

Coacção: Do Autor

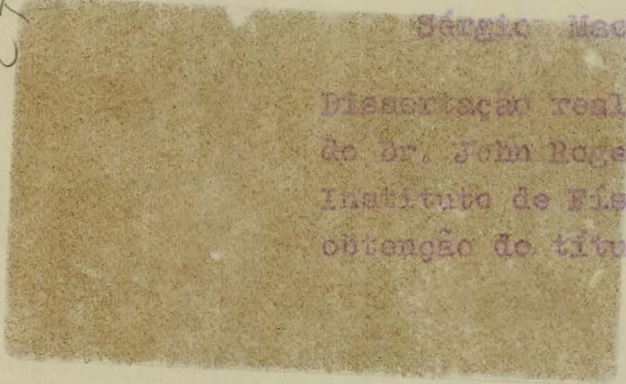
Universidade Federal do Rio Grande do Sul

Instituto de Física

ACOPLAMENTO DE UMA UNIDADE DE FITA

MAGNÉTICA AO COMPUTADOR 2114A

CT 40.00. (Pis)



Sérgio Machado Bordini

Dissertação realizada sob a orientação do Dr. John Rogers, apresentada ao Instituto de Física da UFRGS para a obtenção do título de Mestre em Ciências.

Trabalho parcialmente financiado pelas seguintes instituições: Banco Nacional de Desenvolvimento Econômico, Comissão Nacional de ENERGIA Nuclear, Conselho Nacional de INSTITUTO DE FÍSICA e Organização dos Estados Americanos

INSTITUTO DE FÍSICA
FÍSICA
FÍSICA

Porto Alegre

- 1972 -

SINOPSE

Descrição do acoplamento entre uma unidade de fita magnética incremental (KENNEDY 1400R) e um mini computador (HP 2114A) com 8K de memória. O HARDWARE foi desenvolvido em duas etapas: Modificações na Fita Magnética e Interface pròpiamente dito. Essas modificações na lógica de controle foram necessárias para adequar o Modelo 1400R ao sistema. O SOFTWARE foi escrito tendo em mente a compatibilidade com o Original, afim de poder usa - lo onde conviesse. Construiu - se assim um " DRIVER " (compatível com o SIO DRIVER) e um " LOADER " que carrega programas absolutos. Na geração do Sistema usa - se o " Programa Montador ", construído especialmente para esse fim, que grava na fita os programas componentes. O Sistema operacional compila, carrega e executa programas em FORTRAN, dispensando as fitas de papel binárias.

ABSTRACT

In this work we discuss the design and construction of an interface between a Kennedy incremental magnetic tape (Model 1400R) and a Hewlett Packard computer (HP 2114A) with 8K core. Hardware construction included both construction of the interface itself and the necessary modification to permit usage of the tape drive in the system. Software was written with the intention of maintaining compatibility with the operating system provided by the manufacturer and includes standard SIO driver, and an absolute Loader. A special system generation program was written which produces the system tape. The present operating system compiles, loads and executes Fortran source programs without intermediate binary tapes.

AGRADECIMENTOS

Ao Orientador Dr. John Rogers pelas sugestões e apoio.

Ao Eng. Celso S. Müller pelas excelentes sugestões apresentadas.

Aos colegas da Divisão de Física Aplicada pela cooperação.

Especial à minha esposa que datilografou este trabalho e ao Bolsista Tiaraju V. Wagner pelos excelentes desenhos.

À minha mãe pela ajuda na revisão.

À minha esposa.

I N D I C E

1- INTRODUÇÃO

2- HARDWARE

- MODIFICAÇÕES NA KENNEDY 1400R

- INTERFACE

3- SOFTWARE

I- DRIVER e LOADER

II- Programa Montador

III- Sistema.

4- CONSIDERAÇÕES FINAIS

- APÊNDICES

1-- INTRODUÇÃO

Em um sistema de pequeno porte sem memória de massa, compilação e carga de programas são tarefas trabalhosas. Por exemplo: Um mini computador HP 2114A, com 8192 palavras de memória, leitora de fita e teletipo, compila FORTRAN em 2 passos, perfurando 1 fita binária em cada um. A carga de programas objeto no mesmo sistema, implica no manuseio de várias fitas perfuradas (livraria e subrotinas).

Com uma memória de massa (ex. fita magnética) é possível simplificar a operação, pois todos os programas do sistema passam a residir nela, além do que, etapas intermediárias podem ser escritas temporariamente.

No nosso caso, possuímos uma unidade de fita magnética KENNEDY (modelo 1400R), adquirido originalmente para um Multicanal (NUCLEAR DATA), e estava disponível. Decidiu-se, então, acoplá-la ao 2114A, para usá-la como memória de massa. Porém, essa máquina é demasiado lenta -- 7 pistas, 200 bpi e 1000 car/s, criando o problema da busca de um arquivo afastado.

Para minorar este problema foi sugerido um método de busca peculiar: O rôlo de fita poderia ser dividido fisicamente em arquivos de um certo tamanho, e a busca seria realizada pelo rebobinamento (REWIND) da máquina que é bastante rápido (10 pés/s). Para avançar a fita na mesma velocidade, o REWIND teria que ser invertido, criando uma operação de FAST FORWARD que não existe originalmente no modelo 1400R. O REWIND é terminado pela detecção de marca refletora na fita, sendo, portanto, usadas para dividi-la.

Além do FAST FORWARD outras alterações foram necessárias na lógica da KENNEDY, no entanto, a compatibilidade com o Multicanal foi mantida. Para conectar com o interface, 2 novos conectores foram colocados, deixando os originais inalterados.

O interface foi montado em cartão apropriado (fornecido pela HP), empregando-se Circuitos Integrados da família TTL.

Transistores e componentes discretos foram usados principalmente para conversão dos níveis de tensão. Para DRIVERS do BUS do computador, usou-se Circuito Integrado da família C₁L(9956), que são os recomendados pelo fabricante.

O SOFTWARE foi desenvolvido em 3 etapas:

- DRIVERS e LOADER
- Programa Montador
- Sistema

O DRIVER tem uma seqüência de chamada semelhante aos SIO DRIVERS originais, de modo que pode ser usado pelo SOFTWARE da HP ligeiramente alterado. O LOADER carrega na memória programas absolutos, podendo ser chamado tanto pelo operador como por programa, o que permite a carga automática de compiladores, etc.

No sistema usou-se o SOFTWARE do fabricante, alterado para operar o Periférico. Pode-se compilar FORTRAN e realizar sua Carga e Execução, sem fitas de papel intermediárias. Também é possível compilar obtendo fita objeto, assim como carregar programas a partir de fitas de papel.

2- HARDWAREa- KENNEDY 1400R

A unidade de fita Magnética KENNEDY 1400R pode operar com gravação incremental ou contínua, sendo o processo de gravação empregado o NRZI (Non Return to zero IBM). Optamos pelo modo contínuo, pois a velocidade é maior, e o computador pode atender os dados das linhas em tempo.

Para comandar a operação "WRITE", devemos suprir um trem de pulsos na entrada "WRITE STEP" da máquina, enquanto as linhas de dados são atuadas. A frequência deve ser de 1000PPS em regime permanente, sendo acelerada no início da operação até os 1000PPS.

O selecionamento "WRITE/READ" deve ser providenciado pelo interface, da seguinte forma:

Para escrever (WRITE): Curto circuitos as linhas:

- WRITE SELECT
- SELECTOR COMMON

Para ler (READ): Curto circuitos as linhas:

- READ SELECT
- SELECTOR COMMON

Isto pode ser realizado facilmente por uma chave no interface ou por intermédio de relés! No entanto, optamos por uma solução eletrônica, podendo a operação ser presetada por um bit (14) de comando.

Paridade: A paridade na gravação é gerada pela própria máquina não requerendo cuidados. Pode-se optar por PAR, ÍMPAR ou DESLIGADO mediante a atuação de um interruptor: Escolhemos a paridade ÍMPAR, pois assim sempre haverá sinal na leitura, mesmo quando for lido um caracter nulo. A operação WRITE está esboçada na figura 2.1.

GAPS: Os espaços entre os registros são gerados pela lógica da 1400R. São dois:

- RECORD GAP: 3/4 Pol. com um LCC (longitudinal Check character) no início.

FIG. 2.1

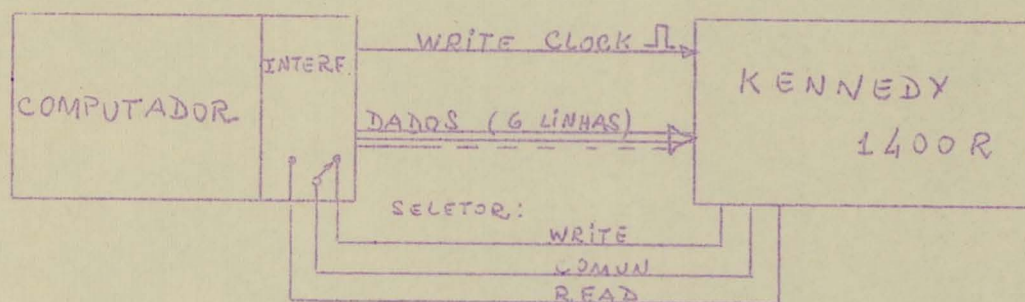
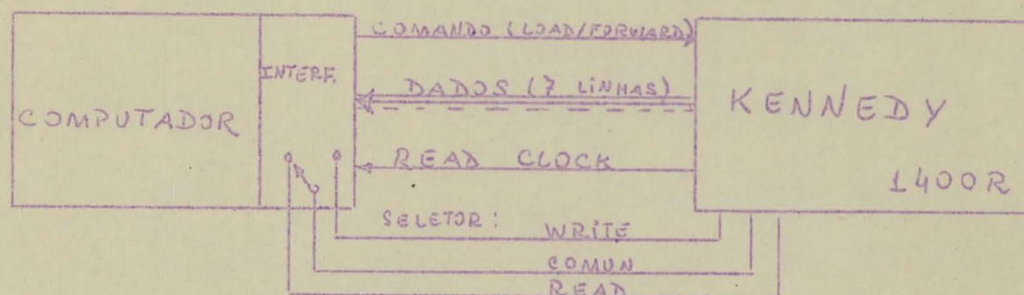
OPERAÇÃO WRITE

FIG. 2.2

OPERAÇÃO READ

- FILE GAP: 3,4 Pol. contendo uma marca de arquivo 178 e seu LCC.

Ambos são comandados por pulsos de 10 μ s (mínimo) nas linhas respectivas, e retornam para o interface o sinal "GAP IN PROCESS" que vai para zero durante a gravação do GAP. Neste caso, a fita é avançada pelo oscilador interno de 1000PPS que também funciona na leitura.

LEITURA

A leitura é contínua, e é inteiramente realizada pela máquina: O sinal "LOAD/FOWARD" atuado, provoca o avanço da fita pelo oscilador interno a 1000 steps/s. Os sinais dos bits captadas pela cabeça, são sincronizadas pelo dispositivo "SKEW DELAY", do seguinte modo: É contado um certo tempo (500 μ s) a partir do primeiro bit lido; enquanto isso todos os bits vão para um registrador interno. Após os 500 μ s um sinal de "CLOCK" é gerado, e este lê os registradores, jogando os sinais para o interface. Após o "CLOCK", os registradores são limpados e o ciclo se repete.

Dêste modo, tem-se nas linhas de saída todos os bits simultâneamente, resolvendo o problema do "DELAY" resultante da gravação. Na figura 2.2 mostra-se como comandar uma leitura contínua.

A paridade transversal é checada no Interface, por uma árvore de "XOR GATES". A paridade ímpar foi a escolhida tendo em vista que no sistema de gravação NRZI um sinal de nível lógico 1 é representado na fita pela variação de fluxo magnético, e portanto um sinal zero não dá nenhuma Saída na cabeça leitora. O sistema de paridade PAR admite uma combinação (00) que não é detectada pelo "SKEW DELAY", e essa combinação existe em se tratando de programas em Binário, enquanto que em ASCII ela pode ser evitada.

Durante a leitura, os Gaps podem ser detectados pelo "GAP DETECT" incluso na máquina, e sua lógica é tal, que "1" representa um registro de informações e zero indica "GAP".

POSICIONAMENTO

O sistema de posicionamento adotado implicou em modificações na KENNEDY: Foi adicionado um cartão contendo circuitos lógicos integrados TTL (juntamente com sua fonte de 5v), que tem por fim controlar a operação de posicionamento através dos servos que atuam os carretéis.

DESCRIÇÃO da OPERAÇÃO de POSICIONAMENTO para FRENTE

Supondo que se esteja em um arquivo de número 1 e se deseje atingir um arquivo 3, a fita deve ser avançada em alta velocidade e parar quando o sensor óptico detectar a passagem do segundo marcador refletivo. Esta operação é comandada pelo computador que conta os marcadores e reinicia a operação até o último marcador ser atingido. (figura 2.3)

DETALHES

A operação original "REWIND" da KENNEDY é realizada pelo aumento de corrente no servo do carretel "SUPPLY", fazendo-o arrastar o servo do "TAKE UP". É fácil portanto obter a operação inversa (FAST FOWARD), aumentando-se a corrente do servo do "TAKE UP". Porém, é interessante ter a fita parada no início do arquivo, antes da marca respectiva; por isso um FAST FOWARD implica em um REWIND subsequente. A parada da fita teve de ser simulada por um DELAY e só ao fim d'êste o computador é informado do término da operação (RW STATUS).

O mesmo sinal foi usado para inibir o solenóide de atuação do "PINCH ROLLER", pois antes dessa modificação a fita poderia ser estragada, ao serem pressionados os botões "REWIND" e "LOAD/FOWARD", ao mesmo tempo.

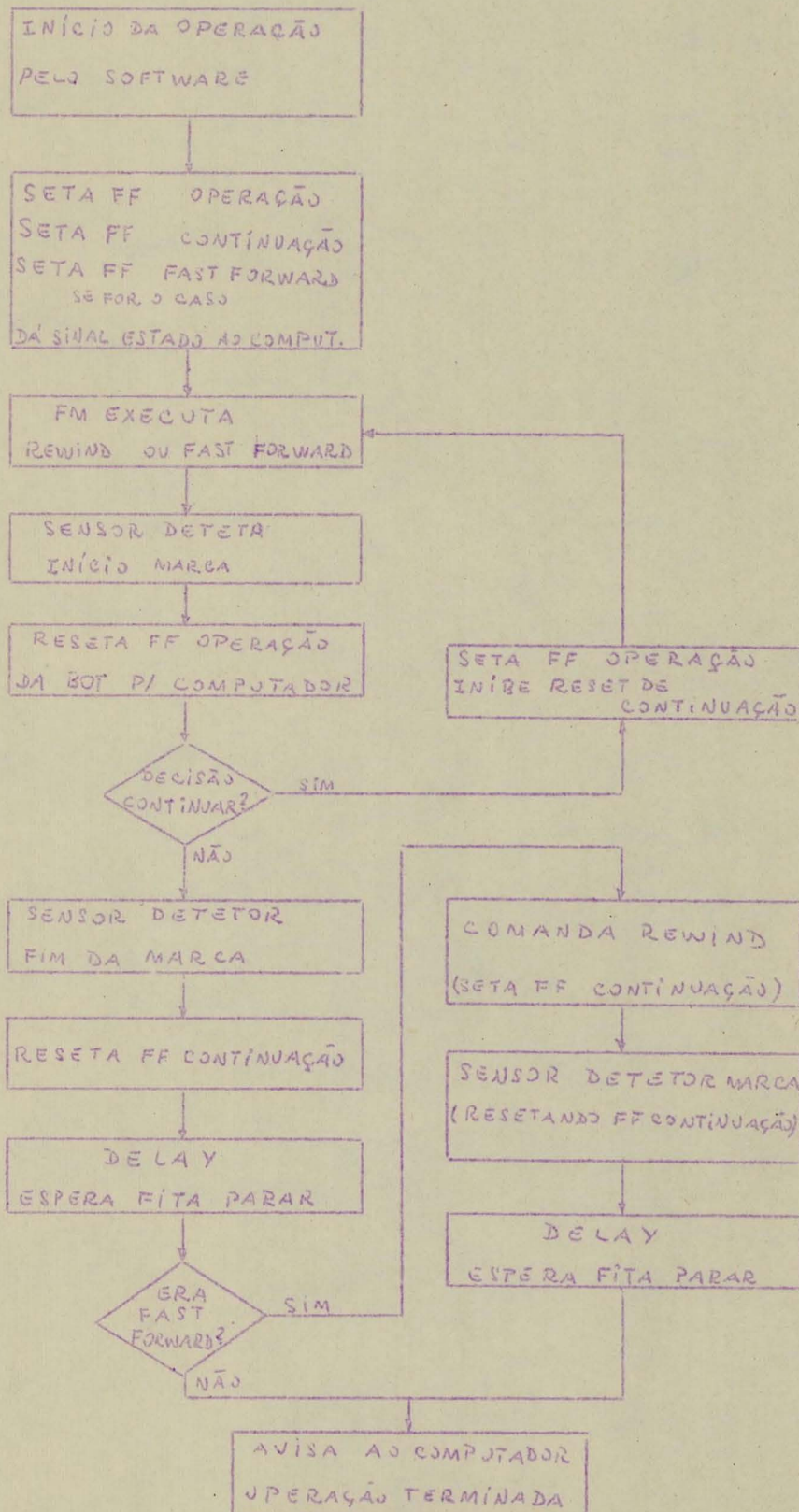
Por ocasião do posicionamento, a indicação do início da marca provoca um estado de "ESPERA" no controle, enquanto o computador decide se vai continuar ou não. Em caso afirmativo, êle atua o comando de operação (FFou RW) e tudo se repete. Caso o controle não seja acionado até a marca acabar de passar sobre o sensor, a operação é terminada com o "RESET" do FLIP FLOP de continuação.

A implementação foi feita na máquina já existente, e o circuito das modificações feitas, está no apêndice A. O diagrama lógico da figura 2.4 mostra como funciona.

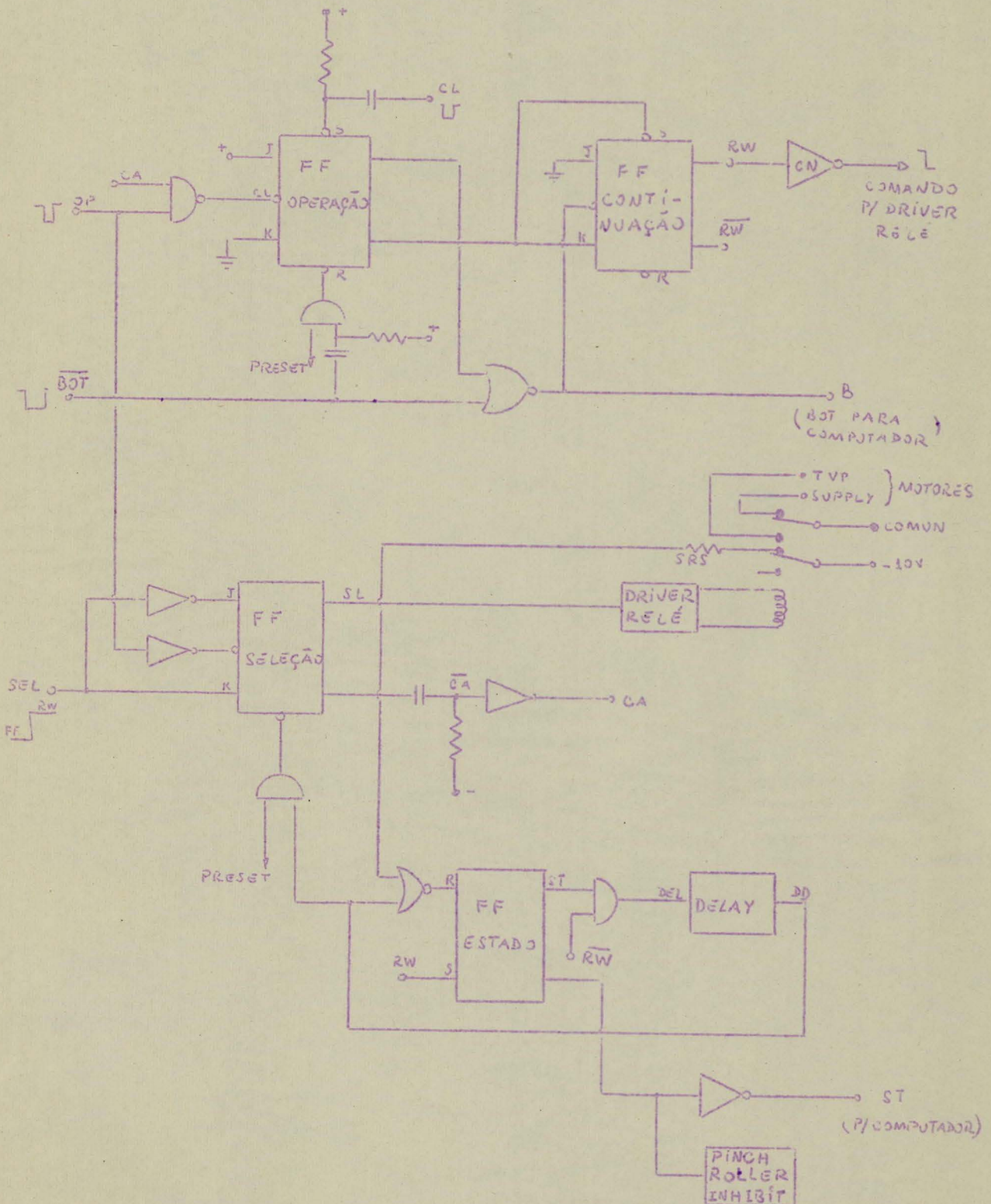
SINAIS: OP: (Operação) Pulso de lógica negativa do interface.
 (refe CL: (Comando local) Lógica negativa, da botoeira da
 re-se máquina.
 à fi- CA: (Comando adicional)
 gura BOT: (Begin of Tape) Provém do sensor ótico. Vai para
 2.4) 1 na presença do marcador.
 RW: Sinal que comanda os relés de acionamento e freio da máquina. Atua no "Control Card" original.
 PRESET: Preseta os flip flops quando for ligada a alimentação.
 B: Sinal de BOT para o Interface.
 FOP: Saída verdadeira do flip flop de operação.
 SEL: Sinal de seleção do Interface { 1- rewind
 { 0- Fast Foward
 ST: Saída verdadeira do Flip Flop de "Status": Estado para Interface.
 DEL: Comando para o circuito de retardo.
 DD: Pulso retardado
 SL: Saída do Flip Flop de Seleção.
 SRS: (Status Reset) Condiciona reset do "Status FF".

POSICIONAMENTOFIG 2.3

DIAGRAMA EM BLOCOS DA OPERAÇÃO.



