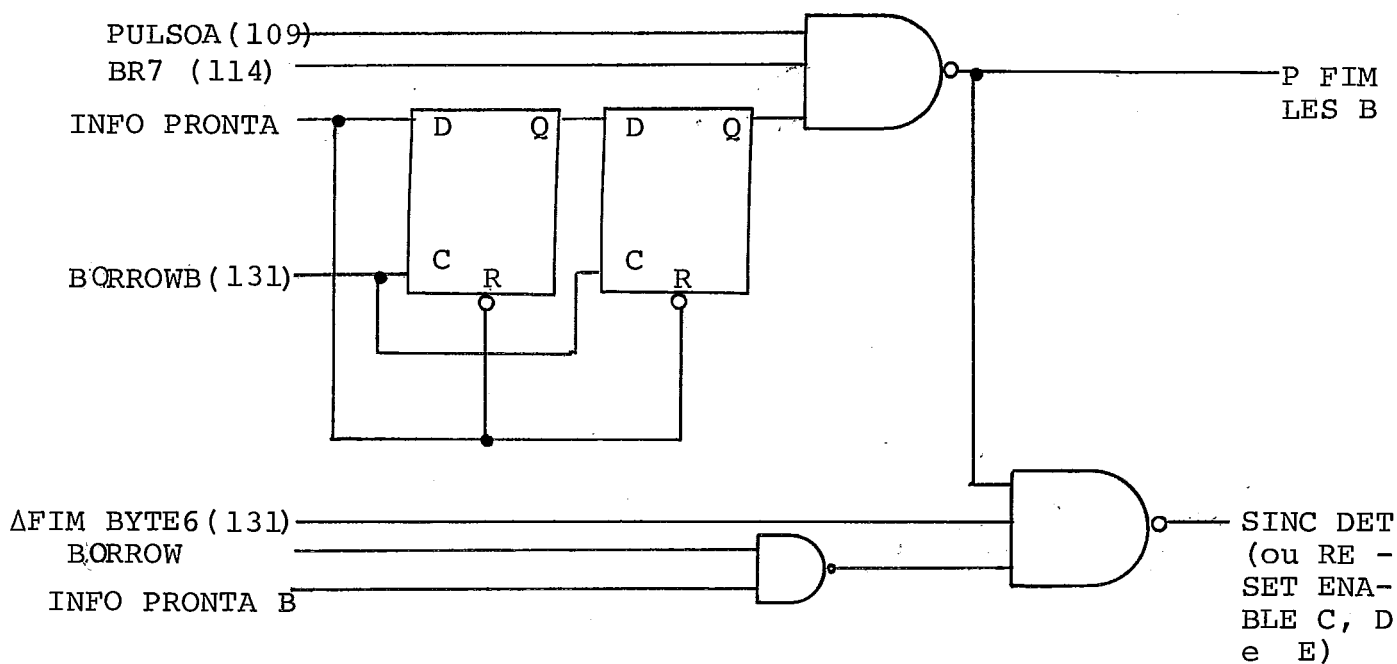
O BORROW será responsável pelo fim de operação em leitura (PFIM LES) e também na desativação do ENABLE, após os campos de informação.

O RESET ENABLE é então gerado de duas formas. Se

INFO PRONTA = 0 então o primeiro BORROW zera o ENABLE. Caso contrário, o RESET ENABLE só acontece ao terceiro BORROW, o que corresponde ao fim do 2º byte do CRC e ao fim de operação de uma instrução LES. Estas condições correspondem, respectivamente, aos itens D e E na descrição do RESET ENABLE, no capítulo SINCRONISMO.

Δ FIM BYTE6 mostrado no circuito abaixo, corresponde à condição C do mesmo capítulo.

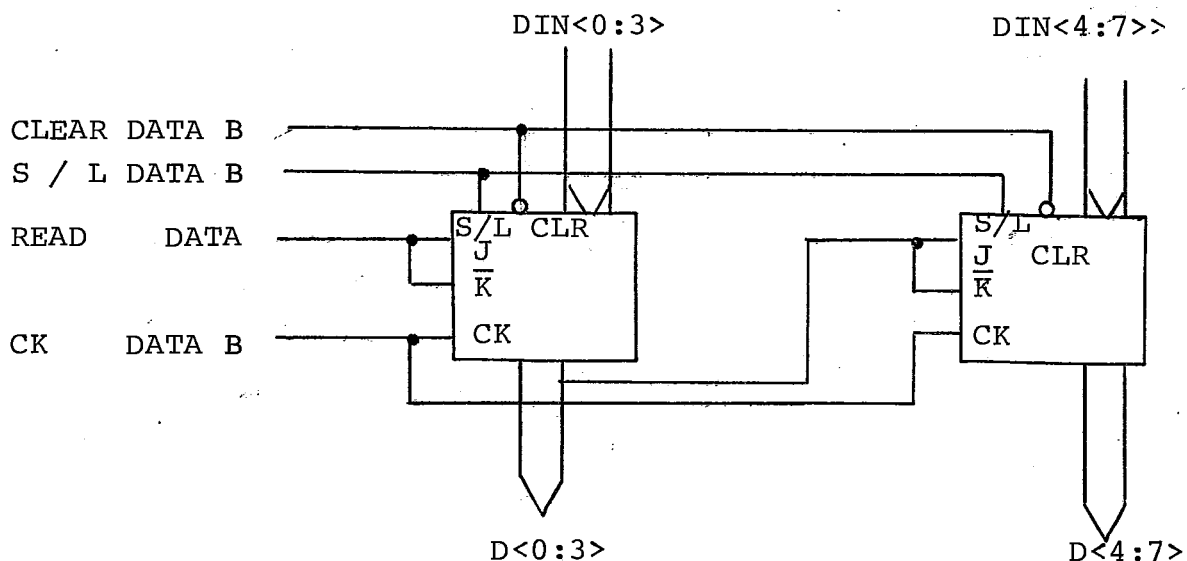
A lógica implementada é a abaixo:



6.9 - Registro de Informação

É a conexão entre o fluxo de informação série e o fluxo paralelo. Durante o estado de Leitura executa a operação de desserialização dos bits vindos da unidade, tornando possível a detecção dos campos e a transferência da informação. Na Escrita, o registro executa a operação inversa de serialização.

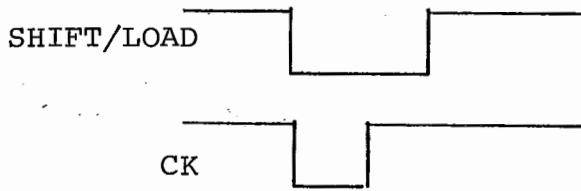
É implementado com CI's 74195 (shift-register) que permitem operações de deslocamento, armazenamento e CLEAR.



A linha READ DATA porém do circuito de Sincronismo. A origem de DIN<0:7> pode ser bem observada no diagrama do fluxo geral e a seleção dos caminhos já foi explicada na descrição da Escrita.

Vale aqui um parentesis para comentar a lógica de funcionamento do registro, que virá justificar uma apa-

rente complicação desnecessária na geração das linhas de controle. Quando queremos armazenar um determinado valor no registro não é suficiente apenas pulsar a linha SHIFT/LOAD, mas gerar um pulso adicional CK, cuja transição positiva carrega o valor existente nas linhas da entrada paralela.