CONTROLE DE POSICIONAMENTO
DIAGRAMA LÓGICO

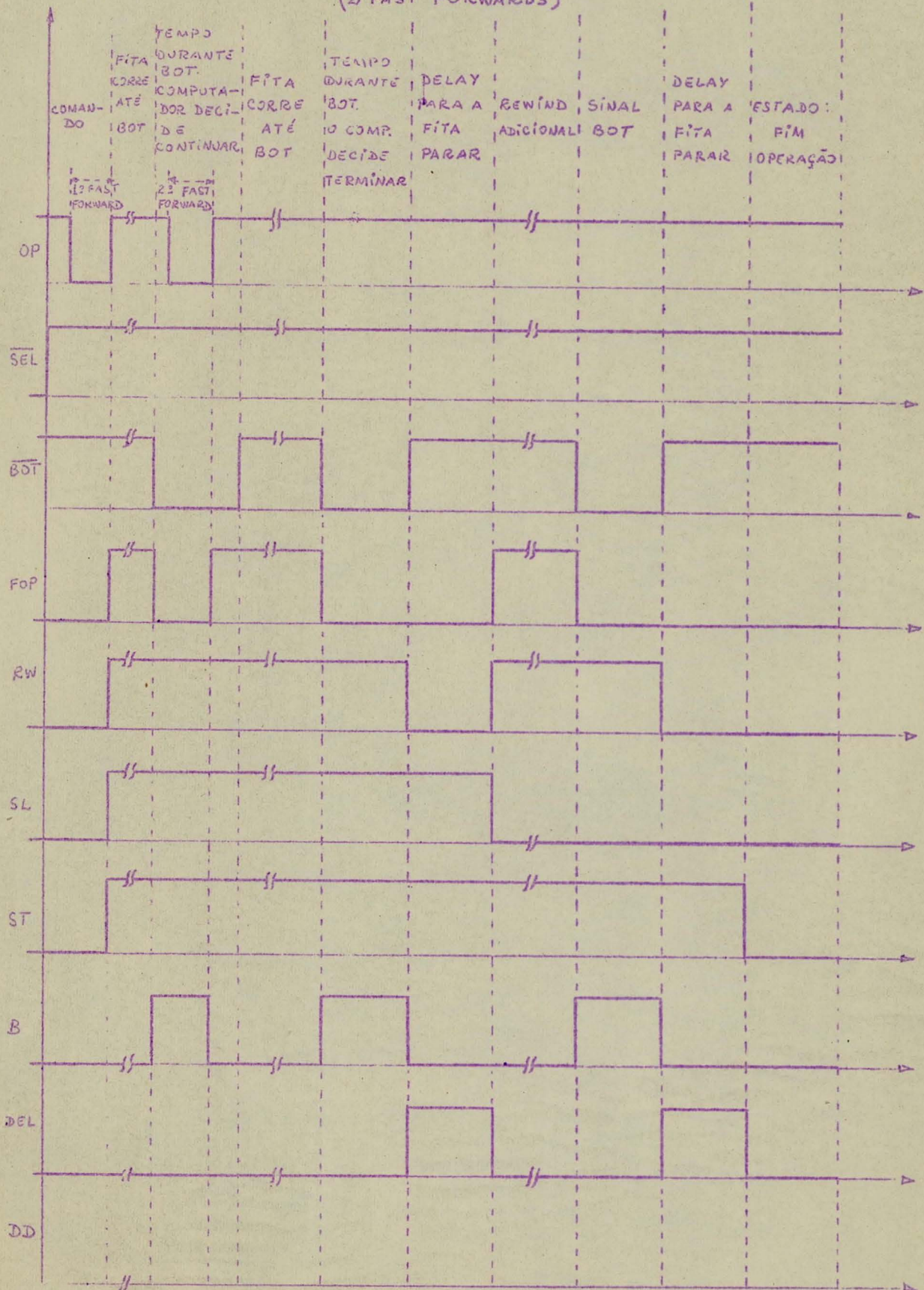


CONTROLE DE POSICIONAMENTO

FIG 2.5

DIAGRAMA DE TEMPOS

(2 FAST FORWARDS)



OBS: OS TEMPOS DE PROPAGAÇÃO FORAM DESPREZADOS PARA SIMPLIFICAR O DIAGRAMA.

RELEITURA

A KENNEDY 1400R originalmente não é dotada de leitura para trás, o que significa que ocorrendo um erro na leitura deve-se voltar a fita com o REWIND até o início, e proceder novamente a busca do registro para relê-lo. É claro que esta operação é demasiadamente morosa, o que implicou em dotarmos a máquina de "BACKSPACE". Parecia ser simples: bastaria inverter 2 fias do estator do motor incremental, para que este invertesse o sentido; porém a natureza do "DRIVER" deste motor incremental impediu o sucesso desta operação. Consta de um Contador em anel com 3 SCS (Silicon Controlled Switches) que são gatilhadas em seqüência pelos pulsos do CLOCK, acionando assim os 3 enrolamentos do motor, através de transistores de potência. Ocorria que qualquer transitório no circuito do motor gatilhava as SCS e travava o Contador. Aliás, outros transitórios provenientes dos relés inclusos na máquina (REWIND e FOWARD) também punham o DRIVER fora de operação, até que fôsse desligada a alimentação da máquina!

Estes fatos nos levaram à substituição de todo o circuito de DRIVER original por outro que utiliza lógica TTL e transistores, permitindo também o BACKSPACE mediante um sinal (RW) que troca a lógica do decodificador.

A unidade consta de um contador (0-2₈), um decodificador com reversor e circuito de saída, utilizando transistores de alta tensão (150v). O contador, construído com flip flops tipo JK possui um elo de RESET que garante a inibição de uma condição inexistente para o decodificador (3₈) e este possui uma lógica de reversão: Quando o sinal RW vai para um, troca os números 0₈ e 1₈, invertendo o sentido de rotação do motor. Este sinal vem diretamente do Interface. O diagrama lógico com tempos da figura 2.6 mostra a idéia, e o apêndice A os circuitos detalhados.

SINAIS: F e F: Pulsos de "CLOCK" proveniente da KENNEDY.

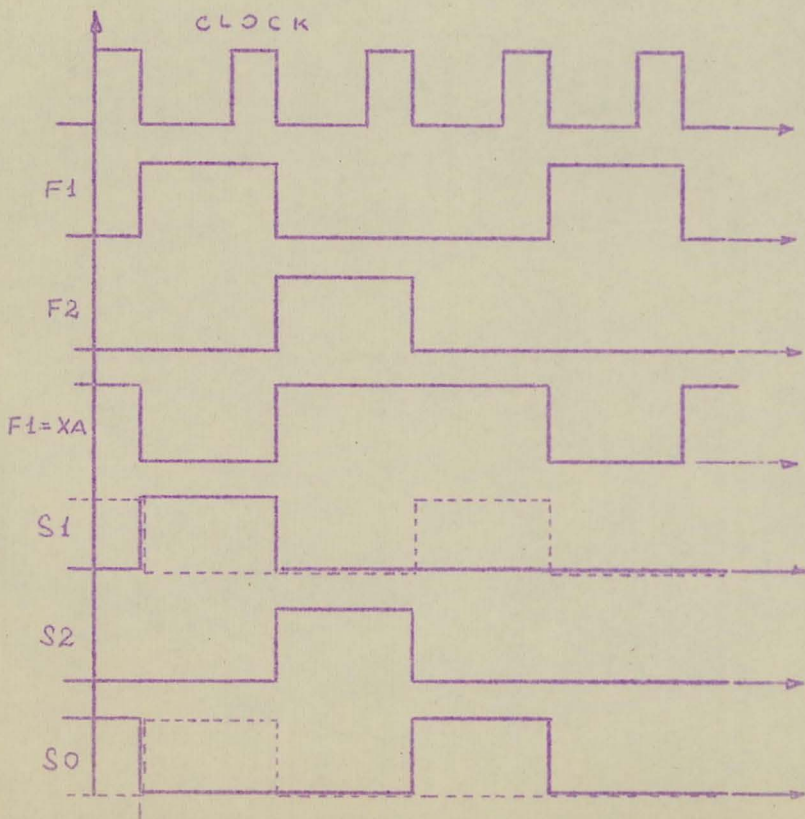
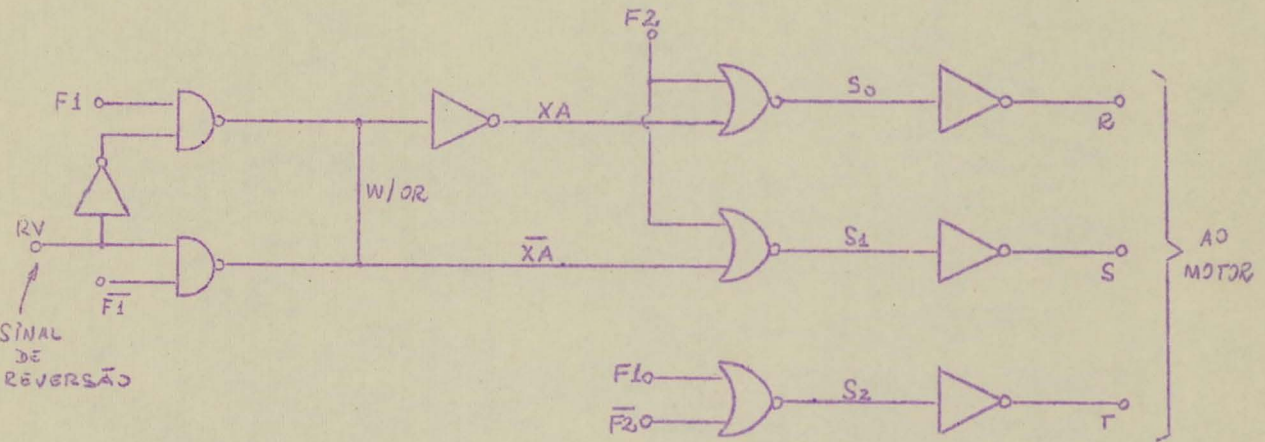
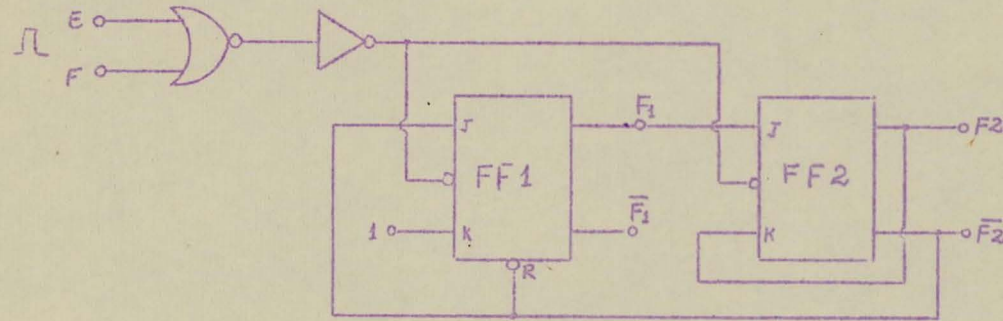
F1, F2, $\overline{F1}$, $\overline{F2}$: Saídas dos "Flip flops" 1 e 2.

RW: Sinal de Reversão.

DRIVER INCREMENTAL

DIAGRAMA LÓGICO E DE TEMPOS

PULSOS
OSCILADOR



TEMPOS:

RV = 0 → LINHA CHEIA
RV = 1 → TRACEJADA

$$\overline{XA} = \overline{(F1 \cdot RV)} \cdot \overline{(F1 \cdot RV)}$$

$$XA = F1 \cdot \overline{RV} + \overline{F1} \cdot RV$$

SE RV = 0 → XA = F1

SE RV = 1 → XA = $\overline{F1}$

$$S0 = \overline{F2 + XA} = \overline{F2} \cdot \overline{XA}$$

$$S1 = \overline{F2 + \overline{XA}} = \overline{F2} \cdot XA$$

$$S2 = \overline{F1 + \overline{F2}} = \overline{F1} \cdot F2$$

XA: Sinal Auxiliar.

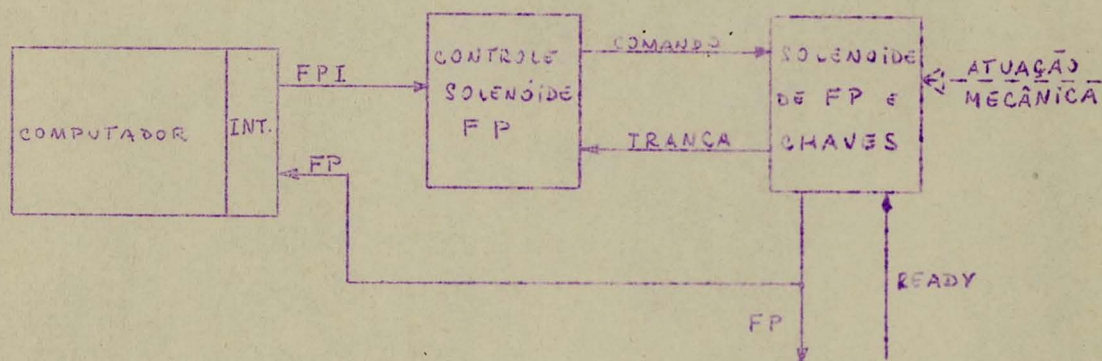
S_0, S_1, S_2 : Saídas decodificador 0, 1, 2 respectivamente

PWR: "DRIVERS" do motor.

PROTEÇÃO DE ARQUIVO

A KENNEDY 1400B possui proteção de arquivo para evitar que uma fita seja apagada acidentalmente. Quando é tentada uma operação "WRITE" sem o anel de proteção no carretel, o sinal FP (File Protect) é ativado e transmitido ao computador, enquanto que a operação é inibida na máquina. No nosso caso, decidimos operar a fita sem o anel afim de aumentar a segurança do sistema, porém os compiladores (FORTRAN, ASSEMBLER, etc.) necessitam gravar durante o processamento na mesma fita de sistema, o que foi tornado impossível pela retirada do anel! Este fato, peculiar do nosso sistema, nos obrigou a modificar o sistema de Proteção de modo que a um sinal do Interface, a gravação sem o anel fôsse possível. Este sinal, FPI (File Protect Inhibit) é decodificado no Interface e enviado diretamente ao comando do solenóide do FP na máquina, que foi modificado para esta operação. (figura 2.7)

FIG. 2.7



OPERAÇÃO

A proteção original opera do seguinte modo: Toda a vez que a máquina entra no estado "READY", o sistema de proteção é ativado energizando o solenóide, estando o anel no lugar. O solenóide retrai o pino e deixa o sinal FP zero, ficando atuado enquanto "READY" permanecer (até o próximo "REWIND"). Se não houver anel, o sistema não é atuado, sendo o FP ativado pela READY.

Para inibir a proteção, bastou usar o sinal FPI para ativar o solenóide que se tranca através da chave, mecanicamente acoplada, permanecendo até o próximo REWIND. Para facilitar a operação usou-se um transistor como "DRIVER" do solenóide.

b- INTERFACE

O INTERFACE é o dispositivo encarregado de acoplar todos os sinais de comando, estado e de dados da máquina KEN-NEDY, ao computador. Em primeiro lugar, necessita-se converter os sinais que são 0-10v em 0-5v, tornando-os compatíveis com a lógica TTL usada no 2114A. Para esta conversão adotou-se circuitos recomendados pela HP, com algumas alterações no caso de sinais 0 a - 10v.

Todo o INTERFACE foi montado em cartão apropriado (BREAD BOARD INTERFACE WITH FLAG), que já vem com lógica de FLAG e INTERRUPT de fábrica.

COMANDOS

As operações que devem ser comandadas são as seguintes:

WRITE .

READ .

RECORD GAP .

FILE GAP .

REWIND .

FAST FOWARD .

FILE PROTECT INHIBIT .

REVERSE .

Essas operações têm que ser preparadas por SOFTWARE, via uma palavra de comando. São 8 operações, o que significa 3 bits, porém usamos 4 bits, pois a lógica fica bem mais simples, enquanto se perde apenas um flip flop a mais no registrador (é relevante, pois o registrador é um TTL tipo 7475, que contém 4 flip flops). Mais um bit foi usado para diferenciar a palavra de comando de um dado de saída.

CODIFICAÇÃO DAS OPERAÇÕES

BIT 15 - O BIT 15 é o identificador da palavra de comando. Permite que as quatro bits de controle sejam carregados no Registrador de Comando.

BIT 14 - Separa as operações WRITE, GAPS, REW/FF das READ, REVERSE e EPI. Esta última foi incluída no grupo READ por razões de Software: Antes de uma operação WRITE ou READ, a máquina deve estar em READY. Para isto, a marca BOT (Beginning of Tape) deve ser localizada pelo sensor óptico enquanto a fita avança na operação "LOAD". Estando o sinal EPI atuado durante o LOAD, o solenóide se tranca assim que a marca de BOT passar pelo sensor, deixando tudo pronto para escrever.

A principal função do bit 14 é comandar o Seleccionador—chave eletrônica que substitui uma mecânica remota, para selecionar operação READ ou WRITE.

BIT 13 - Este é usado para separar, dos demais, os comandos de entrada e saída (READ ou WRITE). Se estiver em 1, significa que a operação é comando de posicionamento ou GAP (neste caso, o Flag é pôsto em 1 imediatamente).

BIT 12 - A função primordial é separar GAP de posicionamento (BIT 12 = 1 = Posicionamento). Outra função é o FPI, que é codificado juntamente com o bit 14 (leitura).

BIT 11 - Seleciona para trás/para frente (Rewind/Fast Forward) e espaços de Registro ou Arquivo (Record Gap/File Gap) e ainda, codificado com bit 14 (leitura), a Reversão (RV).

Este tipo de codificação mostrou ser a mais eficiente, vindo tanto sob HARDWARE como SOFTWARE.

Os 4 bits de comando (11-14) são carregados num registrador especial no momento de executar uma instrução de saída (OTA SC) desde que o bit 15 esteja acionado. Isto evita que o registrador seja alterado durante uma seqüência de escrita, pois a mesma instrução carrega o registrador de dados com os 6 bits menos significativos.

O diagrama lógico (fig. 2.8) mostra a implementação usada para o registrador de comando e lógica associada.

O sinal SEL passa apenas por um inversor usado como isolador, já que daí irá para a KENNEDY através de cabo.

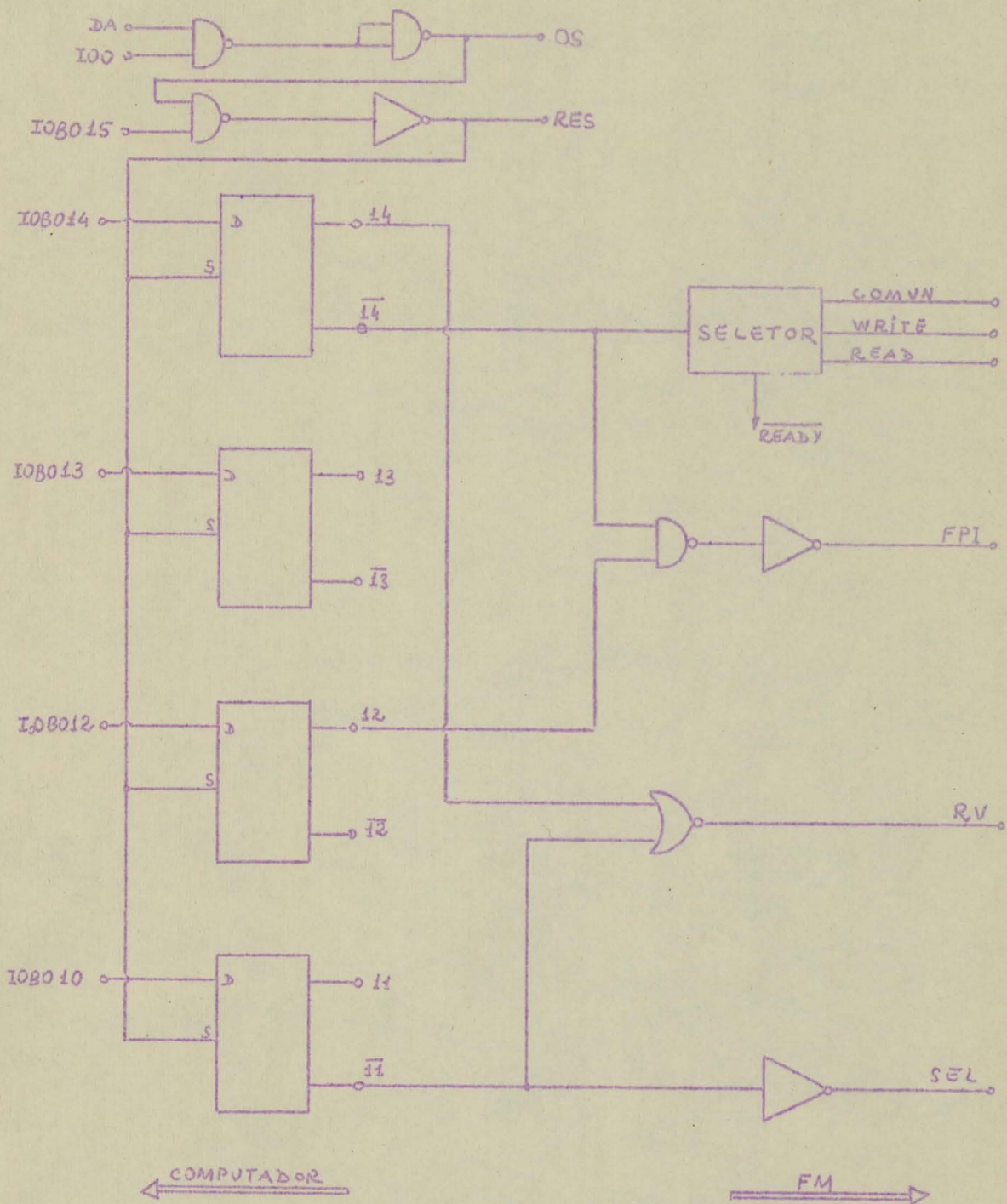
O AND dos sinais IOO e DA nos dão o pulso de carga dos registradores de saída (OS) e este, com o bit 15, resulta no carregamento do registrador de comando e no sinal RES, que é usado para limpar o flip flop "ENCODE" (mostrado adiante). Também é mostrada a lógica de sinais de comando estáticos, que se dirigem diretamente para a FM (fita magnética): RV, SEL, FPI.

SELETOR

O Seletor é uma chave eletrônica necessária para controlar o estado da máquina (WRITE-READ). Nesta, a operação é realizada ligando-se diretamente a linha SELECT COMMON com SELECT WRITE ou SELECT READ, porém no caso do Interface, isto teria de ser realizado eletronicamente e sob controle do Registrador de Comando, bit 14.