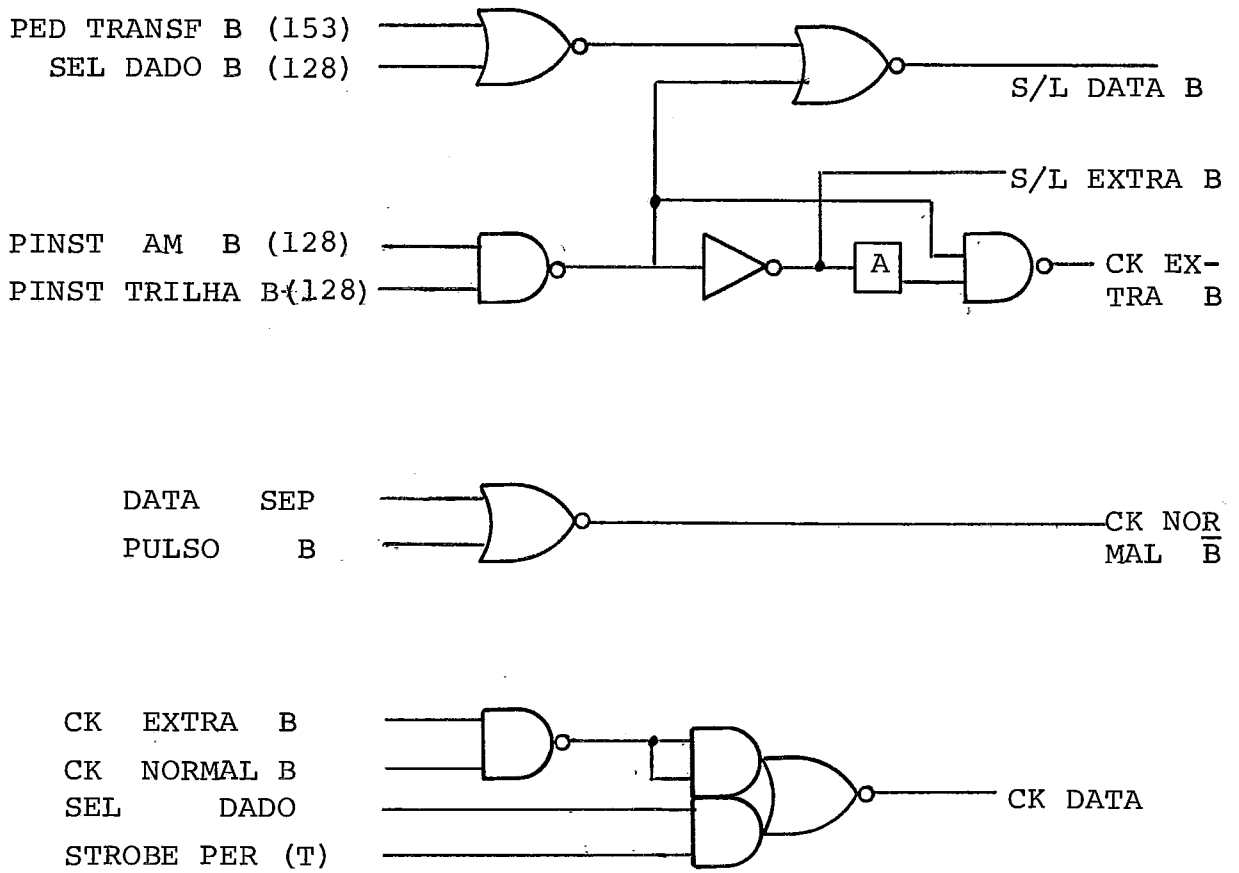


A estes CK adicionais denominaremos CK EXTRA.

Durante a leitura, o registro funciona apenas com deslocamento sincronizado por PULSO B (vide SINCRONISMO) mantendo-se $S/L = 1$ e $CLEAR = 1$.

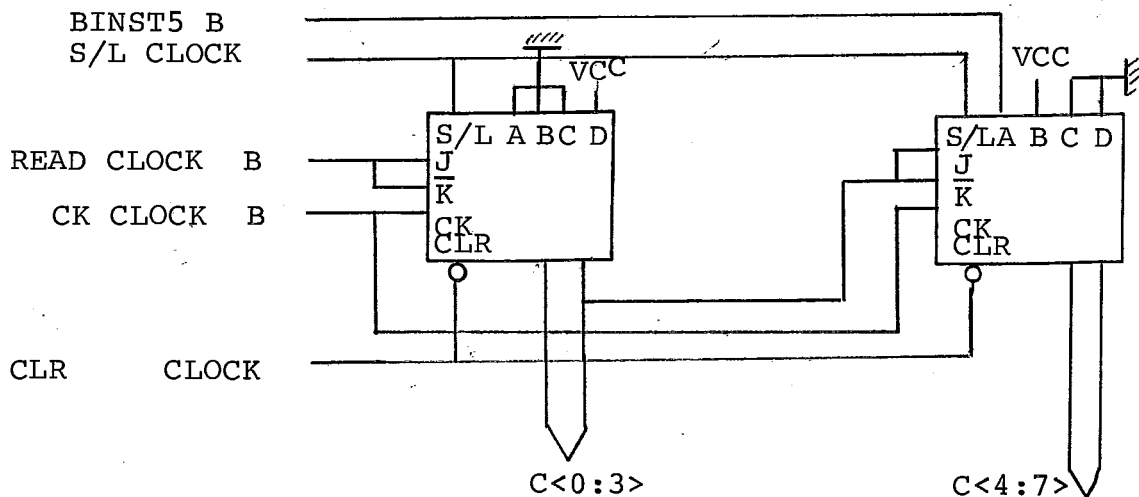
Durante a Escrita o registro deve ser armazenado com $DIN\langle 0:7 \rangle$ na ocorrência de PINST AM ou PINSTRILHA e ainda se, no caso de instrução de Formatação, estarmos numa transferência de dados. Neste caso o próprio STROBE PER é utilizado como CK EXTRA, sendo o Pedido de Transferência utilizado como S/L. Os deslocamentos normais devem ocorrer a cada DATA SEP.

Esquema:

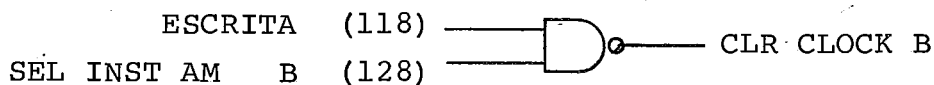


6.10 - Registro de CLOCK

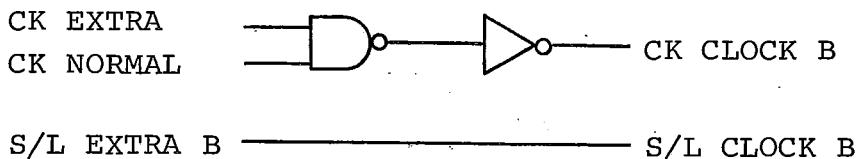
O registro de CLOCK é implementado de maneira análoga ao registro de DATA. Para facilitar a implementação do hardware, o registro guarda os valores em complemento.



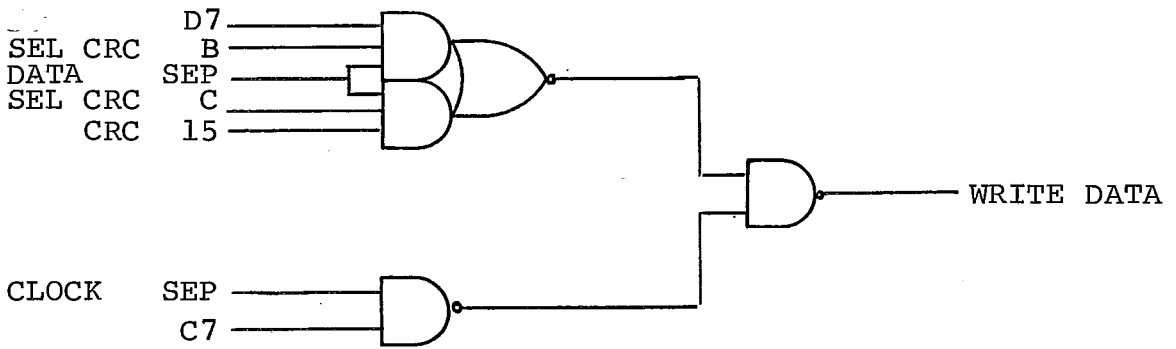
Geração das linhas de controle:



O único caso em que o registro não deve ser zerado é quando eu gravo uma ME. Neste caso SEL INST AM força CLR CLOCK = 1. Observe que quando estou em Leitura CLR CLOCK = 1, forçado por Escrita = 0.



Aproveitando o fato de estarmos envolvidos no tratamento da informação, vamos mostrar abaixo como se imple - mentar a linha WRITE DATA responsável pela gravação no Diskette.