REGISTRADOR DE COMANDO

FIG. 2.8



$$SEL = 11$$

$$RV = \overline{14} \cdot 11 = \overline{(14 + 11)}$$

$$FPI = 14 \cdot 12 = \overline{\overline{14 \cdot 12}}$$

$$RES = OS \cdot IOB015 = \overline{\overline{OS \cdot IOB015}}$$

$$OS = IOO \cdot DA = \overline{\overline{IOO \cdot DA}}$$

A linha SELECT COMMON é o próprio sinal "READY" que alimenta as outras conforme o caso. Afim de implementar uma chave, examinou-se o circuito READY da EM (apêndice A) notando-se que um transistor como Seguidor de emissor, alimenta toda a linha através de um fusível de 100mA. Portanto, a corrente a se esperar nas linhas WRITE ou READ são menores que 100mA. Por outro lado, a lógica destes sinais é a seguinte (lembrando que para operação WRITE, o bit 14 = 1):

OPERAÇÕES	BIT 14	READY	LINHAS	
			WRITE	READ
Write	1	1	1	0
Read	0	1	0	1
Inibidas	1	0	0	0
	0	0	0	0

$$\text{Write} = 14 \cdot \text{Ready}$$

$$\text{Read} = \overline{14} \cdot \text{Ready}$$

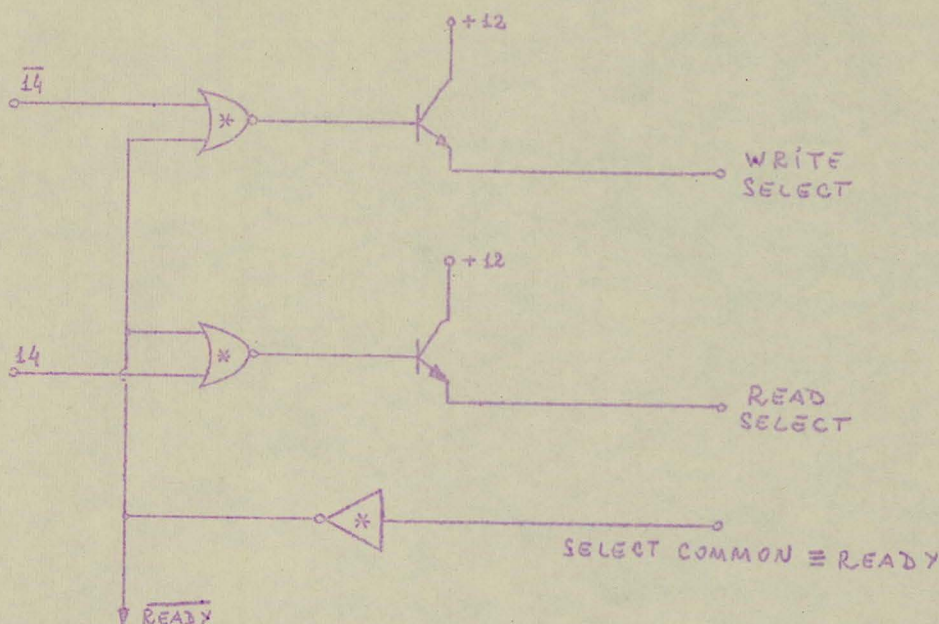
IMPLEMENTAÇÃO

Esta lógica foi realizada com transistores, já que a maioria dos sinais são de nível incompatível com TTL. As linhas WRITE SELECT e READ SELECT foram alimentadas através de seguidores de emissor da fonte do computador, enquanto a SELECT COMMON atua como controle, juntamente com as saídas 14 e $\overline{14}$ do registrador de controle. Uma saída $\overline{\text{READY}}$ (0-5v) é usada como sinal de Estado para o computador. O diagrama lógico está na figura 2.9, enquanto que o circuito completo no apêndice B.

$$\text{WRITE SELECT} = 14 \cdot \text{Ready} = \overline{\overline{14} + \overline{\text{Ready}}}$$

$$\text{READ SELECT} = \overline{14} \cdot \text{Ready} = \overline{14 + \overline{\text{Ready}}}$$

FIG 2.9



* LÓGICA DISCRETA

LÓGICA DE CONTROLE

Todo o controle é realizado sob ordem do ENCODE EF e supervisão do Control Register. O ENCODE é ativado por SOFTWARE (instrução STC SC) e pode ser limpa por SOFTWARE (CLC SC), PRESET (CRS), RES ou ainda por sinais de retorno do Interface. As operações foram divididas em dois grupos: O primeiro é formado pelas Síncronas (WRITE, READ) nas quais o ENCODE permanece ligado durante uma série de transferências (entrada ou saída); o segundo grupo é constituído pelas operações Assíncronas (Posicionamento, GAPS) onde o ENCODE é resetado imediatamente após o comando. Estas duas operações são selecionadas pelo bit 13, conforme já foi visto.

OPERAÇÕES ASSÍNCRONAS (bit 13 = 0)

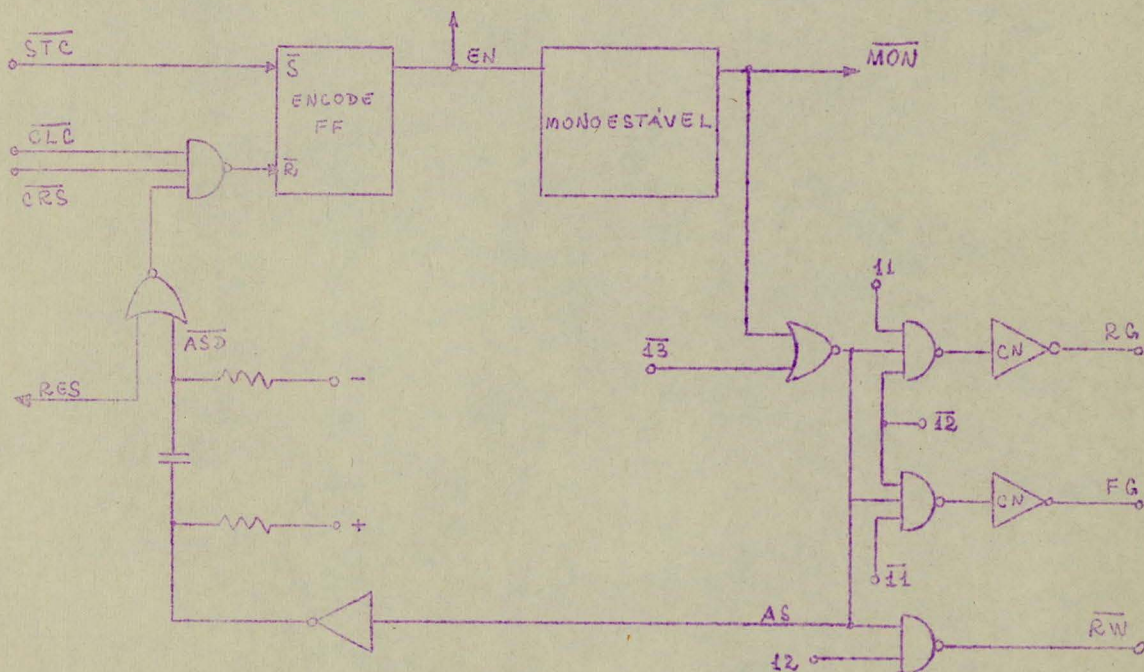
Para comandar a FM, as linhas respectivas devem ser ativadas com pulsos (0-10v largura mínima 50µs), sendo estes gerados por um multivibrador monoestável especial. Este foi ajustado para uma largura de 3ms afin de ser usado também para

retardo ao comandar LOAD/FOWARD (será visto adiante).

As 4 operações (RECORD GAP, FILE GAP, REWIND e FAST FOWARD) são decodificadas em 3, já que REWIND e FAST FOWARD e serão na EM através da linha SEL. O bit 13 é usado para ligar a saída do monostável com o decodificador, e os sinais FG e RG passam por inversores para converter o nível em 12v, enquanto que REW vai em nível TTL para a lógica de Posicionamento da EM. O pulso de comando AS é invertido e diferenciado afim de resetar o ENCODE em tôdas as operações Assíncronas. (fig. 2.10 e 2.11)

FIG. 2.10

LÓGICA DE CONTROLE:
COMANDOS ASSÍNCRONOS



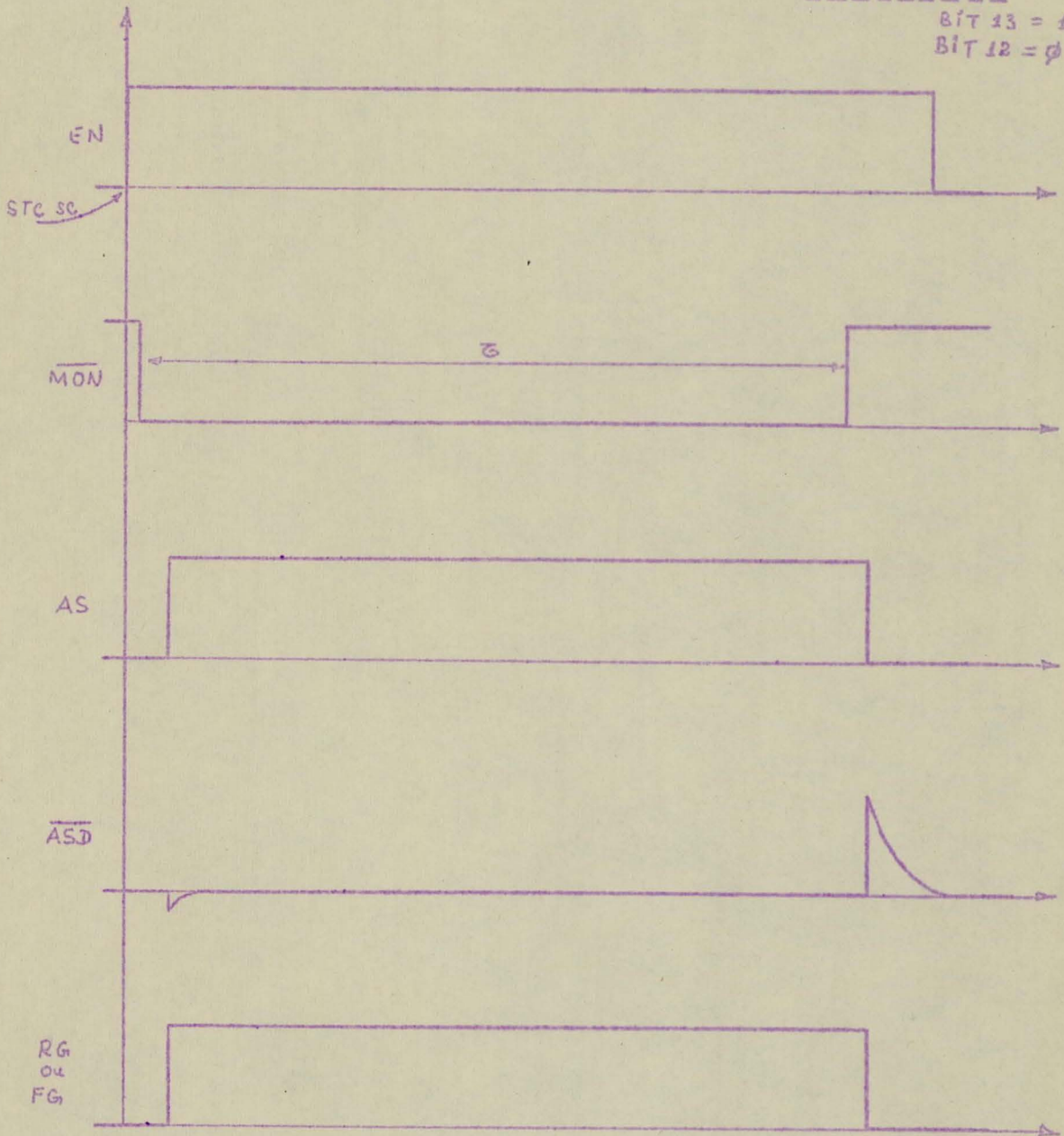
COMANDO ASSÍNCRONO

TEMPOS

CONDIÇÕES:

BIT 13 = 1

BIT 12 = ϕ (GAP)



$$\text{MON. } 13 = \overline{\text{AS}} = \overline{\overline{\text{MON}} + 13}$$

$$\text{RG} = \text{AS. } 11. \overline{12} = \overline{(\overline{\text{AS. } 11. \overline{12}})}$$

$$\text{FG} = \text{AS. } \overline{11}. \overline{12} = \overline{(\overline{\text{AS. } 11. \overline{12}})}$$

$$\overline{\text{RW}} = \overline{\text{AS. } 12}$$

OPERAÇÕES SÍNCRONAS (READ, READ FPI, READ RV)

Sob o ponto de vista do Interface, as três operações são comandadas do mesmo modo, sendo diferenciadas pela presença na FM dos sinais FPI e RV.

A leitura na FM é feita de modo contínuo, usando-se o oscilador da própria máquina. Este oscilador é atuado pelo sinal LOAD/FOWARD, e deve permanecer durante toda a leitura de um registro. Deve haver um retardo no LOAD/FOWARD para compensar a inércia mecânica do motor incremental da FM, permitindo assim comandos sucessivos de READ e READ RV (avançar e retroceder) simplificando o SOFTWARE. É obtido através do mesmo monoestável que comanda as operações Assíncronas, constando este de um flip flop de controle e um circuito de disparo com UJT. O tempo de disparo do UJT é de 3ms sendo mais estável e confiável que um monoestável de NAND GATES. O sinal L/F (LOAD/FOWARD) é de 12v lógica verdadeira e para obtê-lo usa-se um conversor de nível (fig. 2.12). A lógica do L/F permite um pulso estreito antes do "DELAY", porém é desprezado pela máquina.

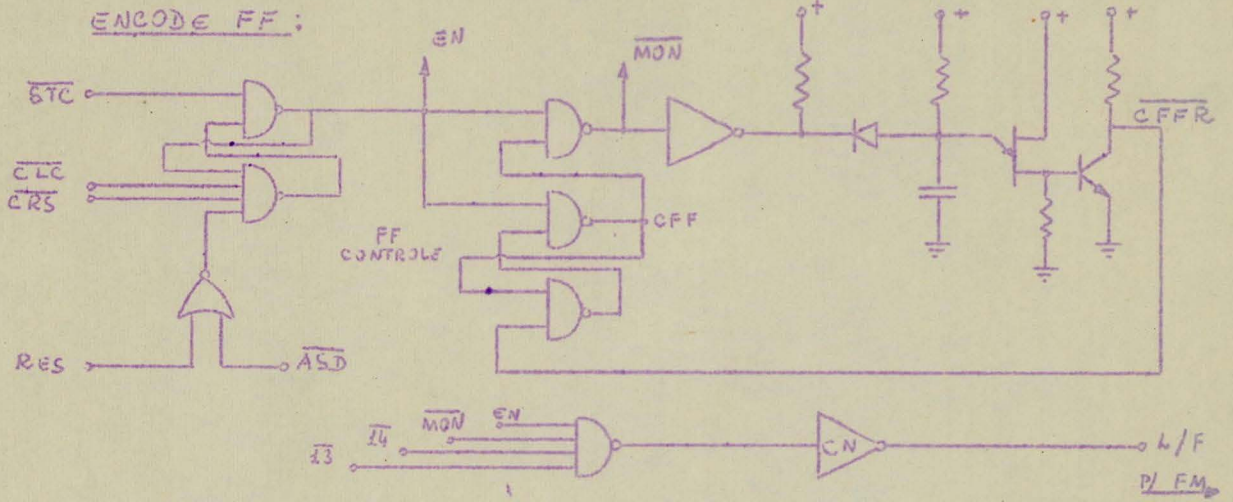
OPERAÇÃO "WRITE"

Na operação de escrita, a FM deve receber um trem de pulsos (1000 PPS) para comandar o motor incremental. Porém, este deve ser acelerado até sua velocidade síncrona, requerendo que o oscilador ao ser atuado trabalhe numa frequência menor e acelere até 1000 Hz. Outro problema é o aviso ao computador da escrita de 1 caracter (FLAG) que deve ser providenciado pelo Interface em tempo certo (após o FLAG, o computador providencia o novo dado ao Interface; portanto deve haver bom espaço de tempo entre o FLAG e o WRITE STEP).

Tendo em vista estes fatos, decidiu-se por um oscilador de Unijunção (a exemplo da própria KENNEDY) com frequência dupla e um JK flip flop divisor por 2: Com isto obtém-se um pulso de WR e um de FLAG igualmente espaçados no tempo (500us).

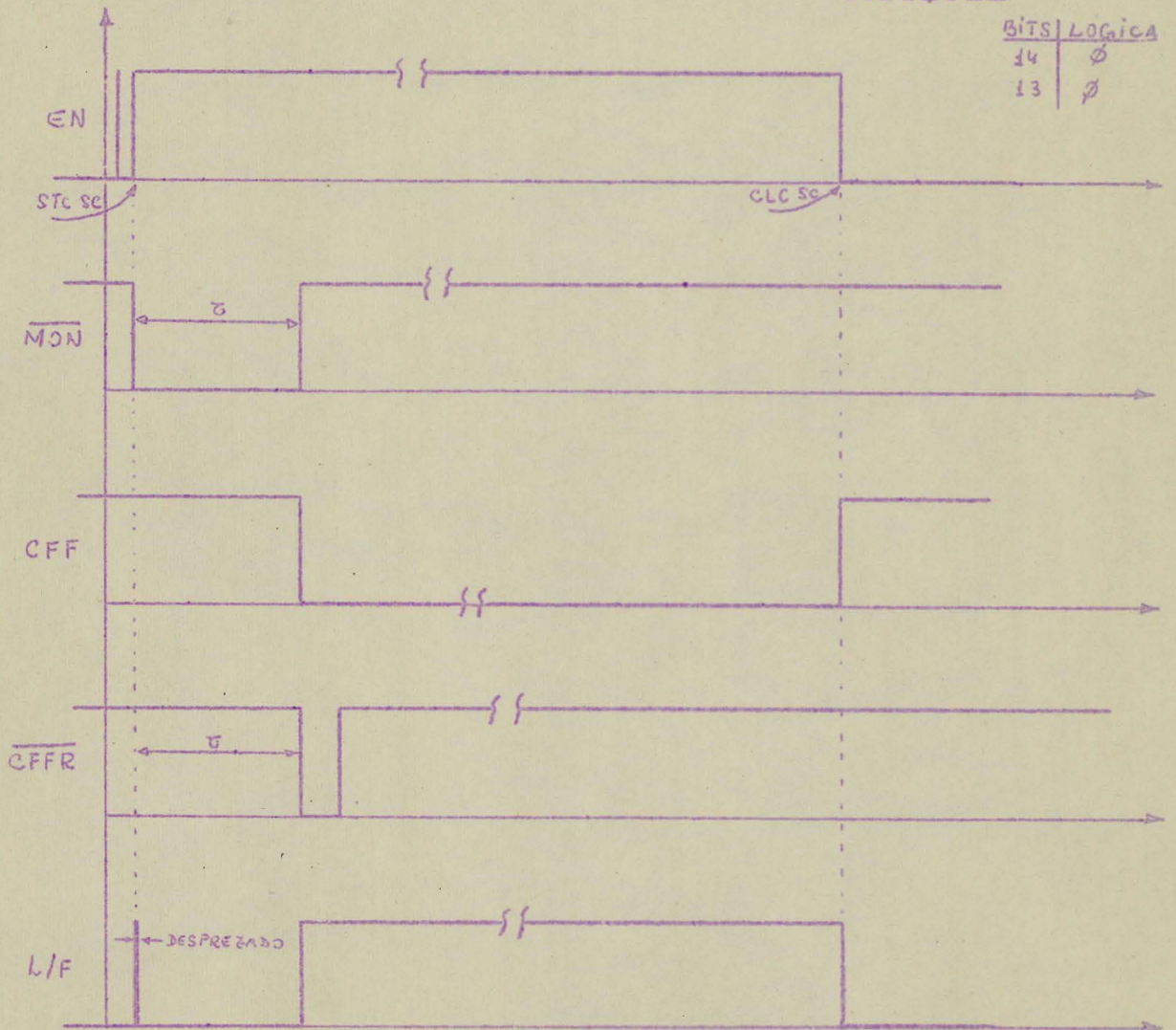
COMANDO READ

FIG. 2.12



TEMPOS :

CONDIÇÕES :

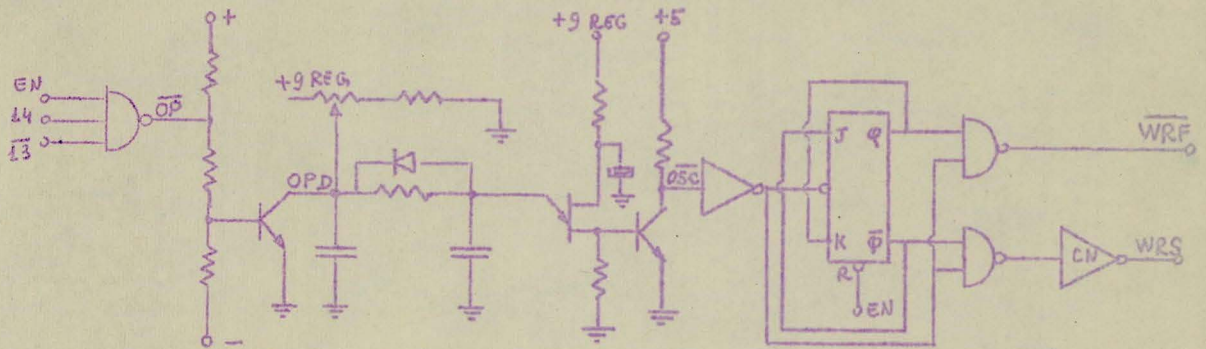


$$\overline{MON} = \overline{CFF} \cdot \overline{EN}$$

$$L/F = \overline{EN} \cdot \overline{MON} \cdot \overline{14} \cdot \overline{13}$$

FIG 2.13

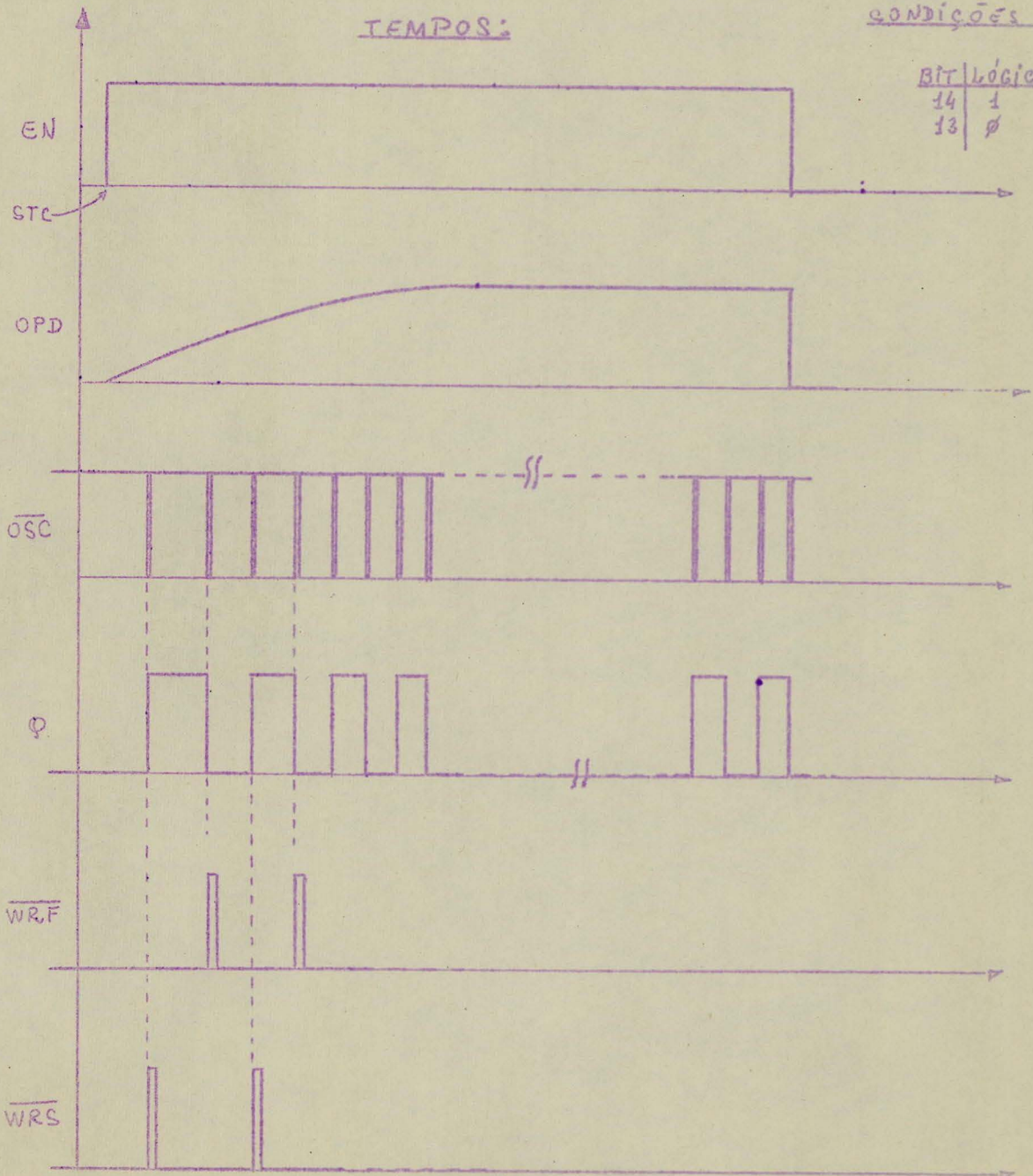
COMANDO WRITE :



TEMPOS :

CONDIÇÕES :

BIT	LÓGICA
14	1
13	0



Para melhorar a estabilidade do oscilador foi necessário regular a fonte de +12v, obtendo-se 9,1v filtrados e estabilizados. Um trimpot regula a frequência, variando a corrente de carga do capacitor (fig. 2.13).