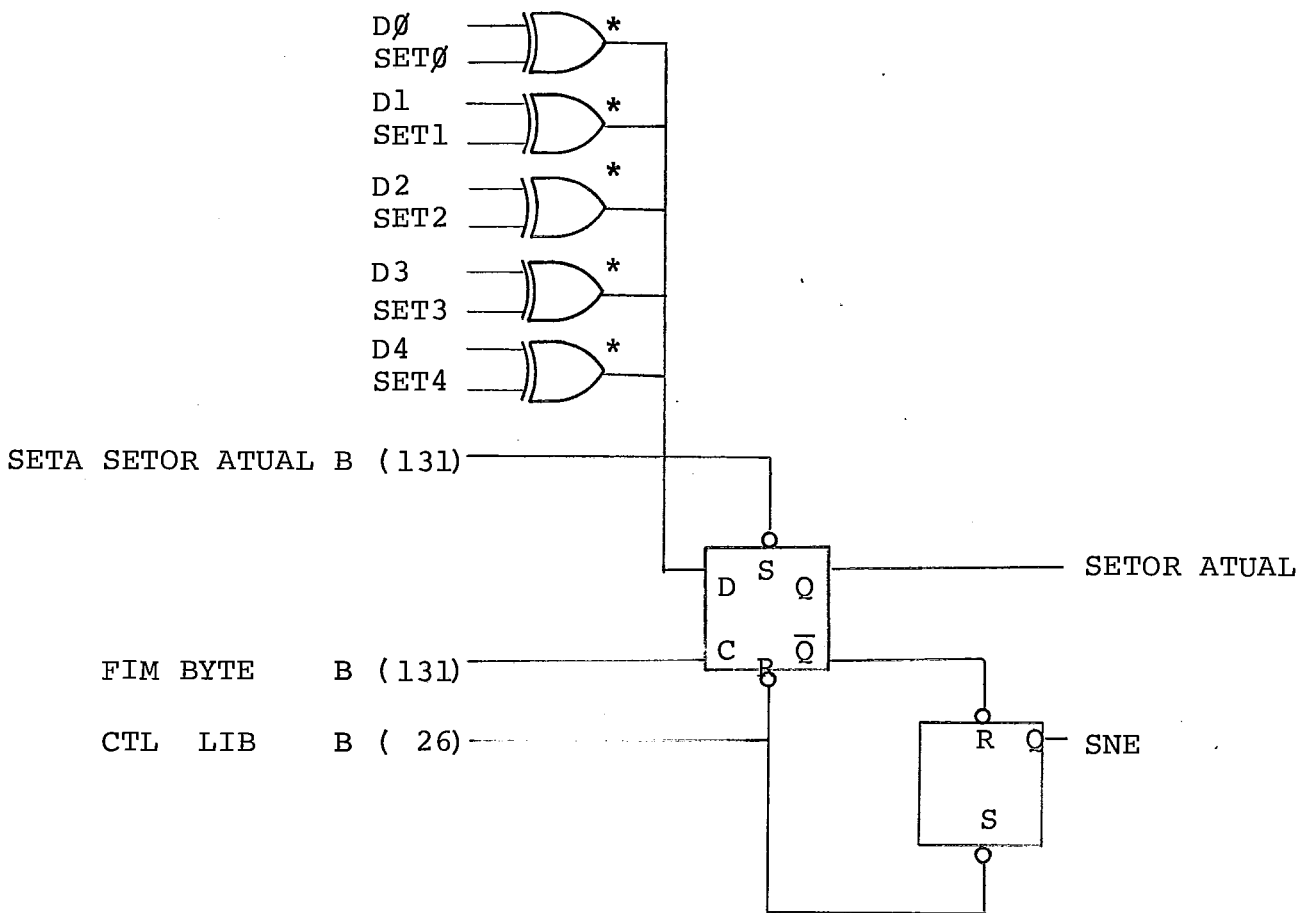


A parte superior é responsável pela geração dos pulsos de informação, selecionando por intermédio de SEL CRC ou a saída D7 do registro de DATA ou a saída CRC15 do registro de CRC, condicionados com DATA SEP.

A porta E abaixo simplesmente gera o pulso de CLOCK em sincronismo com CLOCK SEP, condicionado a C7.

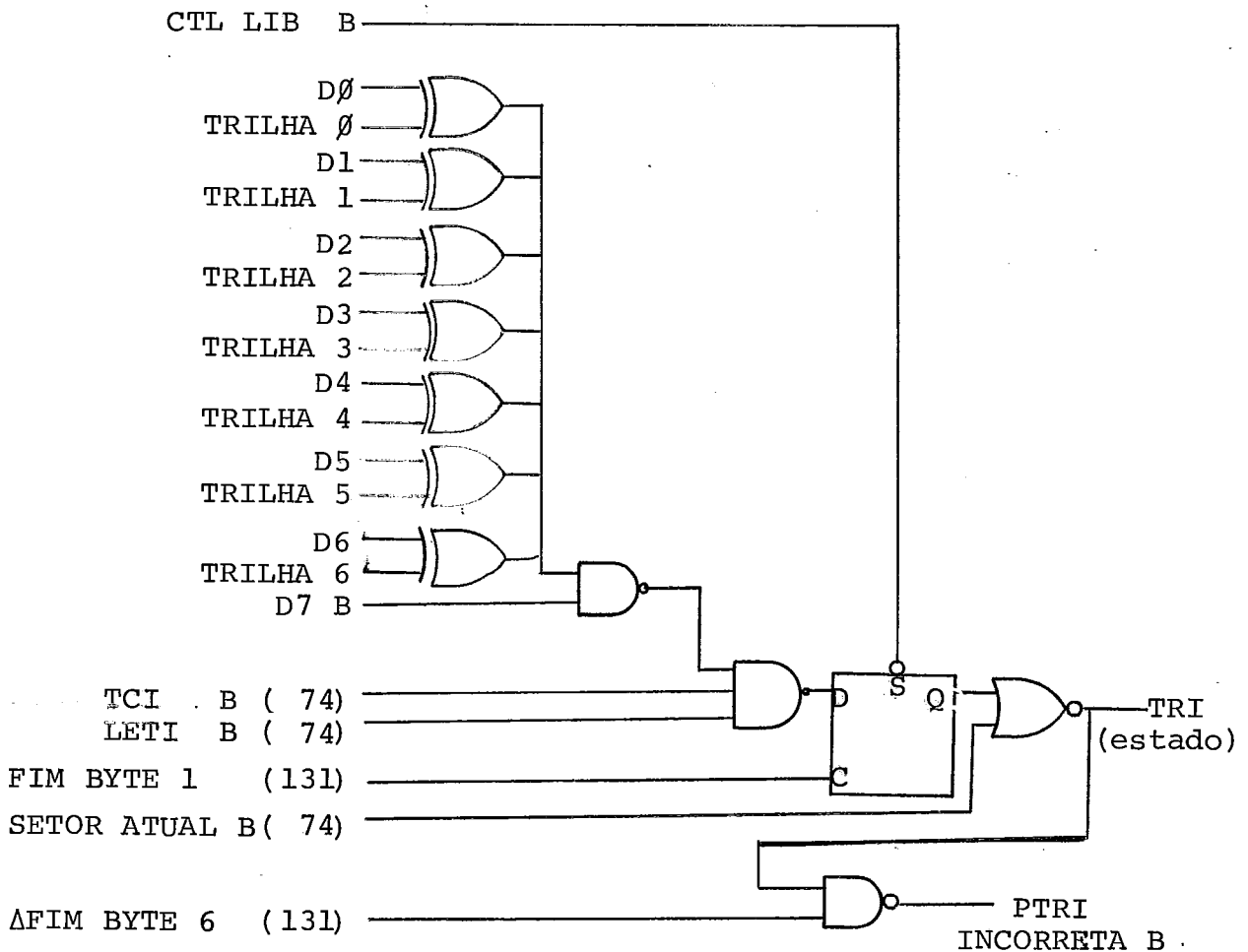
6.11 - Cheque de Setor

A função do circuito é comparar o byte do setor do campo de identificação com o setor armazenado, ativando a linha SETOR ATUAL quando a comparação é satisfeita. Uma vez ativada a linha SETOR ATUAL ela permanece neste estado até a decodificação da instrução seguinte e o estado SETOR NÃO ENCONTRADO (SNE), ativado inicialmente pelo CTL LIB, é desativado.