ACIONAMENTO DO FLAG

O FLAG propriamente dito é realizado pela lógica do "BREADBOARD", bastando entregar um pulso de lógica negativa (mínimo 30ns) no terminal (TP1) do "FLAG BUFFER FLIP FLOP". A geração deste pulso (FLAG signal) é realizada por um monoestável construído a partir de 2 "NAND GATES", e sua duração é de 150ns. O mesmo pulso é usado para carregar o registrador de entrada via um TTL de alto "FAN OUT" (Buffer).

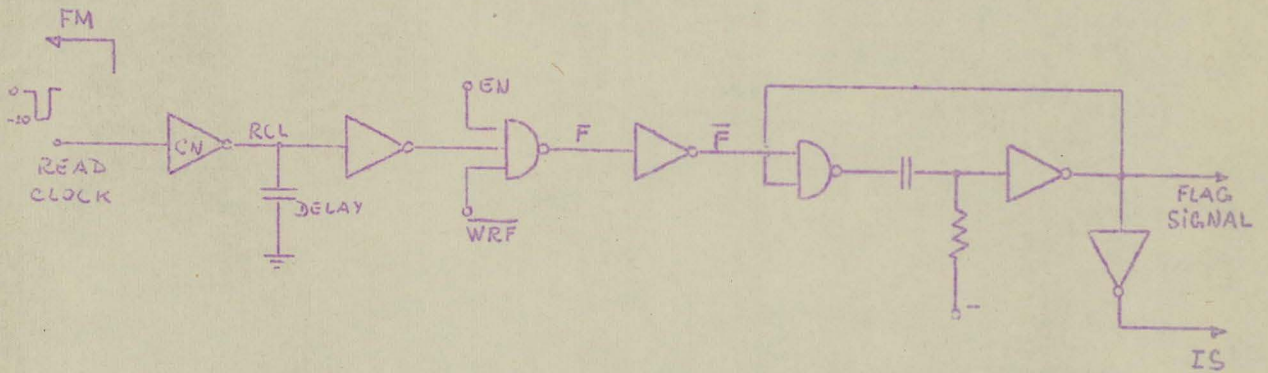
CONDIÇÕES PARA FLAG

Deve-se acionar o FLAG, na leitura e na escrita, informando ao computador do término de uma transferência. O FLAG da operação WRITE é gerado pelo seu "CLOCK" (\overline{WRF}), porém este pulso é demasiado largo para acionar o FLAG (dura cerca de 5 ciclos do computador) e portanto deve ser conformado pelo monoestável. Para a operação de escrita, existe o "READ CLOCK" proveniente da FM que é um pulso de 8us lógica 0 a -10v, devendo ser transladado para TTL, conformado e retardado apropriadamente (vide Registrador de Entrada).

As operações READ FPI e READ RV, usem o FLAG do mesmo modo que a READ.

As operações Assíncronas não necessitam necessariamente de teste de FLAG, visto que todas tem sinais especiais de Estado para verificá-las (são operações lentas), porém para o SOFTWARE é interessante que o FLAG seja liberado imediatamente após uma operação Assíncrona (pela facilidade de usar apenas uma subrotina). Para implementar essa condição, decidiu-se comandar o FLAG toda a vez que o ENCODE FF fôsse resetado.

LÓGICA DE ACIONAMENTO DO FLAG



$$F = RCL + WRF + \bar{EN} = \overline{\overline{RCL} \cdot \overline{WRF} \cdot EN}$$

$$IS = \overline{FLAG\ SIGNAL}$$

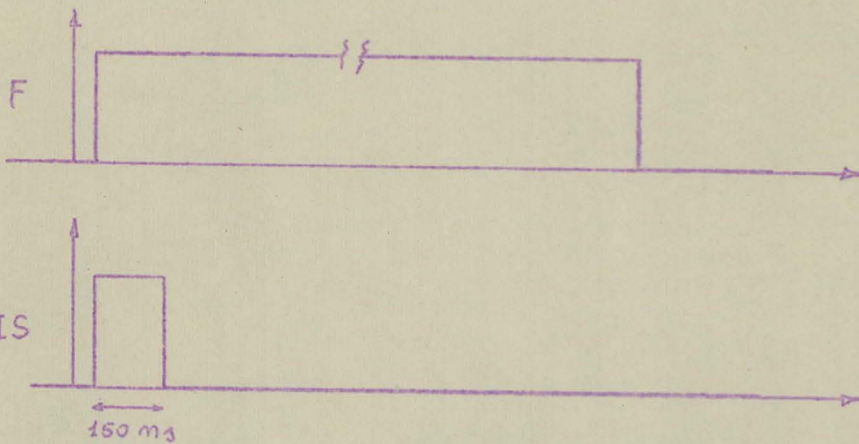
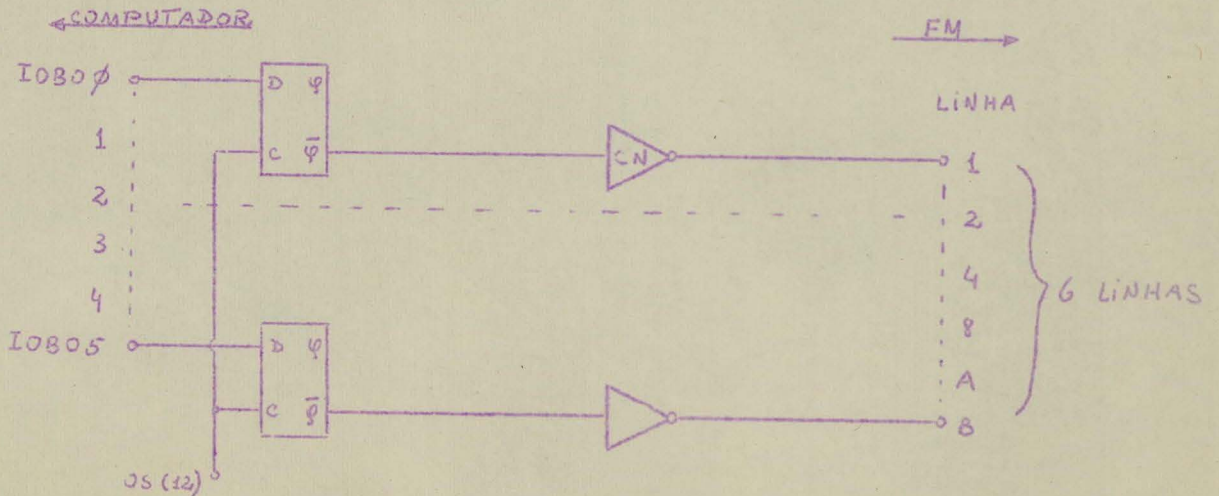


FIG. 2.15

SAÍDA DE DADOS :



Dêste modo resolveu-se 2 problemas:

- 1- Liberar o FLAG em tôdas as operações Assíncronas (já que estas resetam o ENCODE imediatamente).
- 2- Evitar que o programa ficasse eternamente em "WAIT" caso o ENCODE fôsse limpado durante uma operação READ ou WRITE pelo PRESET.

Em resumo; 3 sinais devem causar FLAG:

READ CLOCK: RCL

WRITE FLAG: WRF $\Rightarrow F = RCL + WRF + \overline{EN}$

ENCODE NOT: EN

Na implementação usou-se um GATE de 3 entradas (pô-
dia-se usar de 4) por uma questão de disponibilidade na placa
(fig. 2.14).

FLUXO DE DADOS

SAÍDA DE DADOS

Tanto a entrada como a saída de dados é realizada com registradores, para permitir operação em "interrupt". O registrador de saída (6 bits) é constituído de 6 flip flops tipo PH (POLARITY HOLD). O sinal de controle OS provém de um TTL de alta "FAN OUT" (7440) já que cada entrada tem 2 unidades de carga (12 total).

O sinal OS é obtido a partir de I00 com DA como já foi visto no registrador de controle. São usadas as 6 linhas menos significativas do "BUS" do HP 2114A (Bits 0 a 5) e as saídas dos flip flops são acopladas à conversores de nível que adaptam os sinais à FM. Devido à natureza do processo, os sinais podem ser acoplados em DC (sem Gates), pois por ocasião do WRITE CLOCK êles são amostrados e gravados (NRZI).

As linhas do computador devem ser terminadas com resistores para drenar a corrente de entrada dos TTL. No caso dos flip flops, são duas unidades de carga o que implica em 680Ω (ver fig. 2.15).

ENTRADA DE DADOS e SINAIS DE ESTADO

A saída da KENNEDY são pulsos de lógica negativa (0 - 10v) sincronizadas com o "READ CLOCK", também negativo, que comanda o "FLAG".

A saída do monoestável é invertida para ser usada como sinal de controle do registrador de dados (IS).

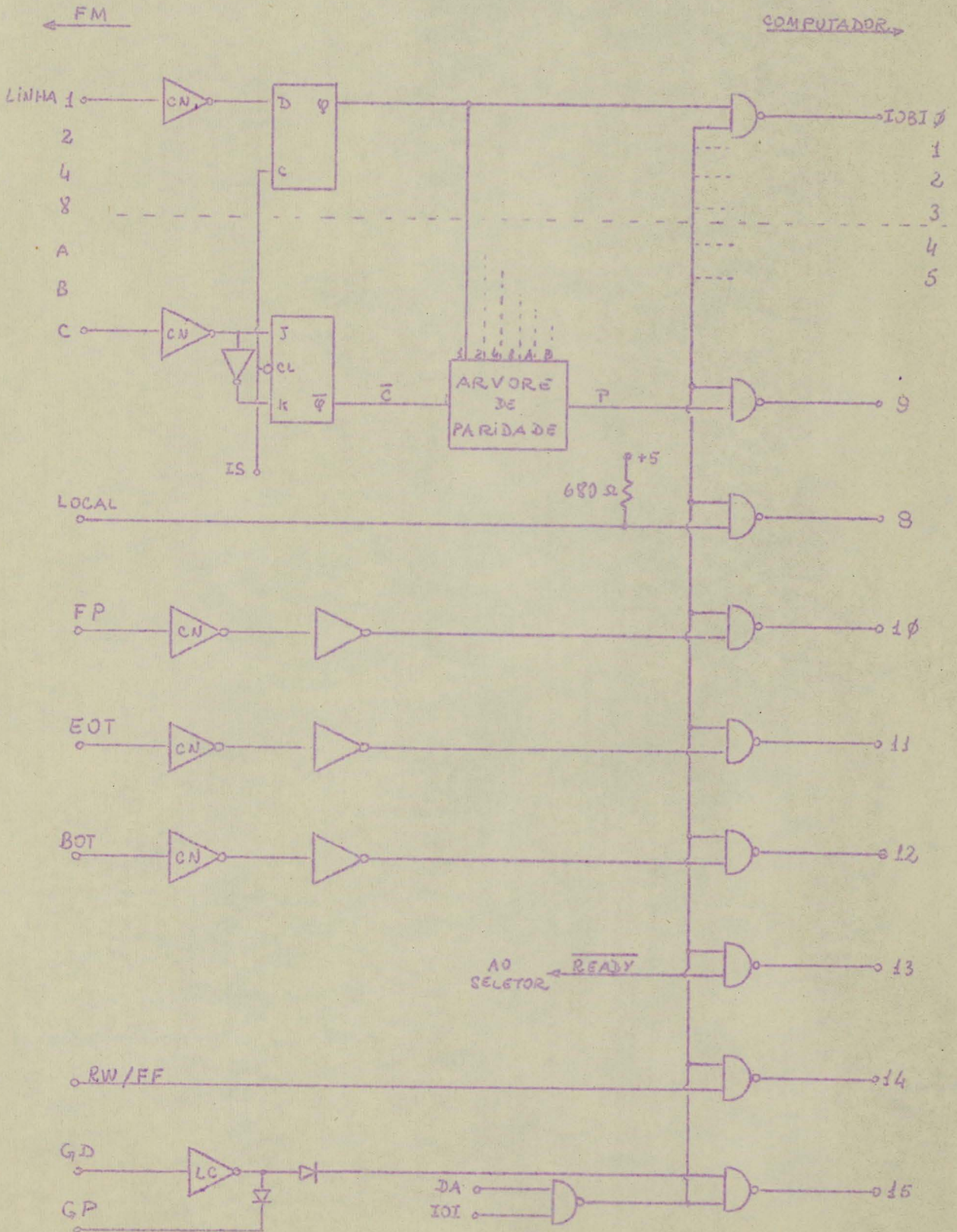
Este registrador é constituído de 6 flip flops idênticos aos usados na saída (PH) mais um JK para o bit de paridade (linha C). Conversores de nível acoplam os sinais da FM ao registrador, e o controle (IS) sofre um retardo tal que os sinais sejam amostrados em tempo certo. Este retardo é obtido no conversor de nível do "READ CLOCK" sendo mostrado na lógica do "FLAG" (fig. 2.14).

As saídas verdadeiras dos flip flops alimentam a árvore de paridade e os "AND GATES" (FARCHILD CTuL 9956) especiais para entrada ao "BUS" do computador. Todos estes "LINE DRIVERS" são comandados pelo "AND" dos sinais IOI e DA, realizado com um "AND GATE" da mesma família (CTuL) (fig. 2.16).

ÁRVORE DE PARIDADE

O cheque de paridade foi implementado através de uma árvore de "XOR GATES", constituídos de circuitos "AND-or-INVERT" e inversores (fig. 2.17). A paridade usada é a ímpar, como já foi discutido; porém a saída da árvore é par (complemento). Foi feito dessa forma porque para o SOFTWARE qualquer bit de "STATUS" em 1, corresponde a erro.

ENTRADA DE DADOS E ESTADO



A árvore é acoplada a um "LINE DRIVER" como as seis linhas de sinal, porém a linha acionada é a do bit 9, por razões de "SOFTWARE".

SINAIS DE ESTADO (STATUS)

Os sinais necessários para manter o computador ao par do Estado da FM são:

1) LOCAL: Refere-se à posição da chave de seleção própria da FM: a linha é posta à massa quando a chave está em REMOTE.

2) FP (File Protection): É acionado pelo Solenóide de FP, e a sua lógica é 1 quando a FM é posta em WRITE sem o anel de proteção.

3) EOT (END OF TAPE): Sinal que indica a presença da marca refletora de fim de fita.

4) BOT (Beginning of TAPE): Análogo ao anterior; indica marca de início de fita.

5) READY: Sinal proveniente do Interface (Seletor WRITE/READ) e permite verificar se a FM está pronta para realizar uma operação de leitura ou Escrita. Usa-se o complemento, ou seja, se READY está em zero, a fita está pronta.

6) RW/FF: Este não existia na KENNEDY 1400R originalmente, sendo implantado juntamente com a lógica de controle das operações REWIND e FAST FORWARD, por isso seus níveis 0 e 1 são de TTL. O nível 1, significa "Em Operação".

7) GAPS: Dois sinais provenientes da FM indicam presença de GAP: O primeiro atua durante a leitura (GAP DETECT) e vai para 1 enquanto um registro é lido. O outro é o GAP IN PROCESS, que vai para zero durante a confecção de um Record ou File GAP.

Como estes sinais são exclusivos por construção, foi possível deixá-los ocupar uma única linha mediante um GATE AND feito com diodos por conveniência. O sinal GAP DETECT foi convertido para nível TTL e invertido; e o AND dos dois é o sinal GAP que vai para 1 na detecção de GAP durante a leitura ou após a conclusão de GAP na escrita.

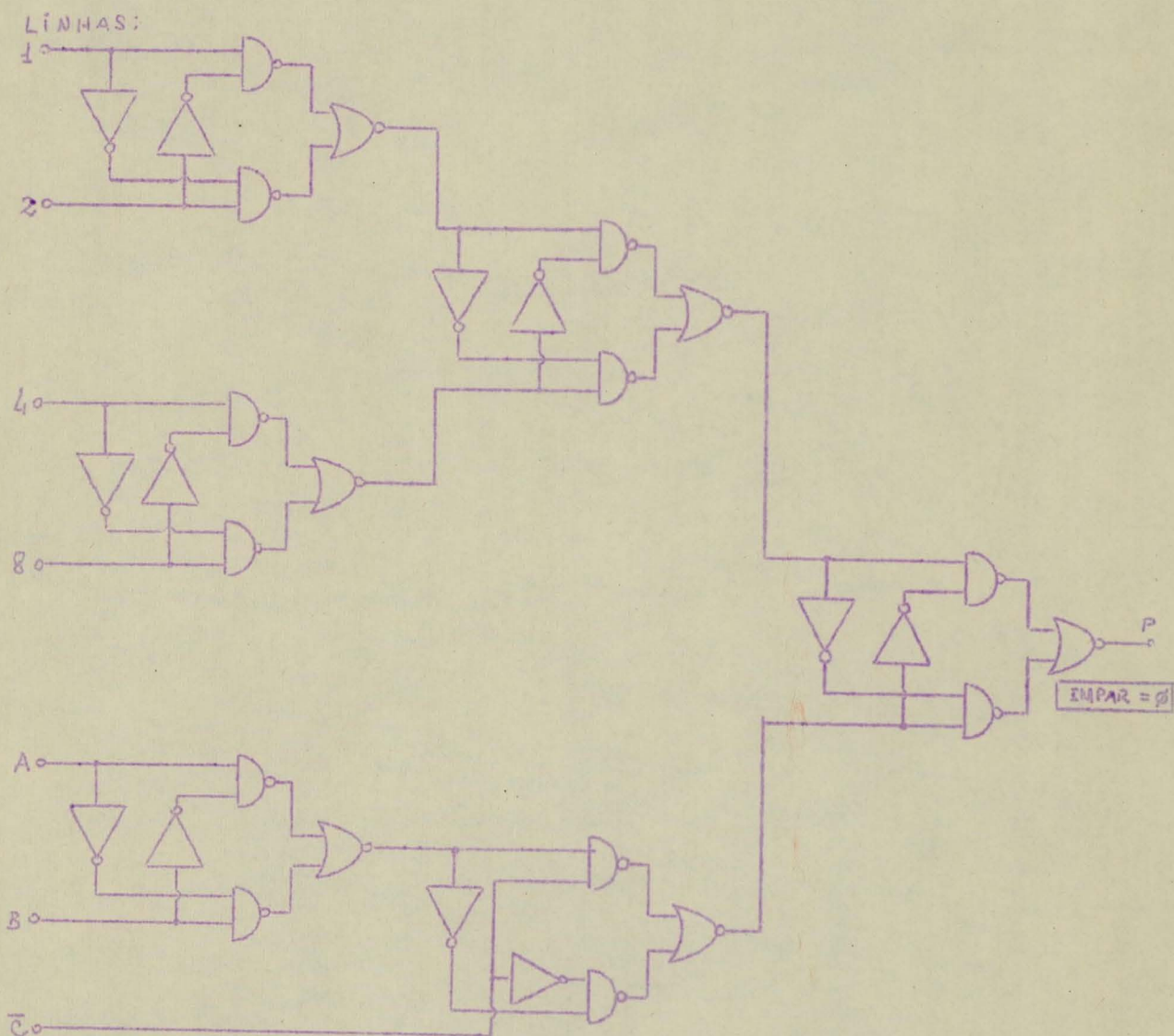
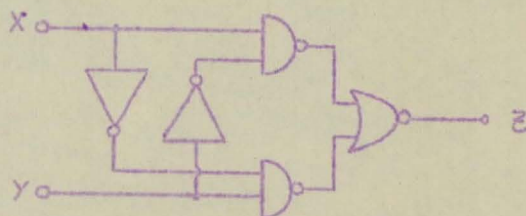
Lógicamente:

$$GAP = \overline{GD} \cdot GP$$

GD : GAP DETECT

GP : GAP IN PROCESS

Todos estes sinais de Estado são acoplados ao "I/O BUS" do computador através de CTL como os sinais de dados e paridade, cada um deles ocupou uma linha, ficando o GAP com o bit 15 e os outros em seqüência decrescente. Este arranjo facilita o "SOFTWARE". A lógica dos sinais é tal que todos devem estar em zero durante uma leitura (fig. 2.16).

ARVORE DE PARIDADEDO REGISTRADOR
DE ENTRADACADA XOR:

$$\bar{Z} = X \cdot \bar{Y} + \bar{X} \cdot Y = X \oplus Y$$

ENTÃO,

$$\bar{P} = ((A \oplus B) \oplus C) \oplus ((1 \oplus 2) \oplus (4 \oplus 8))$$

ENTÃO, SE DER IMPAR $P=1$ ou $\bar{P}=0$

3- SOFTWARE

O "SOFTWARE" para a operação da fita pode ser dividido em 3 etapas:

- I- "DRIVERS" e "LOADER".
- II- MONTADOR.
- III- SISTEMA.

I- "DRIVERS" e "LOADER"

Idealmente, os "DRIVERS" deveriam ser completamente compatíveis com o sistema usado pela HP, porém as características peculiares do sistema forçaram uma constituição diferente.

A divisão da fita pelos marcadores requer um método de posicionamento especial (usando TAGS), pois esta operação requer 2 etapas:

- Posicionamento entre os marcadores.
- Posicionamento entre os Registros.

Para o SOFTWARE o mais fácil foi adotar enderços para os registros (TAG) que permite localizá-lo segundo os dois critérios. O "TAG" é uma palavra de 16 bits sendo os primeiros 10 o número do registro na divisão considerada, enquanto que os últimos 6 bits representam o número da divisão. Portanto, a fita pode ser dividida em 64 divisões, e cada uma delas conterá um máximo de 1024 registros.

Formato de um TAG

Bits	15	10	9	0
	Nº Divisão		Nº Registro	

No início de cada divisão deverá haver sempre 1 registro de identificação com o número do arquivo (o número de registro é Zero).

MARCA de FIM de ARQUIVO

Normalmente o fim de arquivo em fita magnética é um caracter contendo o número 17_8 . A KENNEDY 1400R escreve esta marca na fita durante a execução de um "FILE GAP". Porém, neste caso particular decidiu-se adotar uma marca diferente: Em vez de escrever apenas um caracter, escreve-se um REGISTRO especial, que contém endereço e marcas que tornam fácil e segura sua determinação.

REGISTROS

Existem 4 tipos de registros:

- Registro Binário
- Registro Binário Absoluto
- Registro ASCII
- Registro EOF (END OF FILE)

De maneira geral, todos os registros contêm marcas, palavra que indica o tamanho do registro (SIZE), TAG e Dados.

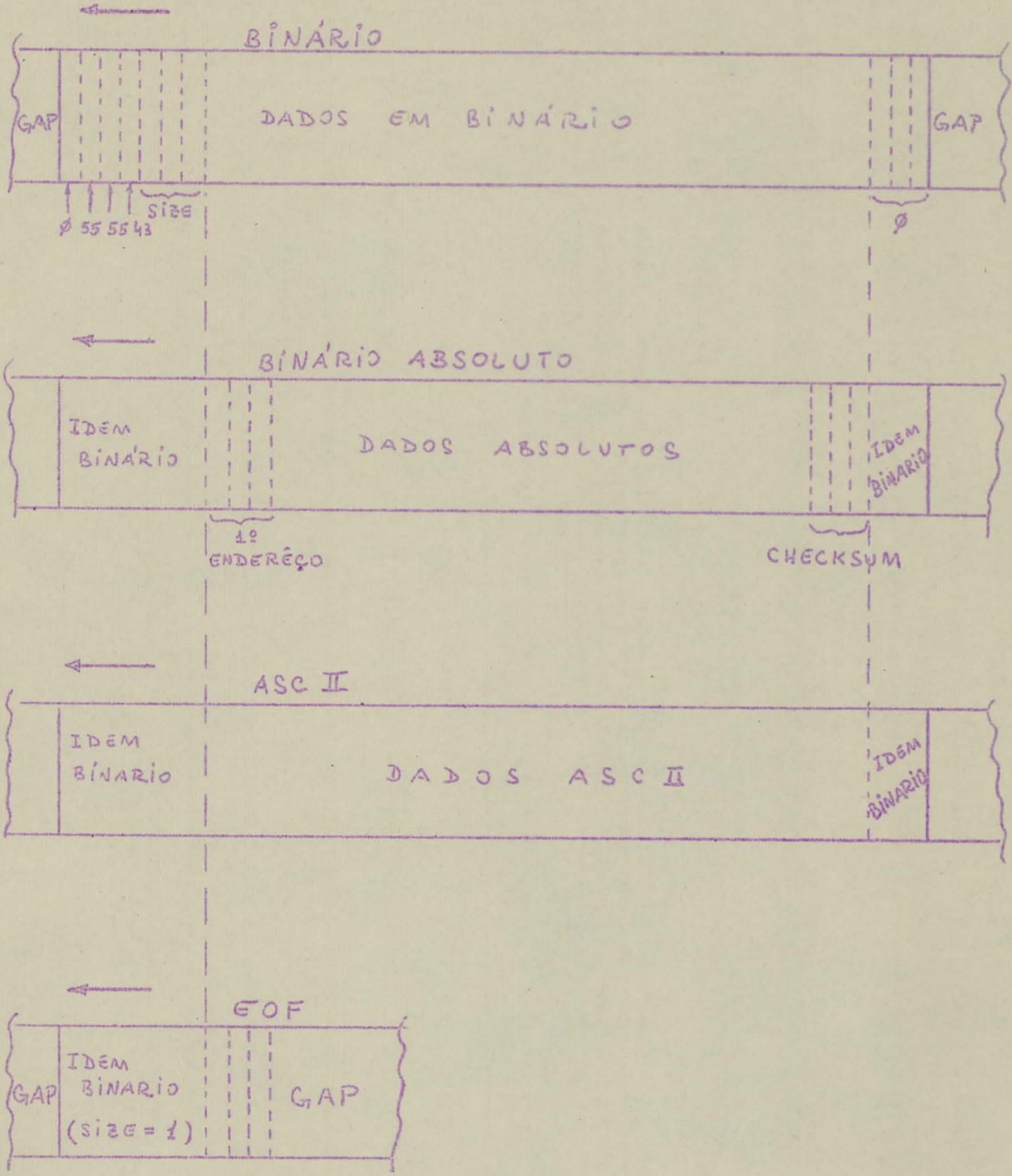
Tôdas as informações de 16 bits são divididas em 3 caracteres de 6 bits cada, de modo que os bits 5 e 10 são gravados duas vêzes. As marcas de identificação no início eliminam a possibilidade de confusão dos registros com outros caracteres da fita (17_8 , LCC). As marcas são 3 caracteres consecutivos: 55_8 , 55_8 e 43_8 . No início do Registro é escrito um caracter nulo (0) e, no fim, três nulos que são desprezados na leitura. Sua função é proteger o registro dos transitórios do sistema de posicionamento.

No Registro Binário Absoluto existem mais duas palavras especiais: Primeiro endereço e "CHECKSUM", sendo este a soma de tôdas as palavras exceto SIZE, TAG e CHECKSUM.

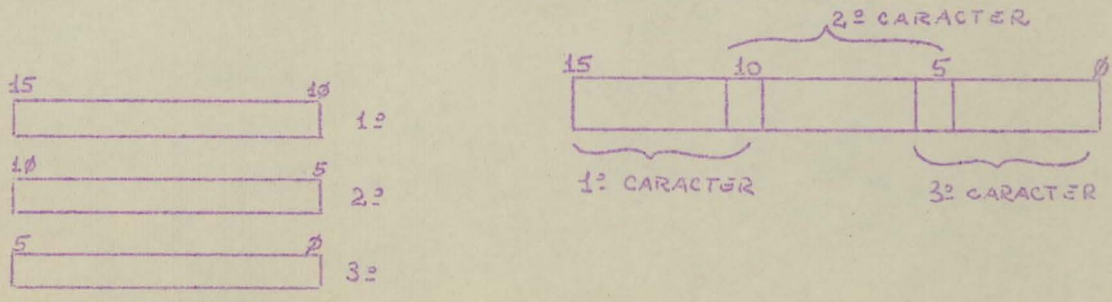
O registro ASCII é diferente do sistema de fita de papel, porque aqui não têm sentido comandos como CR, LF ou RBOUT.

FIG. 3.1

REGISTROS



DESDOBRAMENTO:



Os dados ASCII transcritos para a FM já vem isentos de RBOUF (ação dos "DRIVERS" de fita de papel) enquanto que CR e LF são substituídos por um "SIZE" de maneira análoga ao registro Binário, porém este não é transmitido ao "BUFFER" dos ASCII.

O registro EOF caracteriza-se pelo tamanho (1 palavra), sendo no restante igual ao Binário (fig. 3.1)

DRIVER

O DRIVER para operação da FM foi escrito segundo os padrões da HP (SIO DRIVERS), tendo porém algumas alterações devido a estrutura peculiar deste Sistema. As diferenças estão no Posicionamento e nos bits de Estado. Na chamada para Posicionamento, o Registrador A deve conter o endereço do Registro após o qual será posicionada a FM, enquanto que o registrador B poderá ter qualquer valor diferente de -1 no posicionamento Normal (para ler ou para escrever com o Anel de Proteção). Para posicionar antes de uma escrita sem o anel de proteção, o registrador B deverá ser igual a -1 .

Outra operação possível é o retrocesso de 1 registro quando o registrador A é igual a Zero . As chamadas para o DRIVER, são:

1. READ ou WRITE

	LDA <tamanho>	{ CARACTERES (ASCII) $A > 0$ PALAVRAS (BIN) $A < 0$ $A = 0$ chamada ignorada
	LDB <Endereço "BUFFER">	
P	JSB 107B, I	
P+1	{	
	OCT 0 (READ)	
	OCT 1 (WRITE)	
P+2	retorno EOT/ BOT/ EOF	
P+3	Erro de paridade	
P+4	retorno Normal	

(A contém caracteres lidos e B os bits de Estado)

2. POSITION

	LDA TAG	{ TAG= Endereço na fita se $A = 0 \Rightarrow$ retrocede 1 registro
	{CLB	
	{CCB	
P	JSB 107B, I	
P+1	OCT 4	
P+2	<retorno EOT>	
P+3	<erro Paridade>	
P+4	<Retorno Normal>	

3. WRITE EOF ou GAP

P	JSB 107B, I
P+1	{
	OCT 2 para EOF
	OCT 6 para GAP
P+2	<retorno EOT/BOT>
P+3	<não é usado >
P+4	<Retorno NORMAL >

4. REWIND ou STATUS

P	JSB 107B,1	
P+1	{ OCT 3 OCT 5 OCT 7 }	REWIND STATUS
P+2	<Retorno Normal>	

(B contém os bits de Estado)

O DRIVER transmite sempre o registro de endereço seguinte ao último transmitido. Com isto evita-se erros devido à perda de um registro, pois a subrotina de procura (SEARX) pára quando não encontra o registro endereçado.

Internamente, existe uma palavra (TAG) que contém o endereço do último TAG lido (ou escrito). Quando a FM está READY, a subrotina SEARX calcula o posicionamento em função desta palavra. Por outro lado, não estando READY, o registro mais próximo é lido e em função de seu TAG procede-se o posicionamento.

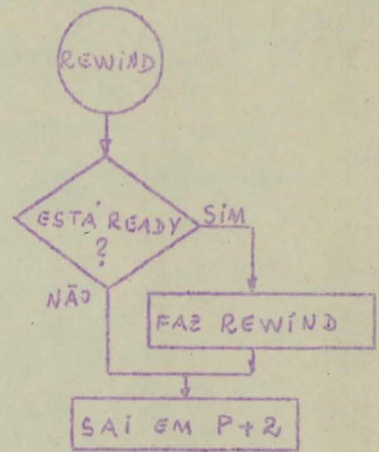
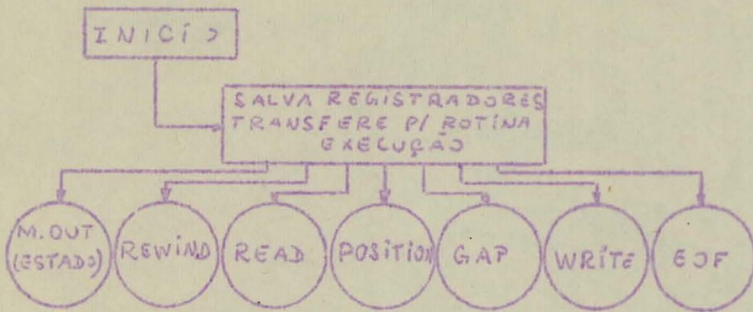
Nas operações que envolvem Leitura (READ ou POSITION), a fita só é parada nos GAPS, sendo usado o sinal GAP proveniente do Interface (BIT 15). Após a leitura de um registro, o DRIVER espera o sinal de GAP para então desligar o ENCODE FF do Interface (CLC SC) (fig. 3.5).

Sempre que o DRIVER é chamado, o Estado da fita é testado:

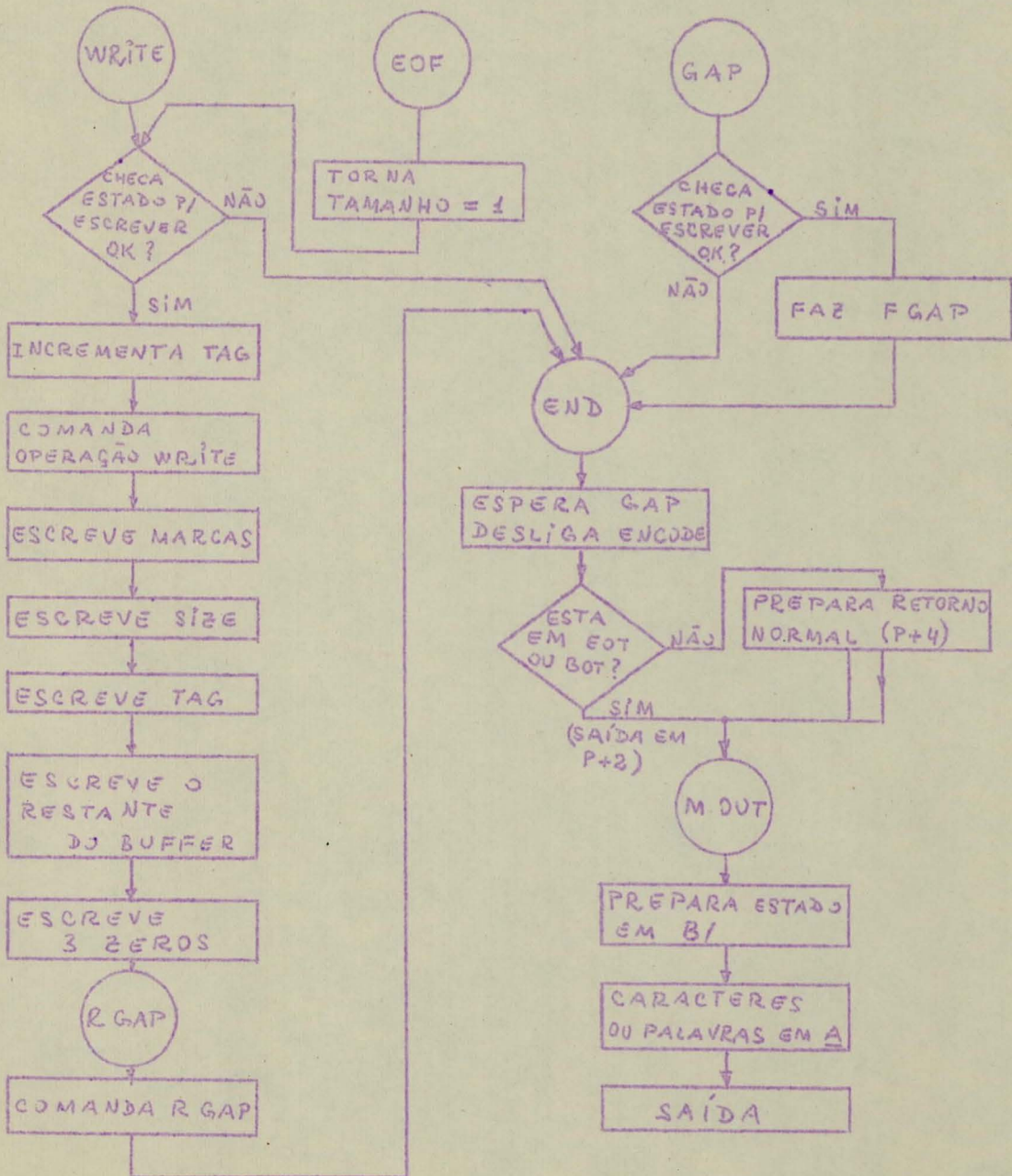
Não estando em GAP, o DRIVER espera que se realize esta condição, e o mesmo acontece se estiver em REWIND ou FAST FORWARD.

Os restantes são divididos em estados para WRITE e estados READ ou POSITION, conforme a tabela:

MT SIO DRIVER
DIÁGRAMA DE BLOCOS



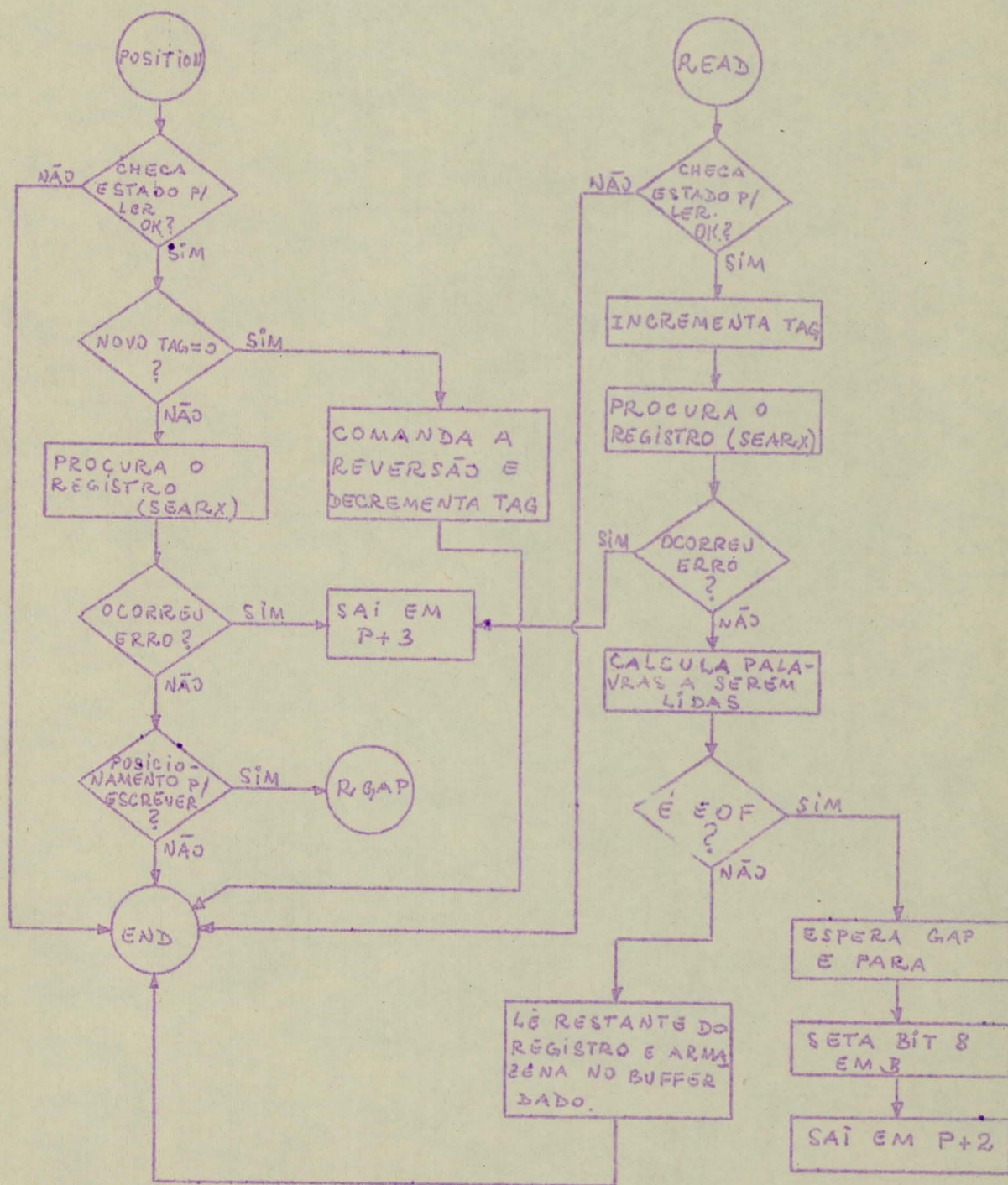
ROTINAS : WRITE, EOF, REWIND E GAP:



MT SIO DRIVER
DIAGRAMA DE BLOCOS (CONTINUAÇÃO)

FIG 3.3

ROTINAS READ E POSITION:



W R I T E	R E A D
EOT	EOT
FP	
LOCAL	LOCAL
<u>READY</u>	
BOT	

Para posicionar, READY e EOT são desprezados, pois a fita pode ser posta em READY (operação realizada pela KENNADY) passando por um BOT. FP não tem sentido em READ.

Caso o Estado não esteja correto, a chamada é abortida saindo em P+2 ou P+4, com Bits de Estado em B e A=0. Se o estado for BOT ou EOT; sai em P+2, do contrário sai em P+4.

No posicionamento para escrever (B= -1 na chamada) após achar o registro certo, a fita escreve um RECORD GAP. Isto é importante para evitar a perda de algum caracter (e consequente erro no Registro) por ocasião da leitura do registro considerado.

A operação "Retrocesso de 1 Registro" (A=0 na chamada) consta do comando RETROCESSO ao Interface, durante tempo suficiente para voltar um registro de 100, palavras e o TAG é decrementado. Na próxima operação de leitura, o registro retrocedido é procurado pela SEARCH, sendo então lido. Este tipo de operação foi necessário, já que a leitura para trás não é segura.

READ

Após ter sido encontrado o registro considerado, procede-se à análise do SIZE : Se for igual ou menor que o dado, o número de palavras lidas será igual a SIZE, se for maior é lido o número dado (registrador A na chamada). Em seguida, o restante do registro é carregado no BUFFER dado (o endereço estava em B na chamada).

Se ocorrer algum erro de paridade durante a leitura do registro, é executada a rotina PARIT (parte da SEARX), que retrocede a fita e tenta ler novamente o registro num máximo de 4 vezes. Se na quarta vez ocorrer erro, é setado bit de paridade B, retornando pela saída P+3. Não ocorrendo erro, executa-se a rotina END, terminando a chamada normalmente (P+4).

REWIND

Na operação REWIND, apenas é testado o sinal READY : Sómente se a magnética estiver READY a operação é executada (para evitar que a fita escape do carretel). Retorna em P+2.

GAP

Antes de comandar EG, o DRIVER examina se o Estado da máquina (mesmo teste de WRITE). Só executa se tudo estiver certo, passando depois à rotina END de saída.

WRITE e EOF

Em EOF, o tamanho do Registro (SIZE) é tornado 1, realizando-se depois uma operação WRITE normal.

Inicialmente, a condição de estado é verificada, sendo o TAG incrementado logo após. A operação WRITE deve ser pre-setada no Interface através do código respectivo, seguido do comando STC SC que põe a fita em movimento. A seguir, são escritos em seqüência as marcas, o SIZE, o TAG e o restante do "BUFFER". Finalmente, os zeros de segurança são escritos, sendo em seguida comandado um RG que também termina a operação resetando o ENCODE FF do Interface. A rotina END é executada procedendo ao retorno do DRIVER.