6.12 - Cheque de Trilha

A função do circuito é conferir o byte de trilha de cada campo de identificação (exceto para inst LETI) com o valor da trilha armazenado. Se a comparação falha então o estado TRILHA INCORRETA (TRI) é ativado e se SETOR ATUAL = 1, então é gerado um pulso no RESET quando correr ΔFIM BYTE6 e a operação é cancelada. A simples detecção de TRI não estando no setor procurado não provoca o RESET, nem o estado é liberado para o registro de Estados. Veja o esquema abaixo:



6.13 - Deteção das Marcas Especiais

Cada ME possui uma configuração bem definida qual se ja:

MEID DATA: /FE

CLOCK: /C7

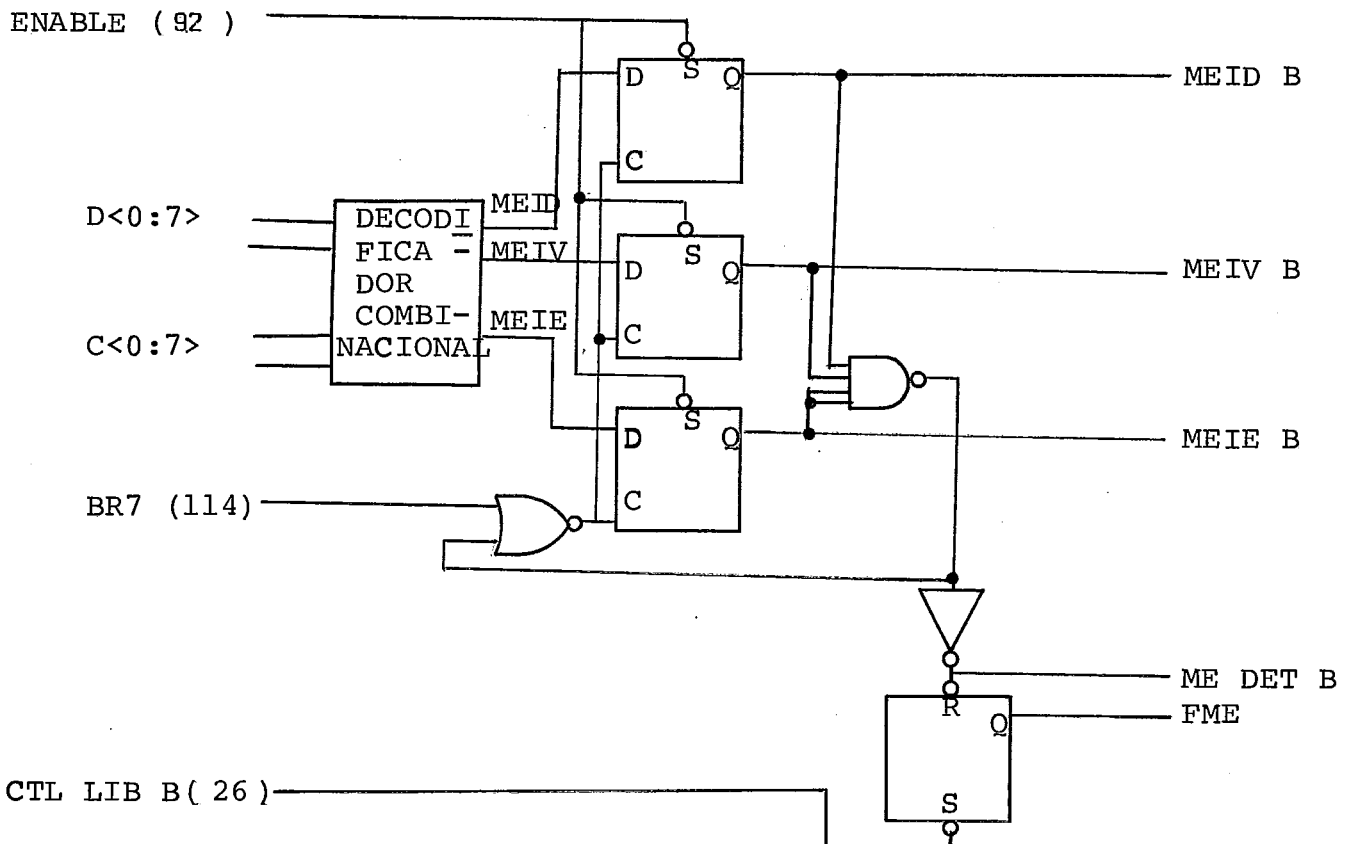
MEIV DATA: /FB

CLOCK: /C7

MEIE DATA: /F8

CLOCK: /C7

O circuito detetor consiste de um decodificador das configurações acima e de um conjunto de 4 FF's implementados da forma abaixo:



A MEIX não é detetada pelo Controlador, embora seja possível de ser gravada.

No início de qualquer execução o estado FME é ativado pelo CTL LIB B. Se, entretanto, pelo menos uma ME for detetada antes do final de operação, ele é desativado. Deve-se observar que, uma vez detetada uma ME, o BR7 é inibido e os FF's só serão desativados quando ENABLE = \emptyset . A continuidade na detecção de ME em uma trilha é portanto função direta da linha RESET ENABLE do circuito de Sincronismo.

6.14 - Geração do CRC

Cada campo escrito no diskette é terminado com dois bytes de cheque de Redundância Cíclica (CRC). Estes dois bytes de CRC são gerados de uma permutação cíclica dos bits de informação, iniciando com o primeiro bit da ME e terminando com o último bit do último byte do campo (excluindo os bytes do CRC). Esta permutação cíclica é o resto da divisão módulo 2 dos bits de informação (representados como um polinômio algébrico $Q(X)$ pelo polinômio gerador $P(X)$.

ANÁLISE MATEMÁTICA

Consideremos um bloco de informação com \underline{n} bits. A função do gerador de CRC é converter este bloco em um novo bloco com $m+n$ bits, onde \underline{n} é o número de bits

da palavra cheque. No nosso caso $n=16$.

Se os bits do bloco original são q_0, q_1, \dots, q_{m-1} então podemos expressar a informação como:

$$Q_m(X) = q_{m-1} X^{m-1} + q_{m-2} X^{m-2} + \dots + q_1 X + q_0 \quad (1)$$

que é denominado POLINÔMIO MENSAGEM e q_0 a q_{m-1} são os bits mensagem, podendo assumir os valores \emptyset ou 1.

A palavra cheque pode também ser expressa como um polinômio:

$$R(X) = r_{n-1} X^{n-1} + \dots + r_1 X + r_0$$

Se transmitirmos a mensagem $Q_m(X)$ seguida pela palavra $R(X)$ o bloco informação completo será:

$$Q(X) = Q_m(X) X^n + R(X)$$

Podemos definir o polinômio gerador como:

$$P(X) = p_n X^n + p_{n-1} X^{n-1} + \dots + p_1 X + p_0$$

O porque do grau n visa melhorar a detecção de erros. Lembrando as propriedades abaixo da álgebra módulo 2:

$$x_{\text{mod}2}^a + x_{\text{mod}2}^a = 0 = x_{\text{mod}2}^a - x_{\text{mod}2}^a$$

A relação básica entre $Q_m(X)$, $R(X)$ e $P(X)$ é:

$$\frac{Q_m(X) X^n}{P(X)} = Q_0(X) + \frac{R(X)}{P(X)}$$

$Q_0(X)$ é o quociente. Estamos interessados no resto $R(X)$ que é somado a $Q_m(X)$ para formar $Q(X)$. O polinômio é checado na recepção.

$$\begin{aligned} \frac{Q(X)}{P(X)} &= \frac{Q_m(X) X^n + R(X)}{P(X)} = Q_0(X) + \frac{R(X)}{P(X)} + \frac{R(X)}{P(X)} = \\ &= Q_0(X) \end{aligned}$$

O resto depois de processada a mensagem é nulo se a mesma é livre de erro. Pode ser provado que a seleção apropriada do polinômio gerador conduz a um resto não nulo com a ocorrência dos erros mais comuns na informação.

CONSIDERAÇÕES SOBRE O POLINÔMIO GERADOR

Assumindo $n=16$ e o polinômio gerador satisfazendo as condições abaixo:

- (1) - $P(X)$ divisível por $1+x$;
- (2) - $P(X)$ contendo os termos 1 e x^{16} ; e
- (3) - O expoente ao qual o polinômio pertence sendo maior que o comprimento do bloco. Entende-se

por expoente do polinômio $P(X)$ ao menor e para o qual $(x^e - 1) = (x^e + 1)$ é exatamente divisível por $P(X)$.