SUBROTINA SEARX (PROCURA DE REGISTROS)

CHAMADA:

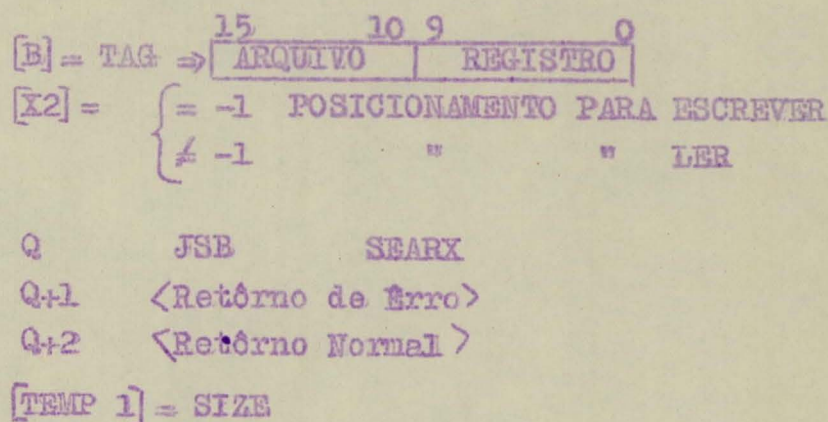


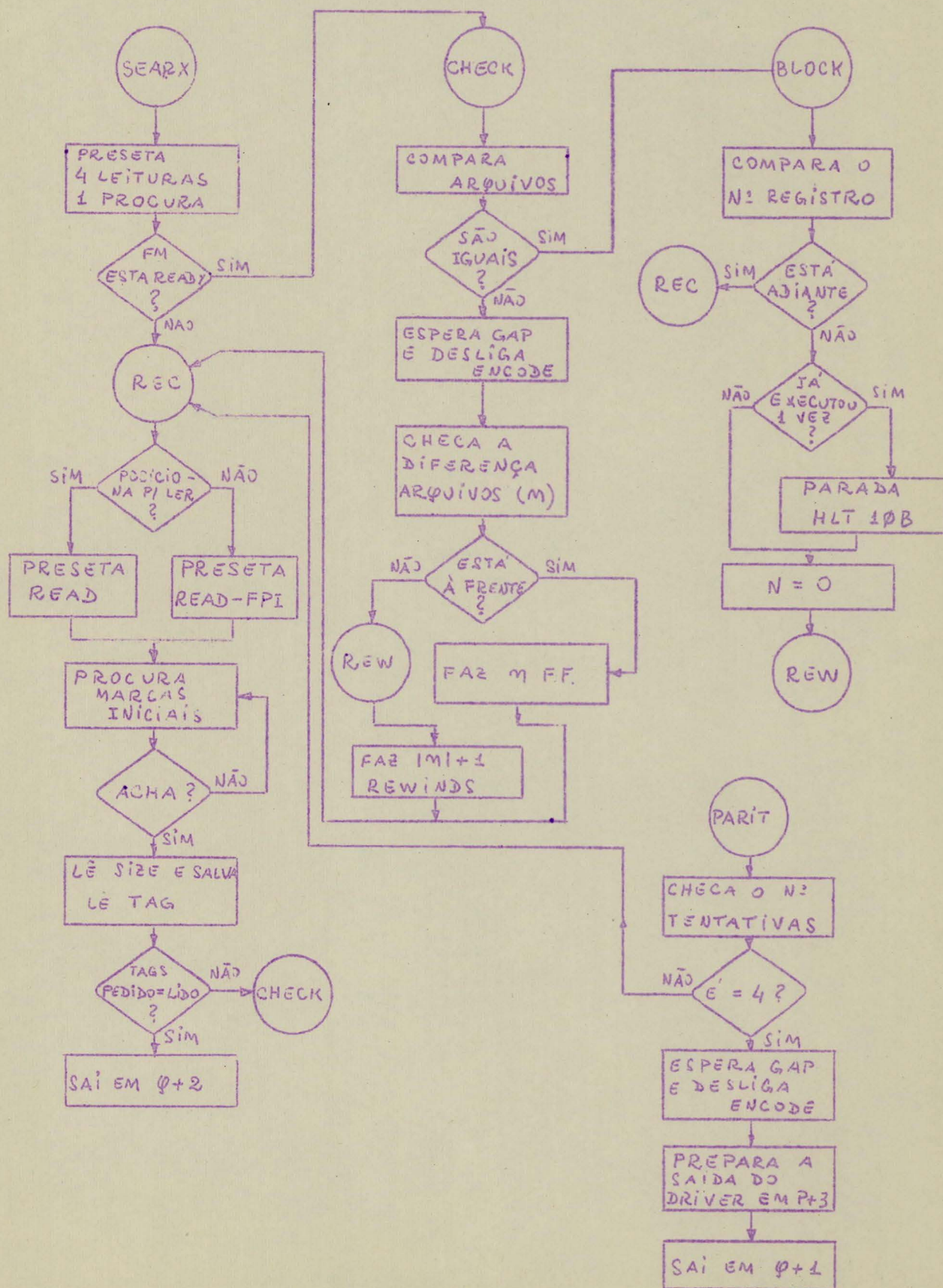
figura 3.4

SEARX: CHAMADA

A subrotina de posicionamento (SEARX) é usada tanto pelo SIO DRIVER como pelo LOADER, diretamente. A seqüência de chamada está na fig. 3.4. O registrador B leva o TAG completo (ARQUIVO e REGISTRO) e num endereço, definido como X2, existe a constante indicadora do tipo de posicionamento. O retorno de erro (Q+1) é usado depois que a leitura de um registro foi tentada 4 vezes, sempre com erro. Na saída normal (Q+2) a fita está em movimento e a última palavra lida foi o TAG, enquanto que o SIZE está armazenado no lugar definido como TEMP 1.

SEARX

Ao iniciar a subrotina, B contém o endereço pedido, enquanto em TAG está o atual. Neste ponto, um teste de estado é feito: Se a FM estiver READY, o cálculo a seguir é feito entre TAG pedido e TAG atual. Se não estiver, um registro é lido, atualizando o TAG atual, sendo então feito o cálculo. O "LABEL" REC marca o início da operação de leitura de registro. Aqui, primeiro é checado o tipo de posicionamento, e depois o código respectivo é presetado no Interface. A seguir, as marcas iniciais de um registro são procuradas; sendo encontradas, lê-se SIZE (armazenando-se em TEMP 1) e o TAG (ver diagrama em blocos, fig. 3.5).

SUBROTINA SEARXDIAGRAMA EM BLOCOS

O TAG atual é comparado com o pedido, se forem iguais, a subrotina retorna em Q+2. Sendo diferentes, passa-se a calcular a diferença em duas etapas: Primeiro, compara-se os 6 bits mais significantes que são o número de Arquivo. Se forem diferentes (está em outro arquivo) comanda-se tantos REWIND ou FAST FORWARD quanto necessários e lê-se novamente 1 registro. Esta operação pode-se repetir até que o arquivo seja encontrado. Tendo encontrado o arquivo, compara-se os 10 bits menos significativos dos TAGS; caso o registro se encontre para trás comanda-se 1 REWIND e passa-se novamente a ler. Esta operação, só é realizada 1 vez: não encontrando após duas buscas é provável que o registro em questão esteja estragado. Caso o registro procurado esteja à frente, o seguinte é lido e assim sucessivamente até encontrá-lo.

A rotina PARIT da SEARX é usada pela subrotina de leitura de palavra (PAC) quando esta encontrar um bit de estado em "1" (PARIDADE, GAP, LOCAL, etc.). A subrotina BACK é chamada para retroceder 1 registro e então é checado o número de tentativas: Sendo menor que 4, o registro citado é relido, porém, se for igual a 4, é **considerado** com erro irre recuperável: A saída do "SIO MT DRIVER" é preparada em P+3 (erro), a FM é parada e a subrotina SEARX retorna em Q+1 de sua chamada.

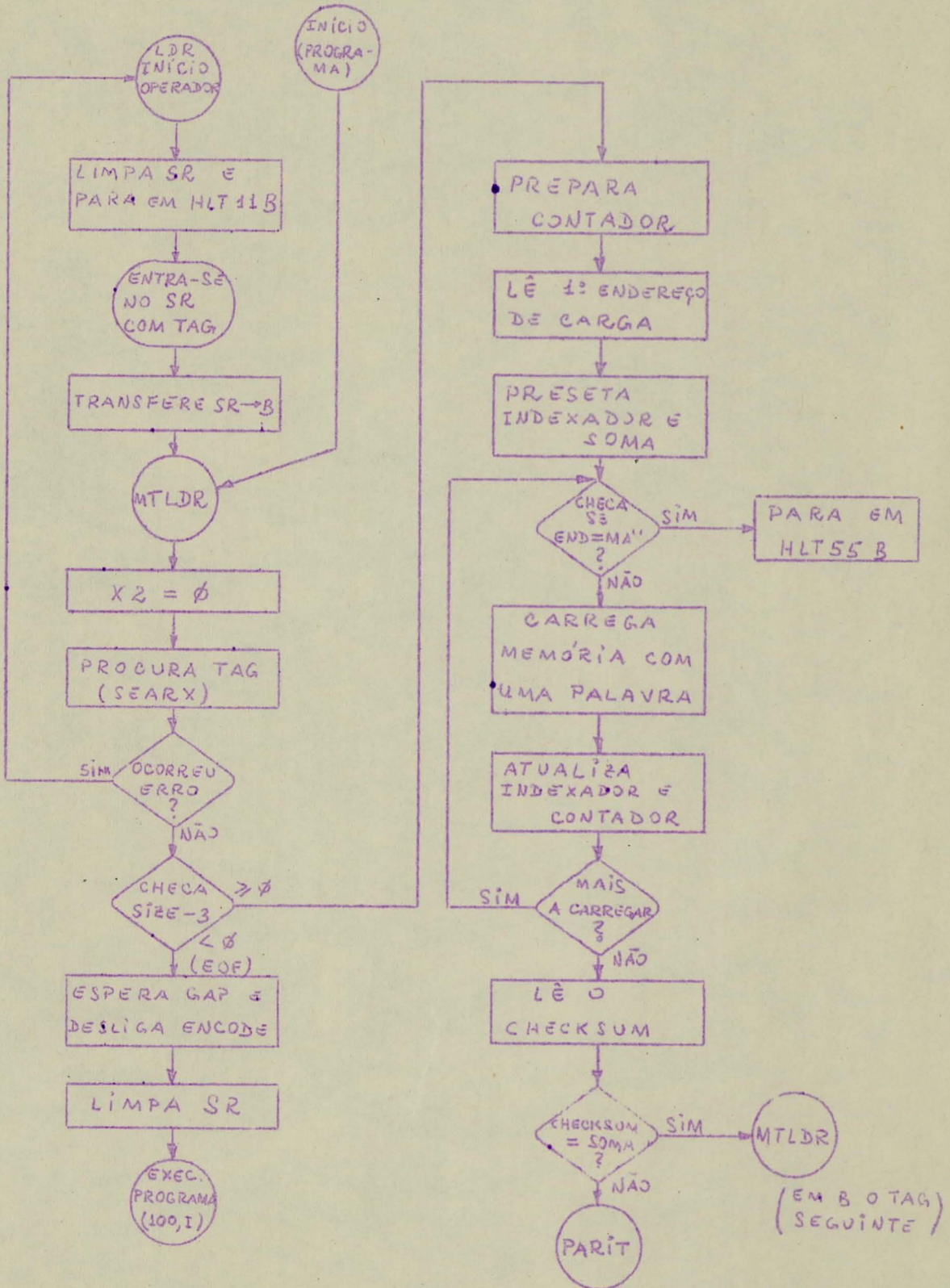
LOADER

O LOADER carrega na memória do Computador programas absolutos que residem na FM. Pode ser chamado de 2 maneiras:

- 1- Diretamente pelo operador, colocando-se o TAG completo no SWITCH REGISTER.
- 2- Por programa, entrando no endereço definido como MTLDR com o TAG no registrador B.

O LOADER carrega até encontrar um EOF que significa o fim do programa. Aí, o controle é transferido para um "LINK" definido em 100_8 onde deverá estar o endereço inicial do programa carregado. O diagrama de blocos da fig. 3.6 mostra como o LOADER trabalha.

MAG TAPE LOADER
DIAGRAMA DE BLOCOS



O TAG do registro pedido entra via SR ou registrador B, chamando-se a subrotina SEARX que se encarrega de buscá-lo. Caso ocorra erro o LOADER pára no início (HLT 11B) e o operador pode tentar outra vez. Tudo estando correto, é testado se $SIZE -3 < 0$ o que significa condição EOF: a Fita é parada no GAP, o SR é apagado e o controle é desviado para o endereço 100_8 indireto.

Se $SIZE -3 \geq 0$ a carga é preparada com a leitura da palavra seguinte que é o 1º endereço. As palavras restantes são lidas e carregadas na memória a menos que o indexador atinja o endereço do LOADER (HLT 55B). Terminado o bloco, é lido o CHECKSUM, o TAG é pôsto em B e incrementado, e a Soma é comparada com o CHECKSUM. Se for igual, desvia-se para MTLDR afim de ler o registro seguinte; se for diferente desvia-se para PARIT na SEARX afim de tentar a releitura do registro.

O LOADER usa muitas das subrotinas do DRIVER e também palavras como TAG. Este fato obriga essas subrotinas a fazerem parte do mesmo bloco absoluto assim como o TAG, pois esta é a palavra imediatamente anterior ao "ENTRY POINT" do DRIVER (por construção), e toda a inicialização. O restante do DRIVER forma um bloco que carrega abaixo dos SIO normais (TTY e Leitora), porém é compilado juntamente com o LOADER (ver lista gem, Apêndice C).

O LOADER leva um configurador que modifica as instruções I/O para "SELECT CODE" dado via "SWITCH REGISTER". Isto é realizado durante a montagem dos SIO DRIVERS, e a ordem de carga é a seguinte:

- | | | |
|-----|--------------------------|---------------|
| 1º) | MAG TAPE LOADER | |
| 2º) | TTY SIO DRIVER * | } SIO DRIVERS |
| 3º) | TAPE READER SIO DRIVER * | |
| 4º) | MAG TAPE SIO DRIVER | |

* Estes DRIVERS são os normais da HP, apenas com as origens mudadas para integrar o novo sistema.

Outra alternativa é prover uma fita sómente com o LOADER, e outra com os SIO, carregados na mesma ordem descrita na página anterior. A fita com os SIO pode ser usada para configurar compiladores, etc. A disposição com os enderços está na fig. 3.7.

II- PROGRAMA MONTADOR

Afim de se operar eficientemente a FM, foi desenvolvido um programa para montar o Sistema. Chamado simplificada-mente de MTAPE, reside no arquivo Nº1 (TAG 2001₈) até Nº 34₈ (incluindo SIO DRIVERS).

O MTAPE constrói uma tabela com nomes de programas e enderços na fita, ocupando também arquivo Nº1 e registros Nº 35₈ em diante. Quando o programa é carregado na memória, segue-lhe a tabela que pode então ser modificada. Por ocasião do fim da operação, a tabela é gravada novamente a partir do TAG 2035₈.

OPERAÇÃO

O MTAPE é um programa conversacional via Teletipo. Estão nele implementados 9 operações:

STORE
 STORER
 STOREL
 LIST
 DELETE
 END
 SPACE
 REWIND
 RETURN

STORE: CARGA de PROGRAMAS ABSOLUTOS

Os programas são carregados pela leitora de fita de papel e transferidos para a fita magnética em registros absolutos.

Por ocasião do comando STORE, o MTAPE pergunta: qual o nome? A resposta deve ser um alfanumérico de 5 caracteres. Em seguida a teletipo pede os números de arquivo e registro onde deve ser carregado o referido programa. A resposta pode ser 00, o que indica que o programa deve ser carregado no TAG seguinte ao atual caso contrário a fita é posicionada no registro anterior ao pedido.

O carregamento é efetuado até que seja encontrada uma condição de EOT na fita de papel, parando em HLT 57B. O bit 0 do SR decide se a carga terminou (BIT 0=1) ou se há mais fitas do mesmo programa a serem carregadas (BIT 0=0).

Terminada a carga, um registro EOF é gravado na fita, e na tabela é posto nome e endereço do programa carregado.

STORER

Esta se refere à carga de programas relocáveis, opera de maneira análoga, exceto quanto ao nome do programa que é lido do registro NAM da fita relocável, em vez de escrito na teletipo. O registro END determina a atualização da tabela, enquanto que a condição EOT gera um EOF na fita magnética. Os programas absolutos se diferenciam dos relocáveis, na tabela, por um asterisco no seu nome.

STOREL

Comando que permite a carga da Livraria de Subrotinas do Sistema. A Livraria poderia ser carregada como programas relocáveis (STORER), porém o excessivo número de programas alongaria demasiadamente a Tabela, e considerando que a Livraria é lida inteiramente pelo BCS original, resolveu-se que teria só um nome na tabela: LIBRY. Portanto, foi necessário um comando separado mas que funciona análogamente ao STORER.

LIST

O LIST comanda a listagem da Tabela de Programas. Na fig. 3.8 mostra-se um exemplo dessa tabela. São 4 colunas: Na primeira aparecem os nomes dos programas (os absolutos levam asterisco); na segunda o Nº do arquivo e nas seguintes os números do primeiro e último registros.

DELETE

A instrução "DELETE" determina a eliminação de um programa da Tabela. Quando comandada, a teletipo escreve "Qual o nome?" e se responde dando o nome do programa a ser apagado (se for absoluto deverá levar o asterisco). Se o nome dado não existir, uma mensagem é escrita na TTY e a operação não é executada. Esta instrução deve ser usada antes de recarregar um programa que já exista, pois ao se tentar tal operação sem eliminá-lo, o MTAPE rejeita com a mensagem "Este já existe".

END

Este comando é usado para gravar na fita um registro EOF de número Zero. Este tipo de registro serve para marcar o início de cada Arquivo para identificação. O número do arquivo deve ser dado pela teletipo à pergunta: "Número de Arquivo".

SPACE

O "SPACE" comanda 10 FILE GAPS consecutivos para a separação de arquivos.

REWIND

Comanda REWIND na FM.

RETURN

Este é o comando usado para terminar o trabalho. Provoca a procura do TAG 2034 e a gravação subsequente da tabela. No fim faz um REWIND e o computador para em HLT 77B.

MONTAGEM DE SISTEMA COM MTAPE

A montagem inicial de um sistema começa com a carga na memória do MAG TAPE LOADER já configurado e do MTAPE com SIO DRIVERS também configurados. A FM deve estar READY e com chave seletora em REMOTE. Inicia-se em 14000_8 , e a TTY escreve "LOAD MTAPE" parando o computador em HLT 00. É carregado na fita magnética o próprio programa MTAPE com os Drivers seguindo-se um espaço, para futura expansão da Tabela, encerrando o arquivo 1. Deve-se marcar com a fita refletora apropriada o arquivo 2, e depois, com a FM READY, comanda-se um END com o número 02_8 . Agora o arquivo 2 pode ser carregado com programas usando STORE, STORER ou STOREL conforme o caso. No fim do arquivo 2 dá-se um SPACE e procede-se a marcação do arquivo 3 e assim sucessivamente.

A operação de marcação pode ser dispensada, caso a fita já tenha sido usada com este sistema. Ao fim da carga pode-se fazer um LIST para guardar e depois um RETURN, que escreve a Tabela na magnética. Toda vez que for necessário alterar o Sistema, chama-se da fita o MTAPE através do LOADER (TAG 2001) e opera-se.

ESTRUTURA DO MTAPE

O MTAPE é um programa extenso (857 palavras, fora Tabela e Drivers). Carrega a partir de 14000₈ e usa os SIO DRIVERS para entrada e saída. A tabela gerada por ele carrega em 12000₈ e pode se estender até 13777₈ (1024 palavras). Como cada Programa ocupa 5 palavras e o tamanho, 1 palavra, a tabela pode conter 204 nomes com endereço. O formato da tabela está na fig. 3.9 onde as 3 primeiras palavras contêm o nome do programa em ASCII. Os 8 bits menos significantes da 3ª palavra podem ser espaço para relocáveis ou* para absolutos.

As 2 palavras seguintes contêm o 1º e o último TAG do programa, respectivamente. No endereço 12000 está armazenado o número de elementos da tabela em octal.

Os comandos do MTAPE são identificados em outra tabela que, por razões de compatibilidade de subrotinas, tem o mesmo formato da de Programas. O nome do comando ocupa 3 palavras (6 ASCII) e a seguinte contêm o endereço da rotina em questão. A 5ª palavra não é usada e portanto serve para guardar 1 constante. Se for dado um comando inexistente a operação é rejeitada e uma mensagem "ERRO DE COMANDO" é impressa.

Os "BUFFERS" de comentários têm o formato mostrado na fig. 3.10 onde a subrotina WRITE se encarrega da sua escrita. Antes de cada mensagem um CRIF é realizado.

CHAMADA DO WRITE

figura 3.10 A

```
P      JSE WRITE
P+1    DEF <BUFFER>
P+2    <Retorno Normal>
```

BUFFER : Em Assembler

```
<BUFFER> DEC 2M
          ASC M, <comentários>
```

NÚMERO	CARACTERS
C	O
M	E
N	T
A	R
I	O
S	espaço

figura 3.10 B

Na entrada de dados, se apenas um caracter for lido, a mensagem é rejeitada. Isto permite que uma operação como STO RE que requer 3 etapas (nome, arquivo e registro) seja abortada pelo operador antes de sua execução.

LISTAGEM DA TABELA :
FORMATO

NAME	FILE	BEGN	END
MTAPE *	1	1	34
TABLE *	1	35	36
PASS 1 *	2	1	171
PASS 2 *	3	1	56

FIG 3.9

TABELA — FORMATO MEMÓRIA :

ENDEREGOS EM OCTAL

12 000	Nº DE ELEMENTOS		
1	N	0	1: PROGRAMA
2	M	E	
3	1		
4	1: TAG		2: PROGRAMA
5	ÚLTIMO TAG		
6	N	0	
7	M	E	
10	2		
11	1:		
12	ÚLTIMO		
13	N	0	

{ * — ABSOLUTO
 ESPAÇO — RELOCÁVEL

III- SISTEMA OPERACIONAL

Sendo possível a carga de programas absolutos na FM e sua utilização através dos Drivers, passou-se a implantar um pequeno Sistema Operacional capaz de dar conta da compilação de Programas fonte em FORTRAN e sua carga na memória.

COMPILAÇÃO FORTRAN (8k)

No Sistema de fita de papel são usados 2 passos, o primeiro produz uma fita intermediária para ser usada pelo segundo na confecção da fita objeto(relocável).

A perfuração no nosso sistema é realizada pela Teletipo (10 CAR/s), o que torna o processo demasiadamente lento. A primeira providência para acelerá-lo seria a supressão da fita intermediária, passando a gravá-la na KENNEDY cuja velocidade é 30 vezes maior que a da Teletipo. A operação também poderia ser melhorada colocando os compiladores também na FM, providenciando sua carga e execução automática.

Para operar os compiladores com fita magnética bastaria em princípio definir o LINK 107₈, pois estes programas têm as chamadas para os DRIVERS de FM originais da HP, e os nossos são compatíveis exceto quanto ao posicionamento. O posicionamento original baseia-se em número de arquivos e registros relativos, enquanto que o nosso, por ter arquivos definidos fisicamente (MARCAS BOT) endereça cada registro e cada arquivo. Portanto, a chamada de posicionamento foi encontrada no compilador e alterada (fig. 3.11).

A carga e execução dos compiladores teve de ser implementada através de um programa absoluto "Controle dos Compiladores" que também providencia ligação com o LOADER do BCS (Carga de fitas relocáveis).

COMPILADOR FORTRAN - PASSO 1
 MODIFICADO
 FIGURA 3.11

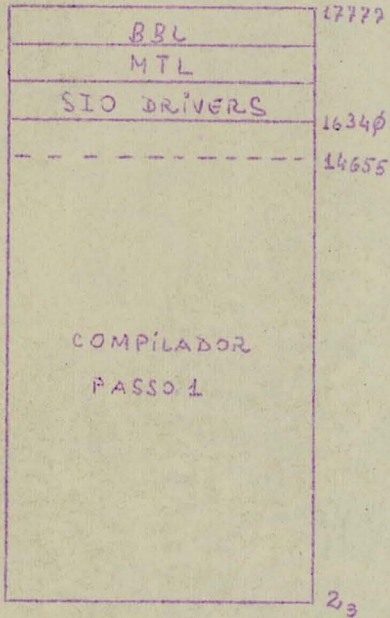
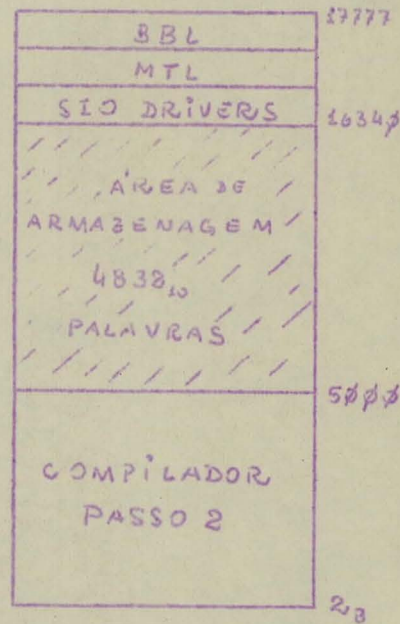
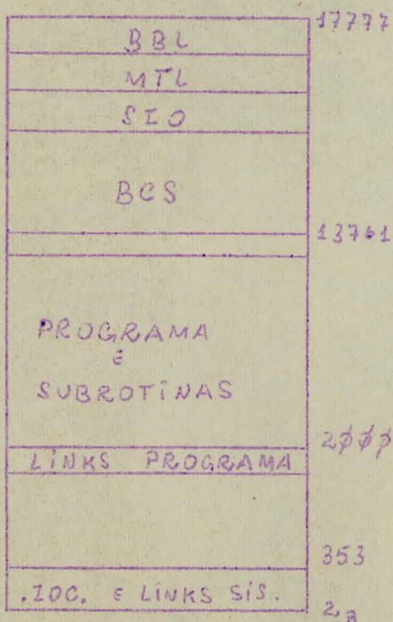
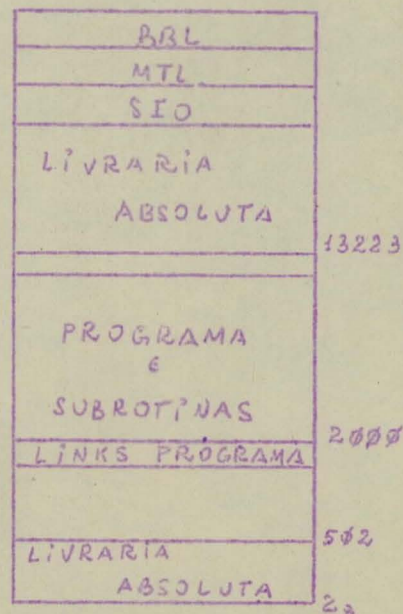
30	124032	JMP	32.1	
31	003723			
32	003725			
33	003715			
34	114107	JSB	107.1	FAZ REWIND ANTES DE
35	000003			LISTAR ASMB.
36	002400	CLA		
37	025772	JMP	1772	
40	000000	NOP		
41	000000	NOP		
42	000000	NOP		
43	002003	SZA,RSS		
44	106612	OTB	12	
45	024001	JMP	B	
46	177730	STB	1730.1	
47	102077	HLT	77	
50	024051	JMP	51	
51	002400	CLA		
52	070050	STA	50	
53	000000	NOP		ENTRY PASSO 1.
54	050040	CPA	40	
55	025101	JMP	1101	
56	070040	STA	40	
57	024064	JMP	64	VAI POSICIONAR.
60	000000	NOP		ENTRY PASSO 2.
61	024064	JMP	64	POSIC. PASSO 2.
62	010000			TAG AREA RASCUNHO.
63	000000	NOP		
64	060062	LDA	62	EM A TAG.
65	007400	CCB		PASSO 2 CARREGA CLB AQUI.
66	114107	JSB	107.1	
67	000004			
70	102040	HLT	40	
71	024006	JMP	6	
72	060110	LDA	110	
73	070210	STA	210	
74	124206	JMP	206.1	
75	014566	JSB	566	
76	052000	CPA	A	
77	124053	JMP	53.1	RETORNO P 1. PASSO 2 (JMP 60.1)
100	016341			LINK MT LOADER. INICIO C. COMP.
101	016722			LINK TAPE INPUT.
102	017264			LINK LISTING.
103	017264			LINK PUNCH.
104	017102			LINK KEYBOARD.
105	014655			PRIM. PAL. DISPONIVEL.
106	016340			ULTIMA PAL. DISPONIVEL.
107	017376			LINK MAG TAPE DRIVER.