Podemos garantir a seguinte capacidade na detecção de erros:

- qualquer número ímpar de erros;
- todos os erros duplos;
- qualquer tipo de erro num comprimento 16 ou menor;
- 99,994% de detecção em rajadas de erro de comprimento 17;
- 99,997% de detecção em rajadas de erro maiores do que 17.

Uma rajada de comprimento K é qualquer número de erros ocorrendo dentro de K células de bits no bloco de informação.

Pode-se demonstrar que, se o polinômio contém $x^{16} + 1$ e mais dois termos, ele satisfará o primeiro critério.

Para qualquer campo do diskette adota-se o polinômio:

$$P(X) = x^{16} + x^{12} + x^5 + 1$$

O expoente e calculado é 32767 (foi desenvolvido juntamente com um programa de simulação do CRC, outro que calcula e para qualquer polinômio).

A divisão descrita acima pode ser efetuada por uma ló-

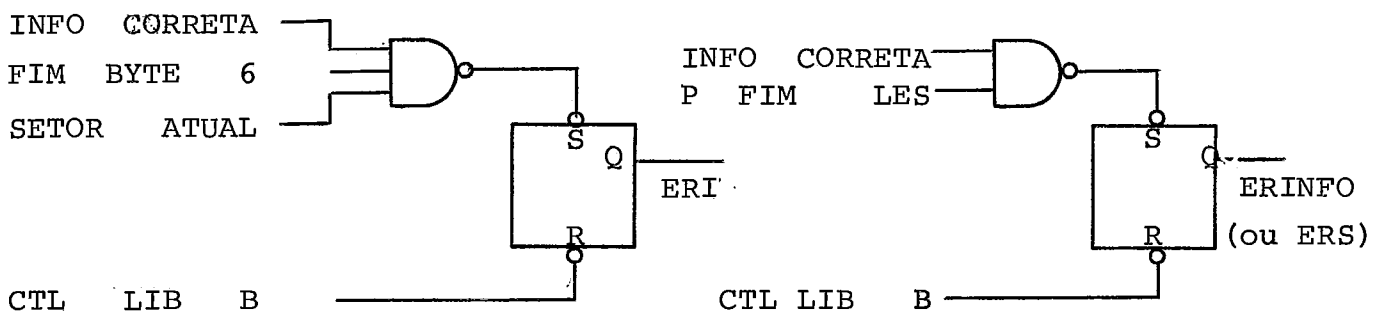
gica relativamente simples. A FIG. 17 mostra o circuito utilizado como gerador e verificador. Para maiores detalhes do porque da implementação, consulte a Bibliografia 1.

A fim de evitar bytes de CRC todos nulos, todos os flip-flops no diagrama lógico são colocados em 1 inicialmente. Na prática como só dispomos de CLEAR, utiliza-se o complemento do CRC.

Na FIG. 17, P INST AM garante-me a condição inicial antes do deslocamento do primeiro bit da ME na escrita, enquanto ENABLE garante a condição inicial para a leitura. Como ENABLE só é ativado após o segundo bit da ME, os dois primeiros FF's durante a leitura são livres.

Detetada uma micro CRC durante uma operação de escrita, SEL CRC B inibe a entrada dos bits de informação durante o tempo de 1 byte e CRC 15 fornece serialmente os bits do CRC correspondente. Duas micro CRC encadeadas gravam os dois bytes completos.

Os circuitos abaixo mostram a detecção de erro de CRC no campo de identificação e no campo de informação.



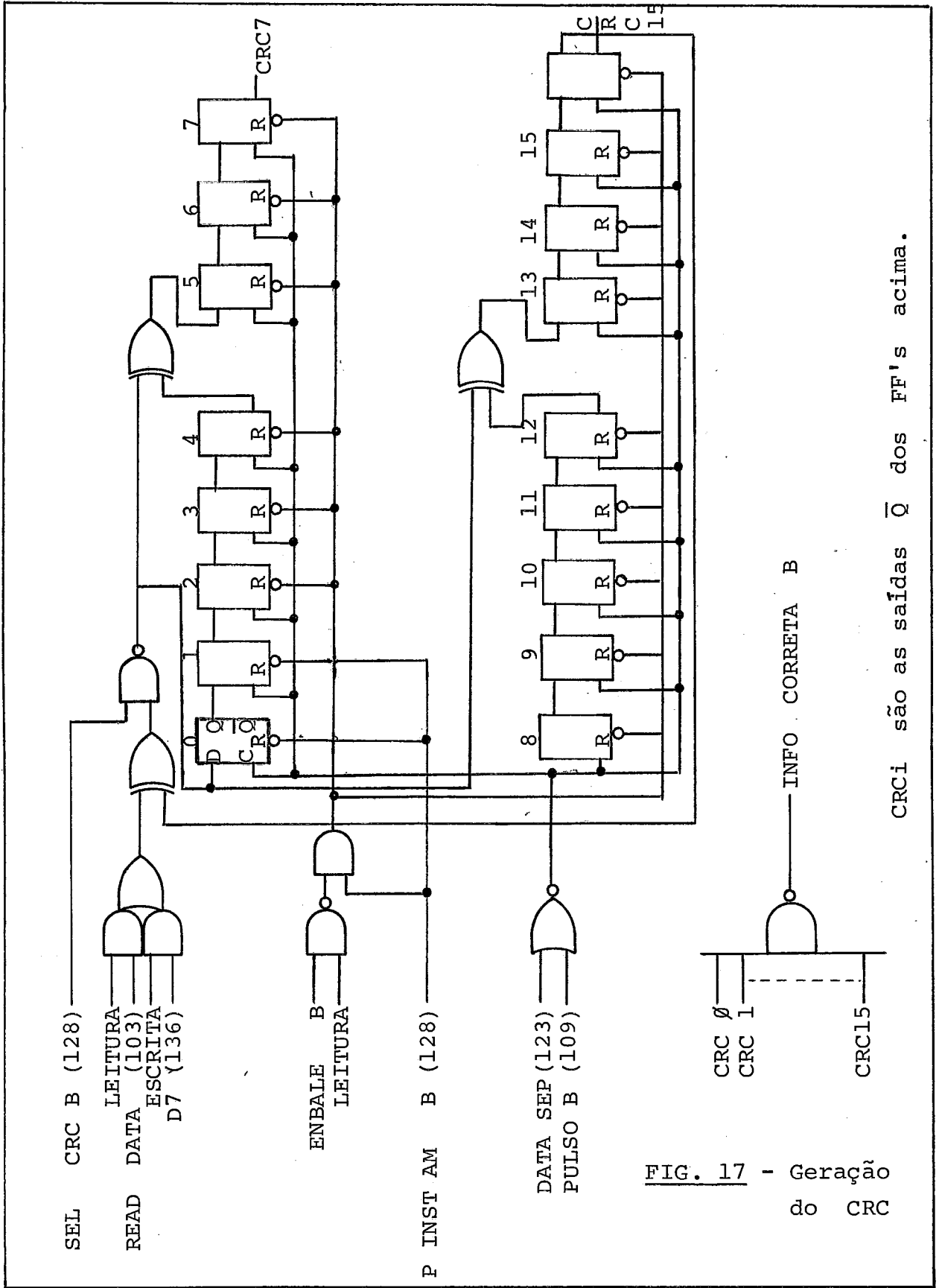


FIG. 17 - Geração do CRC

CRCi são as saídas \bar{Q} dos FF's acima.

6.15 - Controle de Transferências

O Controle de Transferência é responsável pelas transferências de bytes, dados ou microinstruções, entre o Controlador e o Canal. Devido a variedade de pedidos, dividiremos a descrição do circuito em duas partes:

- TRANSFERÊNCIA EM LEITURA; e
- TRANSFERÊNCIA EM ESCRITA.