LOADER e LOAD AND GO

Uma outra etapa no aceleração do sistema é a possibilidade de carregar o programa compilado e executá-lo sem fitas intermediárias de papel. Esta operação só é possível se houver algum lugar para armazenar o programa objeto, durante a execução do passo 2, pois neste ínterim, a FM está ocupada em ler o passo intermediário. Afortunadamente, o Compilador passo 2 deixa uma grande área de memória disponível (fig. 3.12) que foi usada para este fim. Após completada a compilação, o programa objeto é carregado na FM para ser lido pelo LOADER.

O BCS também passou a residir na fita assim como a Livraria de Subrotinas. O LOADER teve de sofrer alterações como início de página zero disponível, fim de memória, etc. As chamadas ao .IOC. (Controle de entrada e saída) não foram alteradas, porém o .IOC. original não foi usado: Construiu-se um novo .IOC. absoluto e particular que transfere convenientemente as chamadas de I/O para as subrotinas de saída (SIO DRIVERS). Os SIO foram usados para operar em todo o sistema, ao contrário do original, onde só trabalhavam durante a compilação. Desse modo, o sistema tornou-se mais compacto. O novo .IOC. também é usado em tempo de execução pela subrotina FRMTR que se encarrega das operações I/O em FORTRAN. As chamadas de posicionamento ficaram por conta do .IOC. e este ainda decide se a entrada será pela leitora de fita de papel ou pela magnética.

O BCS também pode ser chamado pelo operador para carregar programas objetos em fita de papel. Este modo de operar é determinado pelo TAG escolhido: O BCS tem dois TAGS iniciais, um para operação LOAD AND GO e outro para operação externa. Conforme o TAG escolhido, uma palavra chave do .IOC. é alterada por ocasião da carga, decidindo assim aonde vai ser lido o programa objeto.

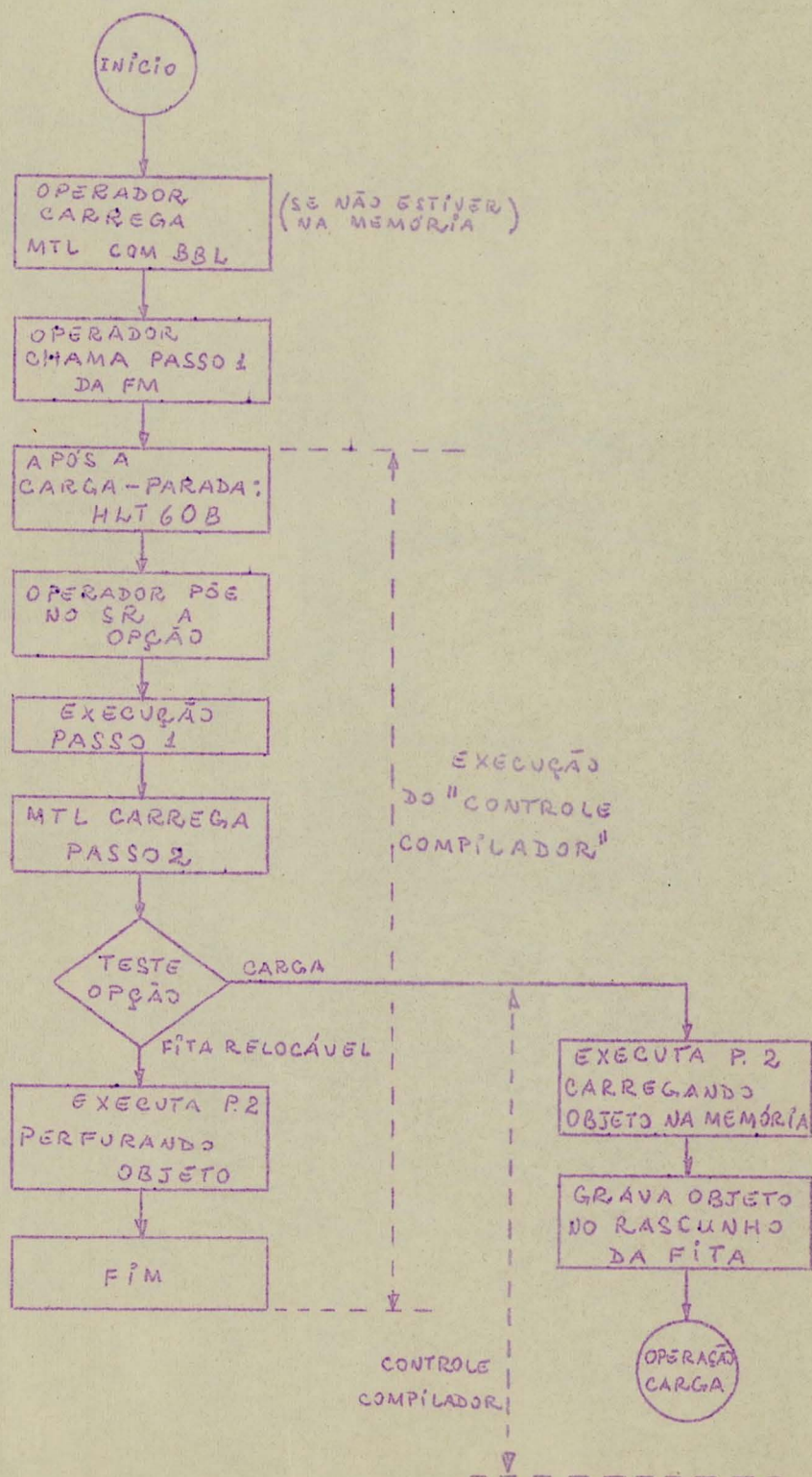
OPERAÇÃO LOAD AND GOMAPAS DE MEMÓRIA :DURANTE COMPILAÇÃO :PASSO 1(ENDEREGOS
EM
OCTAL)PASSO 2DURANTE A CARGAEXECUÇÃO

Nas figuras 3.13 e 3.14 mostra-se o diagrama de blocos do procedimento de Compilação incluindo carga. Após carregado o passo 1, o operador decide a opção:

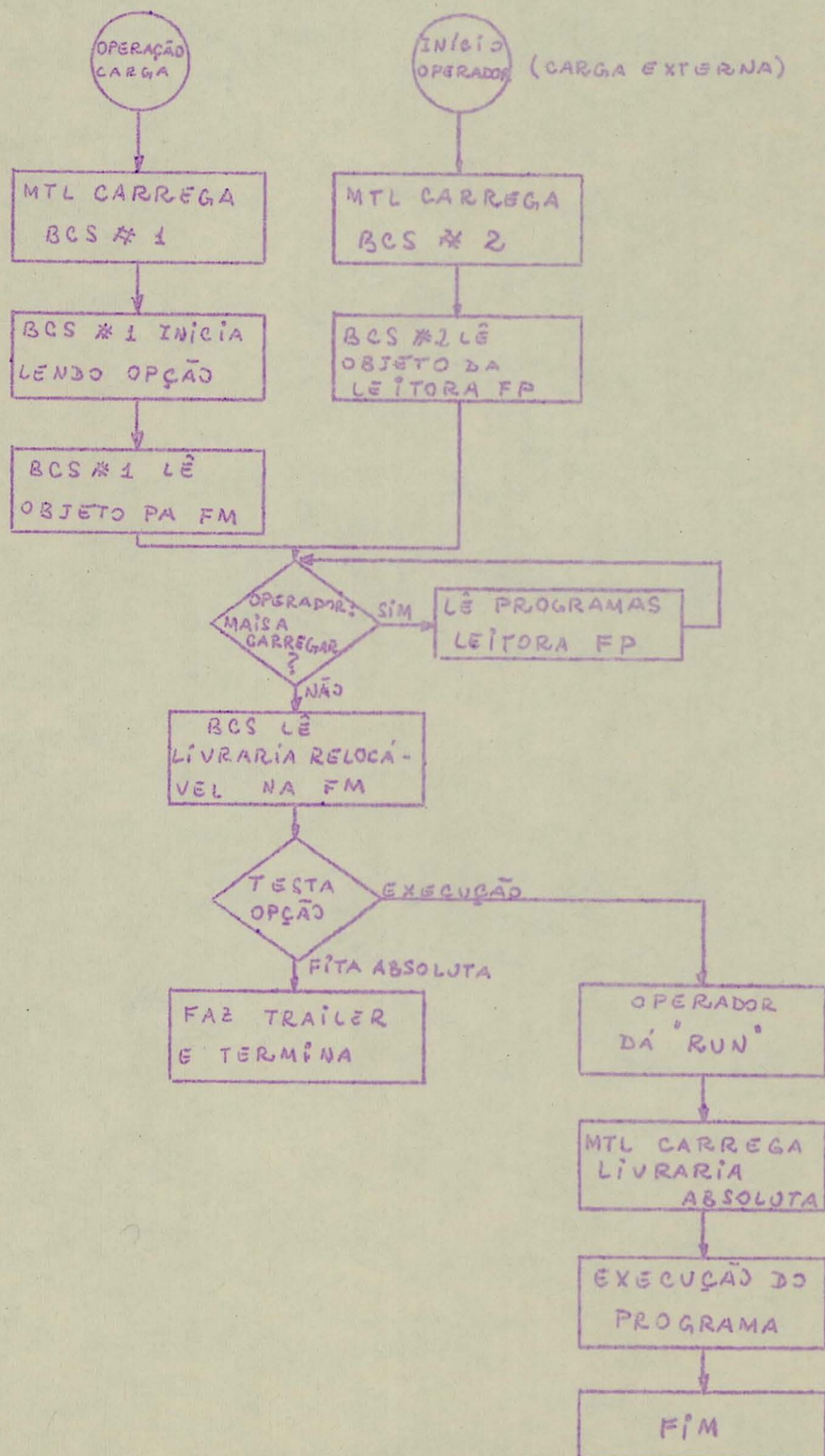
- 1- Confeção fita Relocável (SR=0)
- 2- Carga Simulada em fita Absoluta (SR=140000₈)
- 3- LOAD AND GO (SR=100000₈)

Na figura 3.12 estão os mapas de memória correspondente às diversas etapas de uma operação LOAD AND GO. Nota-se que a livreria foi subdividida em duas: A primeira em formato Absoluto contendo as principais subrotinas aritméticas e de sistema (incluindo FRMTR) necessárias a qualquer programa em FORTRAN. O restante das subrotinas (funções transcendentes, exponenciais, etc.) constituem a livreria propriamente dita que será lida pelo ECS. A livreria Absoluta só é carregada em tempo de execução, e sobrepõe o LOADER do ECS poupando assim boa parte da memória.

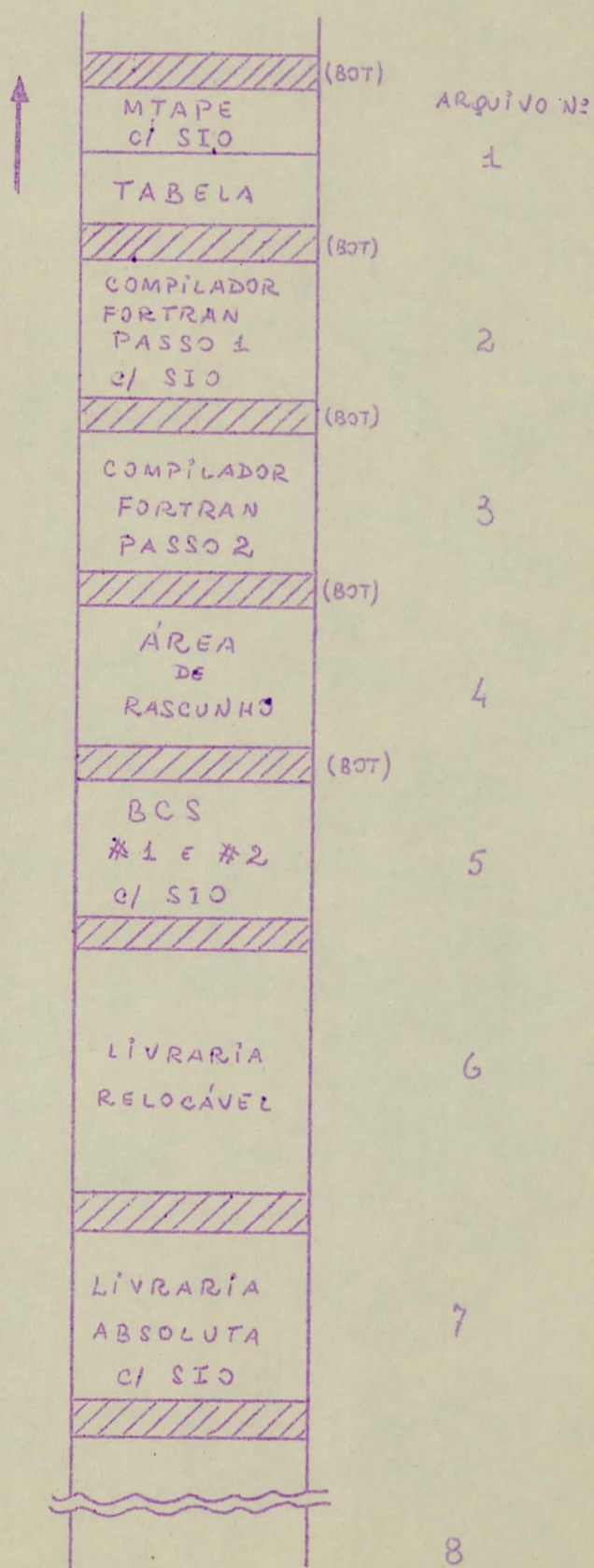
O arranjo atual dos arquivos está na figura 3.15 . A área de rascunho foi disposta logo após os compiladores para maior eficiência na busca. Tanto o passo 1 como o ECS e a livreria absoluta são configurados com os SIO DRIVERS, permitindo seu funcionamento independente.

SISTEMA - COMPILAÇÃO

SISTEMA - CARGA



DISPOSIÇÃO DO SOFTWARE NA FITA:



4- CONSIDERAÇÕES FINAIS

O trabalho mostrou ser possível a operação eficiente de um sistema de fita magnética, utilizando apenas uma unidade lenta. No caso foi usada a KENNEDY por estar disponível. No entanto pode-se implementar este sistema com máquinas mais baratas (ex. Gravador de Instrumentação tipo cassete), obtendo resultados comparáveis. Algumas modificações na lógica seriam necessárias, como a na KENNEDY, para acelerar o procedimento de busca.

O SOFTWARE do sistema atual pode ser ampliado facilmente para dar conta da compilação de programas em ALGOL e ASSEMBLER, editar programas fonte e compilá-los sem perfurar fita, e armazenar dados já processados ou adquiridos em experiências de Física Nuclear.

Em aquisição de dados, a Magnética poderia trabalhar com o Multicanal NUCLEAR DATA e, posteriormente, alimentar o computador com os espectros para processamento. Neste caso, os DRIVERS poderiam ser simplificados, permanecendo as operações POSITION, READ e REWIND.

O SOFTWARE poderia ser adaptado para interligar o sistema HP a outro de maior porte, onde houvesse máquina compatível; os arquivos e os registros obedeceriam ao formato do sistema maior. Seria uma comunicação "OFF - LINE", entre os computadores.

Em resumo: O sistema apresenta boas possibilidades, tanto em Aquisição como em Processamento de dados.

A P E N D I C E S

A- DIAGRAMAS ESQUEMÁTICOS DAS MODIFICAÇÕES NA KENNEDY 1400R

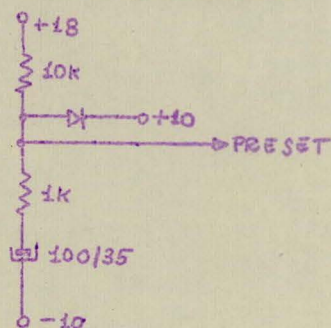
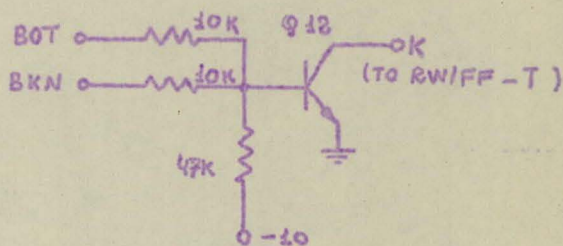
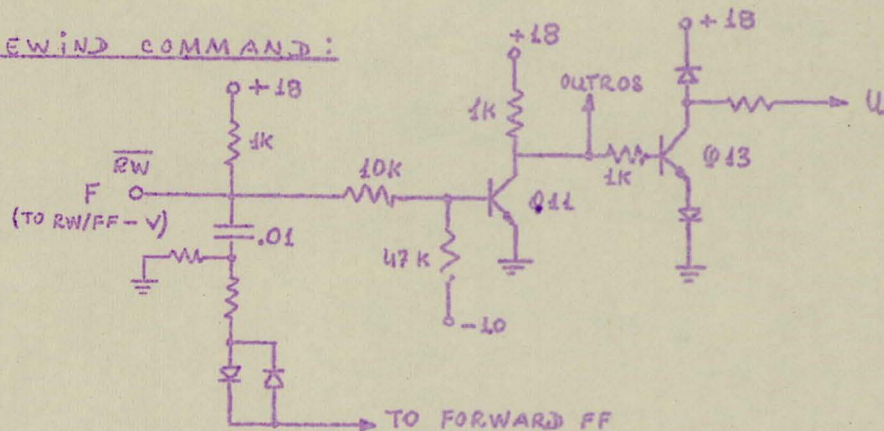
- 1- Modificações Internas
- 2- Lógica para Posicionamento
- 3- Nôvo STEP MOTOR DRIVE
- 4- Cabo de Interligação

B- DIAGRAMAS ESQUEMÁTICOS DO INTERFACE

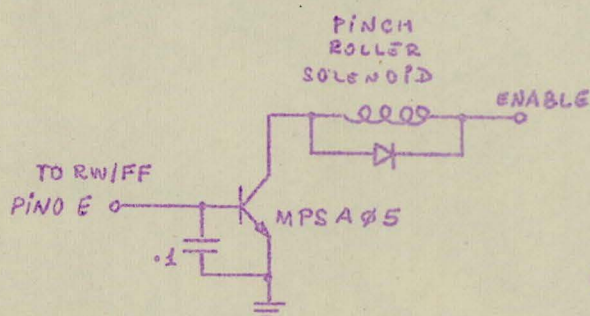
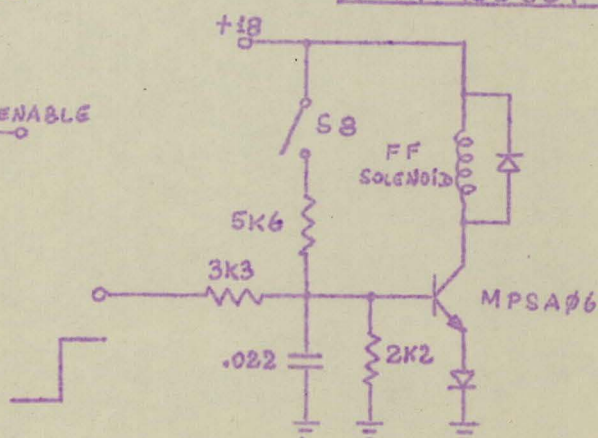
- 1- Registrador de Comando e FLAG
- 2- Lógica de Controle
- 3- Entrada de Dados
- 4- Saída de Dados e Paridade
- 5- Conversores de nível e seletor
- 6- Bits de Comando e Estado