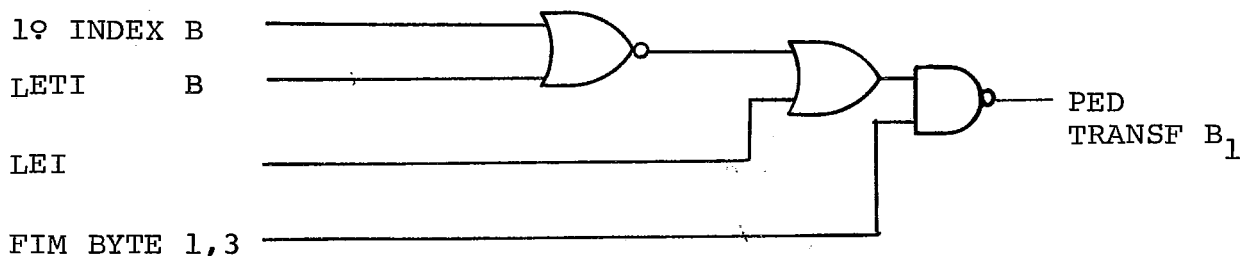
TRANSFERÊNCIAS EM LEITURA

O pedido de transferência ocorre:

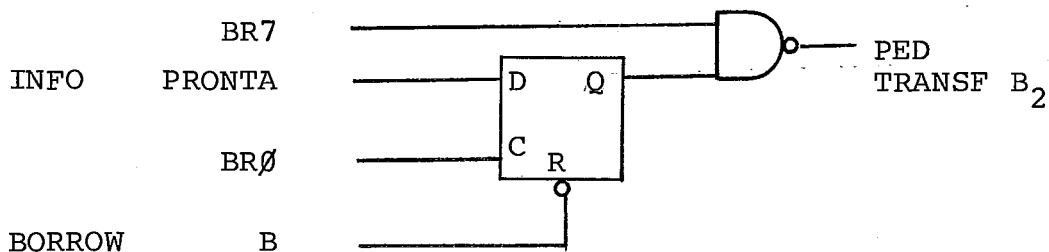
- a - para as instruções: LEI e LETI com FIM BYTE1, 3
- b - para as instruções: TCI e LES ao final de cada byte do campo de informação com BR7.

Em ambos os casos acima significa que, após o armazenamento do último bit do byte a ser transferido, é feito o pedido. Pressupõe-se, uma vez que o próprio Registro de Informação é usado diretamente como registro de armazenamento, que o pedido será atendido antes do próximo pulso no CK DATA, ou seja dentro de 4 us. O protocolo do CANAL me garante este atendimento. No caso da instrução LETI, as transferências devem ocorrer sequencialmente a partir da primeira identificação da trilha e, por isso, os pedidos só são liberados após a detecção do 1º INDEX.

Implementação do CASO /A/



Implementação do CASO /B/



O FF garante que o BR7 que faz o pedido é realmente o correspondente ao fim do byte. O BARROW evita que sejam feitos pedidos para transferências de bytes do CRC.

OBS: INFO PRONTA = TRANSF SETOR.SETOR ATUAL.INFO SETOR

TRANSF SETOR = TCI + LES

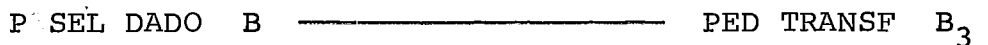
TRANSFERÊNCIAS EM ESCRITA

Complicação maior aqui surge, porque devemos levar em conta dois fatos:

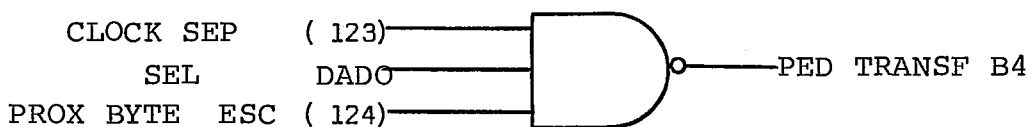
- Como é buscado o primeiro byte;
- como são buscados os bytes seguintes.

Se a instrução não é de Formatação então os pedidos de transferência devem acontecer quando SEL DADO = 1. As condições que satisfazem a transferência são:

- a - O SEL DADO que acontece na decodificação de SEL DADO, deve pedir a busca do primeiro byte de informação .

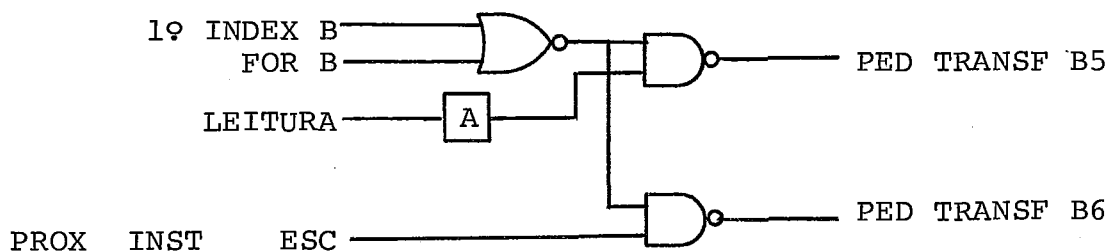


- b - Os bytes seguintes são buscados a cada PROX BYTE ESC, condicionados a CLOCK SEP. O condicionamento a CLOCK SEP não é relevante podendo ter sido usado GAT MICRO ou qualquer outro pulso durante PROX BYTE ESC. O protocolo do Canal garante que terei o byte antes da ocorrência do DATA SEP, 2us depois.

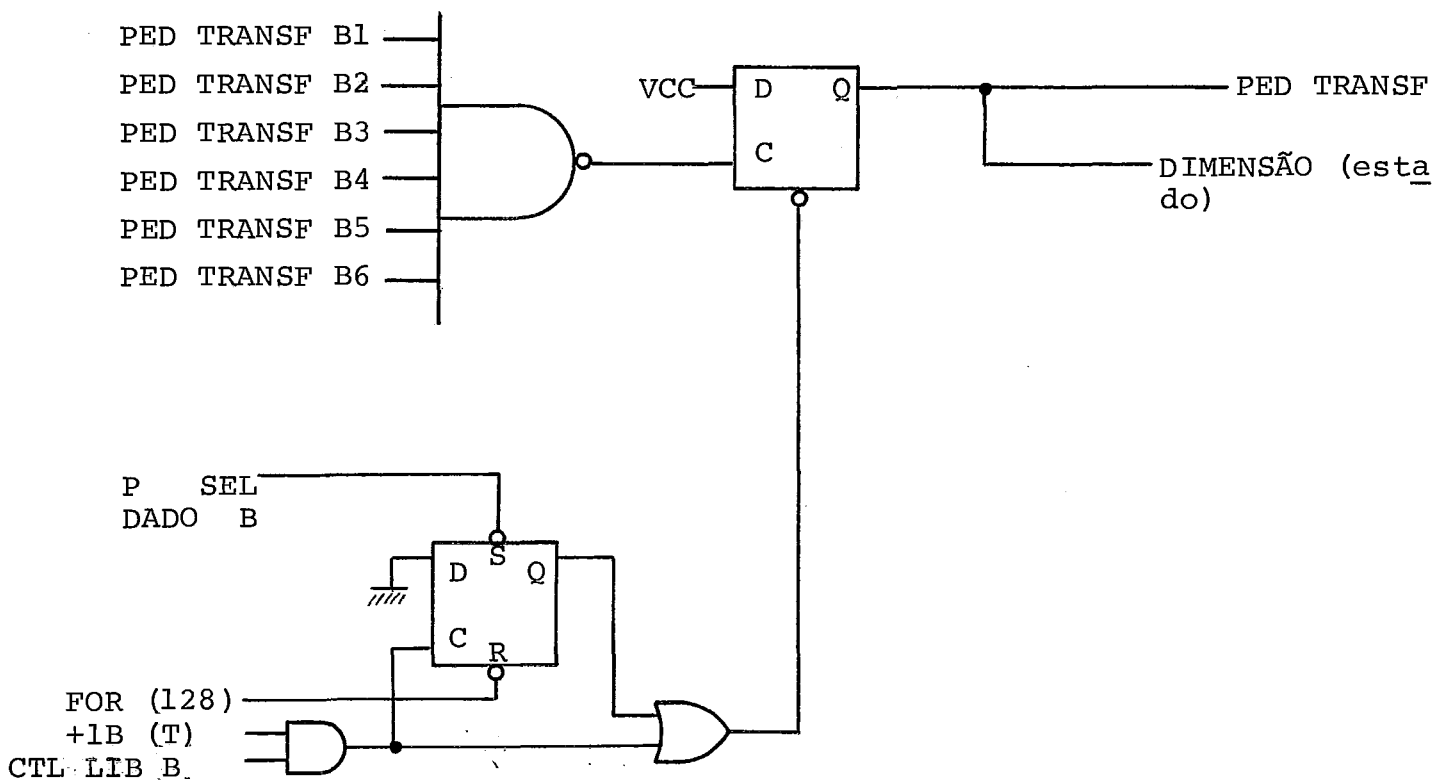


No caso de uma instrução de Formatação, além da busca acima de dados é preciso fazer os pedidos dos bytes

de microinstrução. Quando a operação de Formatação é liberada com a detecção do 1º INDEX então o Controlador deve buscar o primeiro byte de microinstrução. Os bytes subsequentes devem ser pedidos com PROX INST ESC Temos a implementação abaixo:



Todos os pedidos de transferência gerados acima são unificados em um só e compõem o circuito total.



O Protocolo simples seria zerar o PED TRANSFER com +1 B. Entretanto, durante a Formatação, ocorre um caso particular em que devemos inibir o +1. Quando se busca uma instrução BYTE ou DADOS, a decodificação de SEL DADO gera um novo pedido. Este pedido ocorre antes que o +1 correspondente à busca da microinstrução anterior tenha já acontecido. Para se evitar que a chegada deste +1 desative o pedido antes dele ser aceito, foi colocado o FF. Observe que, quando P SEL DADO B pulsa, ele ativa o FF que inibirá o +1 que chega. O +1 seguinte, que confirma a transferência do byte de dado, desativa o FF e, ao mesmo tempo, o PED TRANSF também. Se a operação não é de Formatação, o FF não influi, sendo mantido zero pelo FOR = 0.

Se ao final da operação executada, o PED TRANSF é 1, então ocorreu um erro de Dimensão, que é um estado fornecido pelo Controlador.

6.16 - Estados

Os Estados do Controlador são completos, e fornecem toda a informação necessária para um desempenho perfeito do software. Eles já foram descritos na Descrição do Software e sua geração mostrada no decorrer da explanação do hardware. Acham-se agrupados em duas palavras:

PRIMEIRA PALAVRA - (lida com INPUT 4)

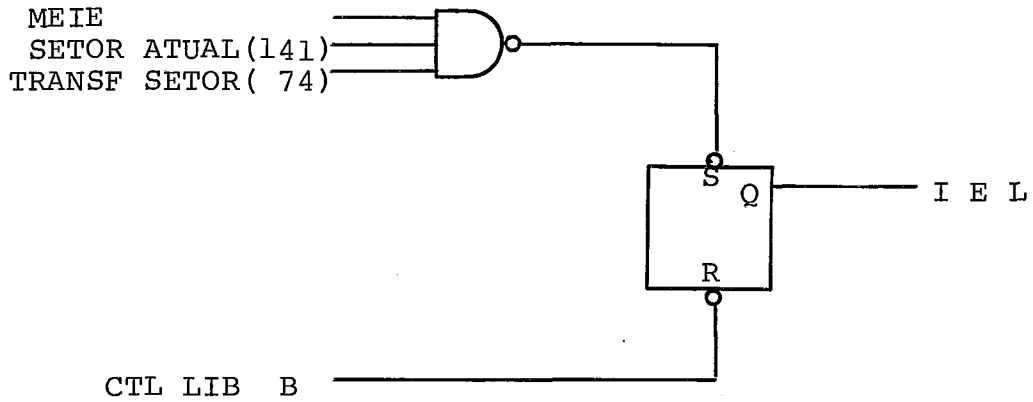
- b∅ - OPERAÇÃO INCORRETA
- b1 - INSTRUÇÃO REJEITADA
- b2 - PERIFÉRICO PRESENTE
- b3 - INFORMAÇÃO ELIMINADA
- b4 - TRILHA ZERO
- b5 - UNIDADE b SELECIONADA
- b6 - UNIDADE SELECIONADA NÃO-PRONTA
- b7 - OCUPADO

SEGUNDA PALAVRA - (lida com INPUT 5)

- b∅ - DIMENSÃO
- b1 - ERS
- b2 - ERI
- b3 - ---
- b4 - SNE
- b5 - TRI
- b6 - FME
- b7 - ESCRITA PROTEGIDA

Dos estados acima dois são fornecidos diretamente pela unidade TR∅ e ESCRITA PROTEGIDA. Os outros 4 são imediatos: PERIFÉRICO PRESENTE (END), UNIDADE PRONTA, UNIDADE SELECIONADA e OPERAÇÃO INCORRETA (OEI).

OEI = DIMENSÃO + ERS + ERI + SNE + TRI. Dos restantes falta apenas mostrar a detecção de INFORMAÇÃO ELIMINADA que poderá ser vista a seguir:



Convém ressaltar que todos os estados detetados pela lógica são zerados inicialmente pelo CTL LIB B, exceto SNE e FME, que são ativados pelo mesmo CTL LIB B, inicialmente.

7 - CONSIDERAÇÕES FINAIS

O projeto do Controlador teve início em Maio/75 com a fase de pesquisas bibliográficas, e o seu funcionamento completo e perfeito deu-se em Agosto/76.

Durante a fase de depuração e testes, a unidade de disco flexível foi simulada por um Gerador de Palavras, enquanto circuitos, como o gerador de CRC, foram simulados em software, permitindo acompanhar o desempenho do hardware, passo-a-passo, para as várias condições de simulação.

Devido a falta de informações relativas ao desgaste do disco flexível, foram elaborados programas com o intuito de avaliar a taxa de erro devido a desgastes físicos e magnéticos. Foram efetuados cerca de 5.200.000 acessos sobre a trilha 76 (mais crítica) em leitura e escrita, constatando-se todos os erros serem recuperáveis numa segunda tentativa, bem como não repetitivos consecutivamente, mas só em intervalos bastante grandes e aleatórios, caracterizando-se erros devido a estática. Pode-se perfeitamente concluir ser o disco flexível viável, mesmo em sistemas de acessos constantes, como em sistemas operacionais.

O custo total de componentes (circuitos integrados, resistências e capacitores) foi da ordem de Cr\$ 4.000,00 (Maio / 77). Atualmente, o custo da unidade oscila em torno de \$500, havendo uma previsão de queda dos preços para \$350, em larga escala, em 1980 (USA).

O Controlador é estruturado e uma modificação para dotá-lo de um protocolo geral para interfaceamento com outros computadores é viável e fica como sugestão para um trabalho futuro. O custo acessível do conjunto, torna-se bastante atraente a sua reprodução e utilização em outros sistemas.

B I B L I O G R A F I A

PETERSON, W. W. & BROWN, D. T.

"Cyclic Codes for Error Detection" In: Proceedings of IRE ,
JAN/61.

CALCOMP - "Technical Manual: Model-140 - Floppy Disk Drive",
Pat. Nº 10150:901.003-01.

NCE/UFRJ - "Manual do Usuário do Terminal Inteligente", Rio de
Janeiro - NCE/76.

SHUGGART ASSOCIATES - "IBM - Compatibility Reference Manual", 1973.

SHUGGART ASSOCIATES - "SA-900 Diskette Storage Drive", 1973.

IBM - "Introduction to the IBM Diskette".

PEATMAN, JOHN, B.

"The Design of Digital Systems"

McGraw Hill

FLORES, IVAN

"Peripheral Devices" - Prentice Hall.

TEXAS INSTRUMENTS INC. - "The TTL Data Book for Design Engineers".

TAKANO, D. F. - "Sistemas Digitais: - Um Controlador Cassette"

Rio de Janeiro - COPPE/74

Tese (M.Sc.) - COPPE/UFRJ.

RCA SOLID STATE DIVISION - "RCA COS/MOS Phase Locked Loop"

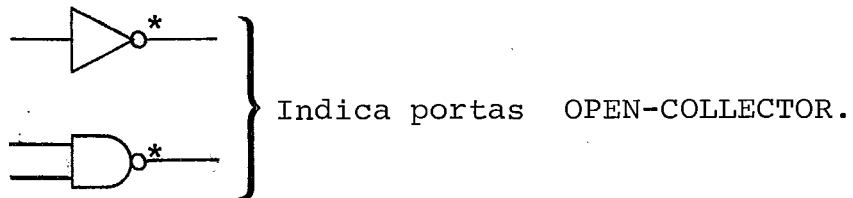
Application Note ICAN-6101.

A P Ê N D I C E

1 - CONVENÇÕES

SINAL "C" - Indica um sinal cujo estado normal é "0" e ativo em "1".

SINAL "B" - Indica um sinal cujo estado normal é "1" e ativo em "0".



BARRA<0:7> - Ao nome BARRA estão associados oito (8) sinais a saber: BARRA0, BARRA1,, BARRA7.

/XX - Indica uma representação em HEXADECIMAL.

E - Indica uma operação AND.

OU - Indica uma operação OR.

FF - Condensação de Flip-Flops.

SINAL (número) - O "número" entre parêntesis, indica a página origem do sinal.

2 - EXEMPLO DE PROGRAMAÇÃO

A seguir é descrito um programa de formatação para o Controlador, escrito em Linguagem ASSEMBLER do Terminal Inteligente. O programa é real e, através de dados fornecidos pelo painel do TI, possibilita uma formatação qualquer do disco flexível.

ZASH
XREF
ISIM

```
*****
*          PROGRAMA DE FORMATAÇÃO INTERATIVO          *
*          ***          *
* LUZES (HEXA)      FORNECER          *
*          *          *
*          /FF      /00  OPÇÃO NÃO IBM  *
*          *          /01  OPÇÃO IBM   *
*          *          *          *
*          /F7      /HH  TAMANHO DO SETOR *
*          *          *          *
*          /F3      /HH  NÚMERO DE SETORES *
*          *          *          *
*          /F1      /HH  NÚMERO DE ZEROS  *
*          *          DO PREAMBULO 2     *
*          *          *          *
* APÓS A INTERAÇÃO ACIMA O PROGRAMA PEDE          *
* SUCESSIVAMENTE AS IDENTIFICAÇÕES DOS          *
* SETORES A PARTIR DO PRIMEIRO COLOCANDO          *
* NAS LUZES A ORDEM FÍSICA DO MESMO SEGUE          *
* A INTERAÇÃO ABAIXO          *
*          *          *          *
*          1- O PROGRAMA COLOCA NAS LUZES A OR-   *
*          DEM DO SETOR          *
*          *          *          *
*          2- O OPERADOR COLOCA NAS CHAVES A      *
*          IDENTIFICAÇÃO E APERTA VALIDADE        *
*          *          *          *
*          3- O PROGRAMA COLOCA NAS LUZES O      *
*          VALOR LIDO E ESPERA CONFIRMAÇÃO        *
*          *          *          *
*          4- O OPERADOR FORNECE A IDENTIFICA    *
*          ÇÃO DEFINITIVA COM O BIT7 DAS          *
*          CHAVES DESLIGADO          *
*          *          *          *
*          5- SE O BIT7=1 ENTÃO O PROGRAMA RE-   *
*          PETE OS PASSOS A PARTIR DE 1          *
*          *          *          *
*****
```

***** DADOS INICIAIS *****

```
*
      ABS          /0100
      ENTRY       INIT
INIT   LDI /FF          FAZ OPCAO DE FORMATACAO
      CAL LUZFS       LUZES=/H0 FORMATO NAO IBM
      LBA           LUZES=/H1 FORMATO IBM
*                               B=OPCAO DE FORMATACAO
*
      LDI /F7
      CAL LUZFS       L=TAMANHO DO SETOR
      LCA           C=TAMANHO DO SETOR
*
      LDI /F3
      CAL LUZFS       L=NUMERO DE SETORES
      LEA           E=NUMERO DE SETORES
*
      LDI /F1
      CAL LUZFS       LUZFS=PREAMBULO2
      ORI /80        A=PREAMBULO2
*
      SUI 1
      LRS LPZXAD+1   COLOCA O LPZXAD
      LMA
*****
```

***** INICIO DE TABELA P/ FORMATO IBM *****

```
*
      LRS TABELA
      LMI LPZ46
*
      RST INCHL
      LMI HEX
*
      RST INCHL
      LMI LPZ32
*****
```

```
*
      LAB
      RAR
      JTC IBM
*
```

***** INICIO DE TABELA P/ FORMATO NAO-IBM *****

```
*
      LRS TABELA
      LMI LPZ10
*****
```

```
*
***** INTERACAO DOS SETORES *****
IBR          LBI 1          APONTA O PRIMEIRO SETOR
*
LOOP         RST INCHL
            LMI WID
*
            RST INCHL
            LMI WTR
*
            RST INCHL
            LMI LPZI
*
            RST INCHL
            LMI BYTE
*
DEHOVO      LDR          VALOR DO SETOR EM D
            CAL LUZES    LE VALOR DO SETOR
            CAL LUZES    CONFIRMA VALOR DO SETOR
            RAL          BIT8=1 SETOR INVALIDADO
            JTC DEHOVO   BIT8=0 SETOR VALIDADO
            RAR
            RST INCHL
            LDA
*
*
            IBR          APONTO PROXIMO SETOR
*
*
            RST INCHL
            LMI LPZI
            RST INCHL
            LMI CRC
            RST INCHL
            LMI CRC
            RST INCHL
            LMI LPZI7
            RST INCHL
            LMI BDA7
            RST INCHL
            LMI LPZS
            RST INCHL
            LMI CRC
            RST INCHL
            LMI CRC
            RST INCHL
            LMI ***
LPZXAD
*
OCE
JFZ LOOP    DESVIA PARA PROXIMO SETOR
```

RST 10CHL
LMI LPZS
RST 10CHL
LMI LPZS
RST 10CHL
LMI LPZS

*
***** TABELA PRONTA

*
LRD TABELA
LAL
SUE
LFA DE=TAMANHO DA TABELA
LAH (N BYTES - 1)
SRD
LDA
*
LBI 76 B=REFERENCIA A TRILHA 76
*
LAI /EO
SFL 0 SELECIONA DISKETTE
*
LAI 0
CTL 3 SELECIONA UNIDADE 1
*
LAC
CTL 1 TAMANHO DO SETOR
*
*
FOR LAI /CV
SEL 0 SELECIONA CANAL
*
LAF
CTL 2 PARTE BAIXA DO TB
*
LAD
CTL 1 PARTE ALTA DO TB
*
LRH TABELA
LAH
CTL 3 PARTE ALTA DO END
*
LAL
CTL 4 PARTE BAIXA DO END
*
CTL 5 SELECIONA ESCRITA
*
*
LAI /EO
SEL 0 SELECIONA DISKETTE

```
*
OCUP          STATUS
              RAL          TESTA OCUPADO
              JTC OCUP

*
NPRT          JOP
              RAL          TESTA UNIDADE PRONTA
              JTC OCUP

*
              LAB
              CTL 7          POSICIONA NA TRILHA=0

*
OCUPI         STATUS
              RAL          TESTA FIM DE POS
              JTC OCUPI

*
              LAI 1          FORMATAÇÃO
              CTL 5

*
              CTL 0          INICIA OPERAÇÃO DE CANAL

*
              DCB          APONTA PARA NOVA TRILHA
              JFS FOP

*
              HLI

*
* A ROTINA LUZES COLOCA NAS LAMPADAS INICIALMENTE O VALOR DE D **
LUZES        LAI 0          END DAS CHAVES F LAMPADAS
              SEL 0

*
              LAD          CARREGA A COM D
              CTL 0          ESCREVE NAS LAMPADAS
              CTL 1          ESCREVE NO DISPLAY

*
ESPERA       STATUS
              RAL          ESPERA VALIDAÇÃO
              JFC ESPERA

*
              READ        LE CHAVES PARA ACC
              LDA          COPIA A EM D
* O VALOR DAS CHAVES LIDO ESTA ARMAZENADO EM D E EM A
              RET          FIM DA ROTINA

*
*FIM DO PROGRAMA
```

TABELA	DS	400
INCHL	EQU	/28
LPZ1	EQU	/80
LPZ16	EQU	/89
LPZ17	EQU	/90
LPZ32	EQU	/9F
LPZ46	EQU	/AD
LPZ5	EQU	/30
WEX	EQU	/E4
WID	EQU	/C6
WDAN	EQU	/C3
WTR	EQU	/02
BYTE	EQU	/01
CRC	EQU	/08

*

END INIT

/MONTE
/CADARCO
/FIN