C- LISTAGEM DO DRIVER e LOADER

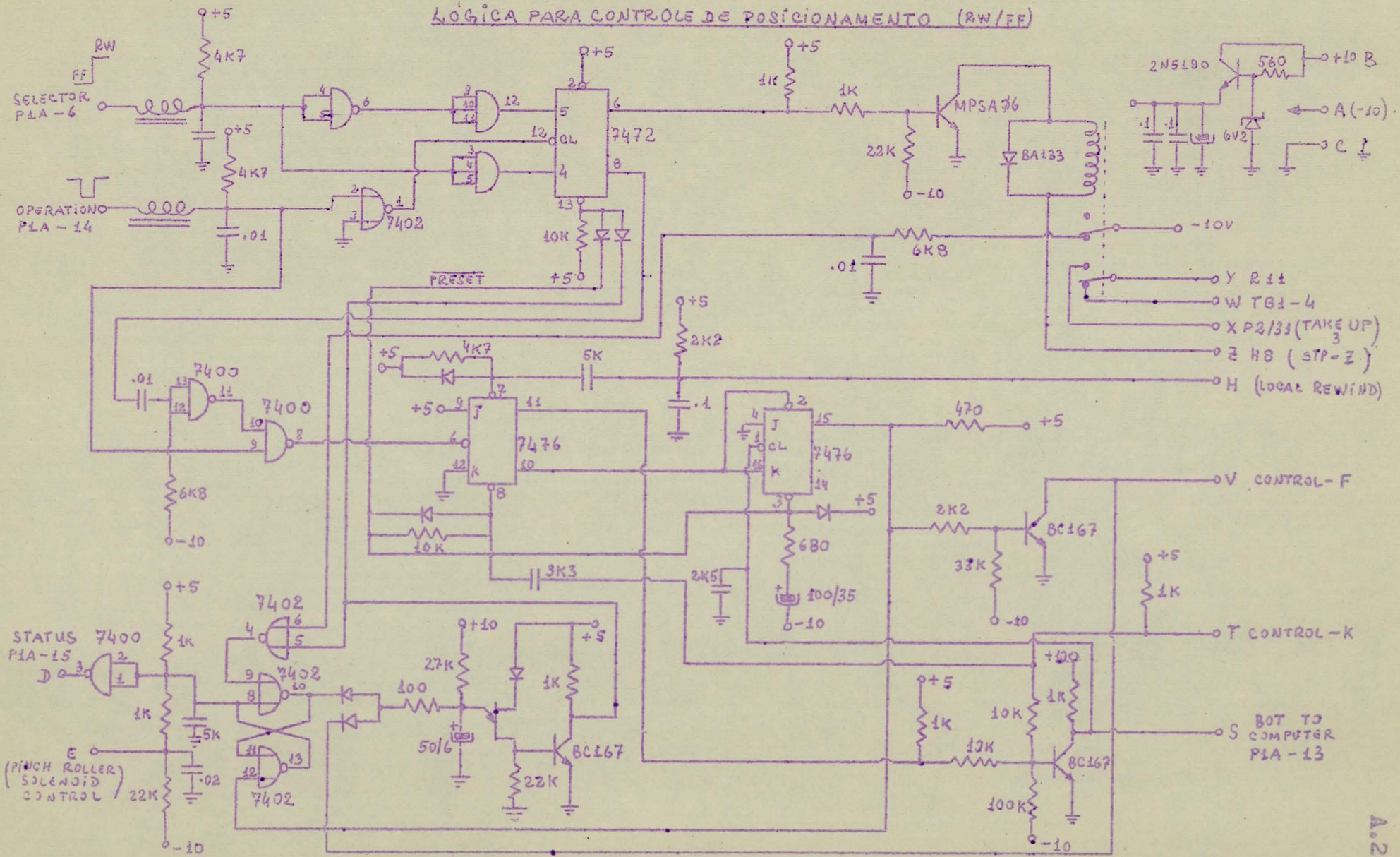
D- BIBLIOGRAFIA

KENNEDY - 1400RMODIFICAÇÕES INTERNAS:1. CONTROL CARD:BOTE SKN TAPE:REWIND COMMAND:

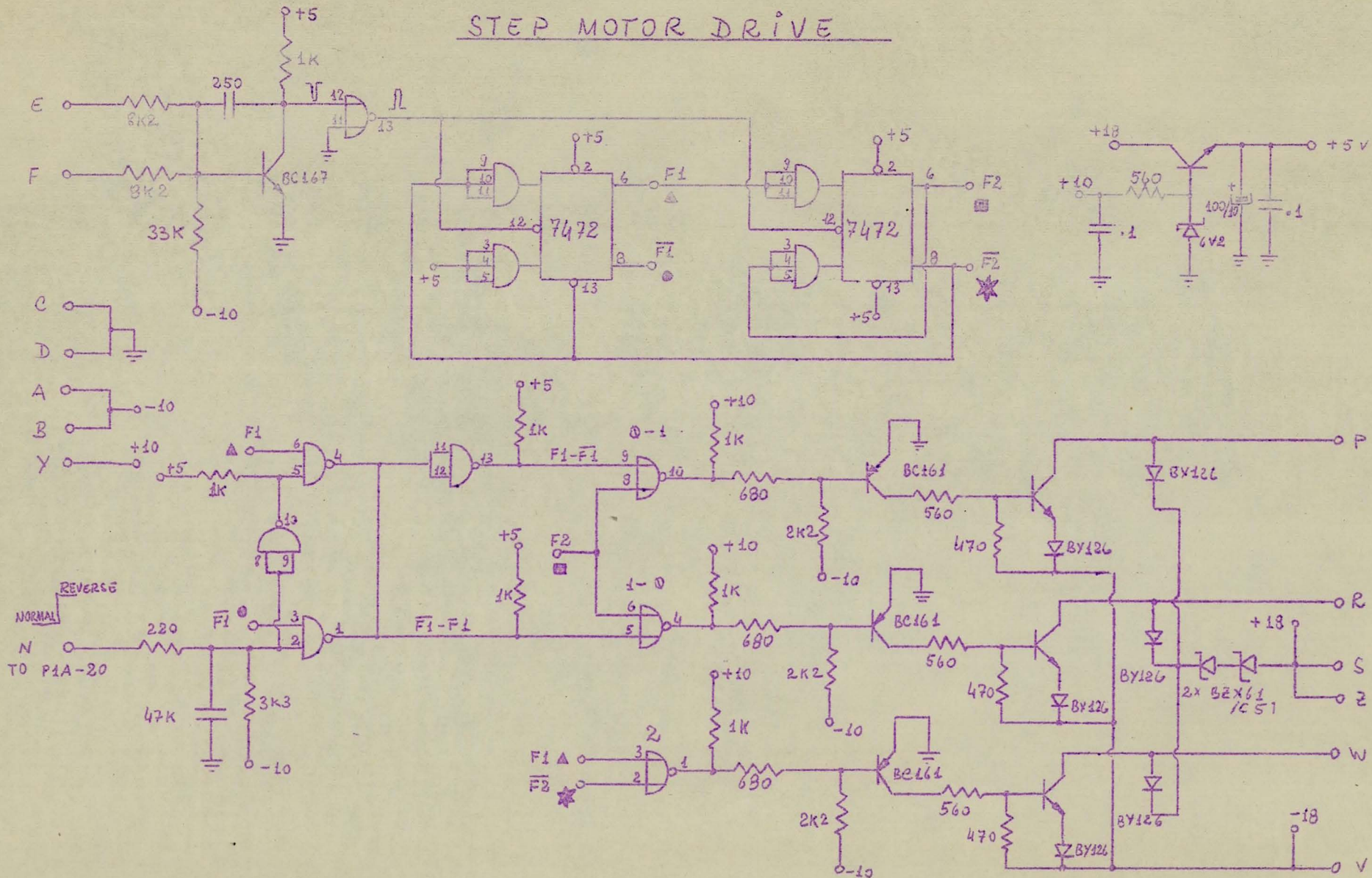
O COMANDO LOCAL DO BOTÃO FOI DESLOCADO DO PÍNO C-P PARA RW/FF - H

2. PINCH ROLLER INIBITOR:3. FF RESET:

LÓGICA PARA CONTROLE DE POSICIONAMENTO (RW/FF)



STEP MOTOR DRIVE



KENNEDY - INTERFACE

Cabo de Conexão

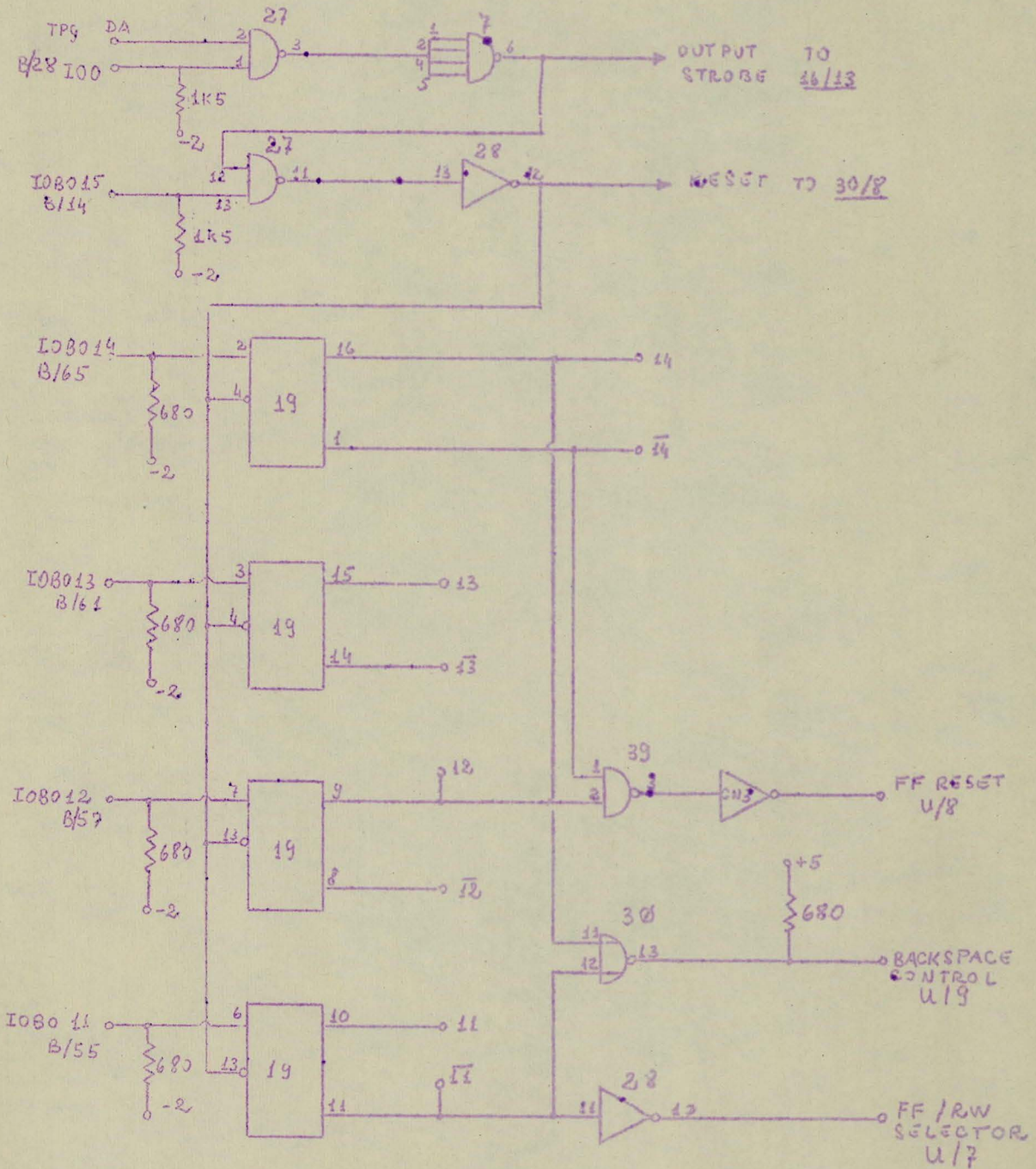
Significado	KENNEDY		INTERFACE UC
	Pl	PLA(nôvo)	
EOR Input	A	1	Y
EOF Input	E	2	AA
wr clock input	C	3	Z
signal ground	G	4	24
data in chn.1	H	5	V
FF/RW selector	pino J da lógicaRW/ FF	6	7
data in chn.2	J	7	U
" " " 4	K	8	T
" " " 8	L	9	S
" " " A	M	10	R
" " " B	N	11	P
EOT out	Y	12	M
EOT out	pino S da lógicaRW/ FF	13	E
rw/ff command	pino K da lógica rw/ ff	14	X
rw/ff status	pino D da lógica rw/ ff	15	6
gip out	e	16	1
chassis ground	b	17	n/c
foward in	F	18	W
fp reset	vai no co mando FP inhibit	19	8
reverser	pino N do STEP MOTOR DRIVE	20	9
vago		21	10

KENNEDY -- INTERFACE

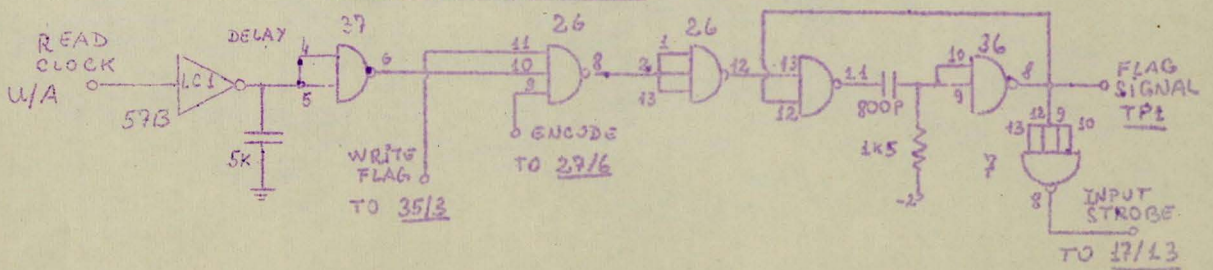
Cabo de Conexão

Significado	KENNEDY		INTERFACE UC
	P4	P4 A	
data in chn.1	1	1	K
" " " 2	2	2	J
" " " 4	3	3	H
" " " 8	4	4	F
" " " A	5	5	E
" " " B	6	6	D
" " " C	7	7	C
read clock out	10	8	A
gap det out	12	9	2
remote read select	14	10	5
remote write select	15	11	4
select common	16	12	3
FP out	17	13	N
remote status	20	14	L
LCC inhibit	21	15	nc
" "	22	16	nc
signal ground	35	17	BB
chassis ground	36	18	BB

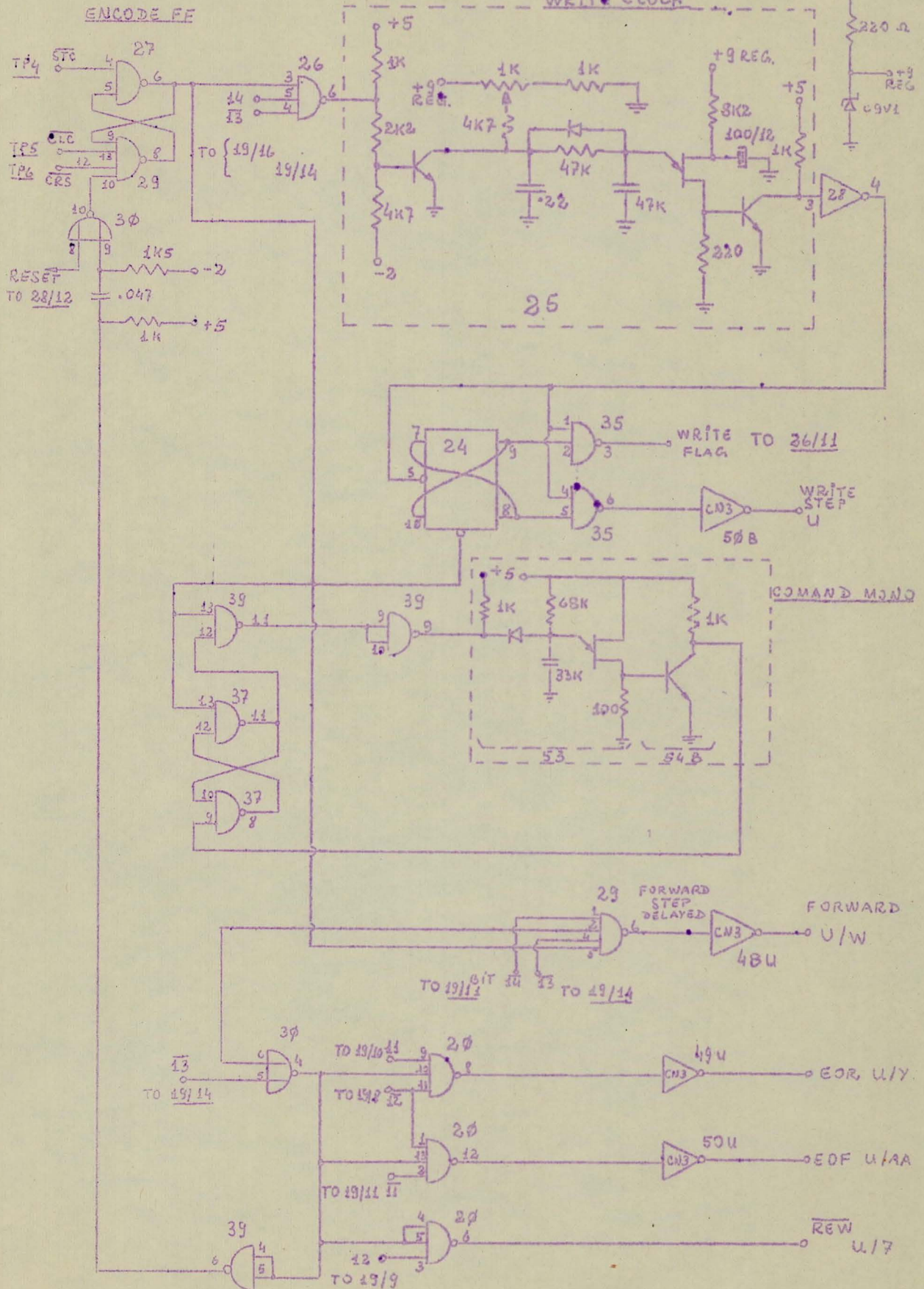
CONTROL REGISTER



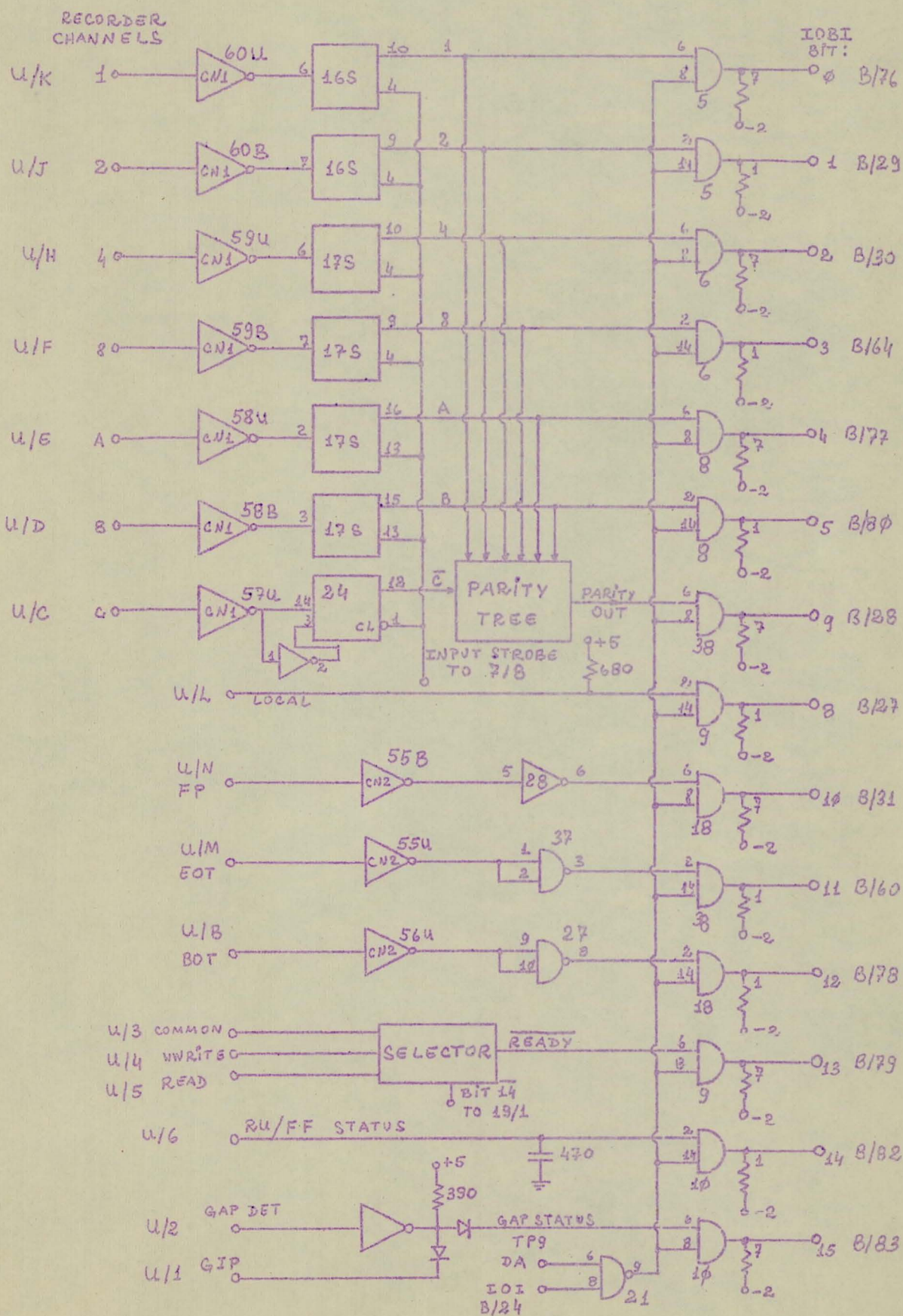
FLAG MONO:



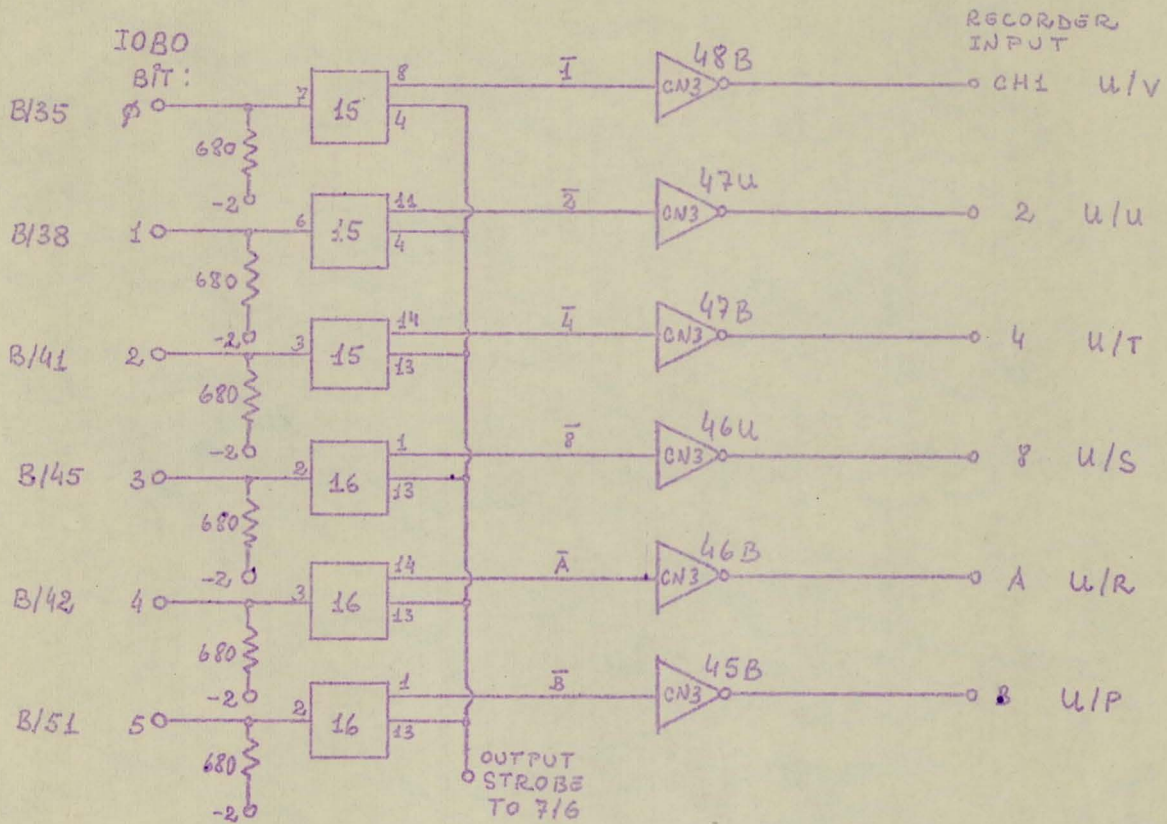
CONTROL LOGIC



DATA INPUT

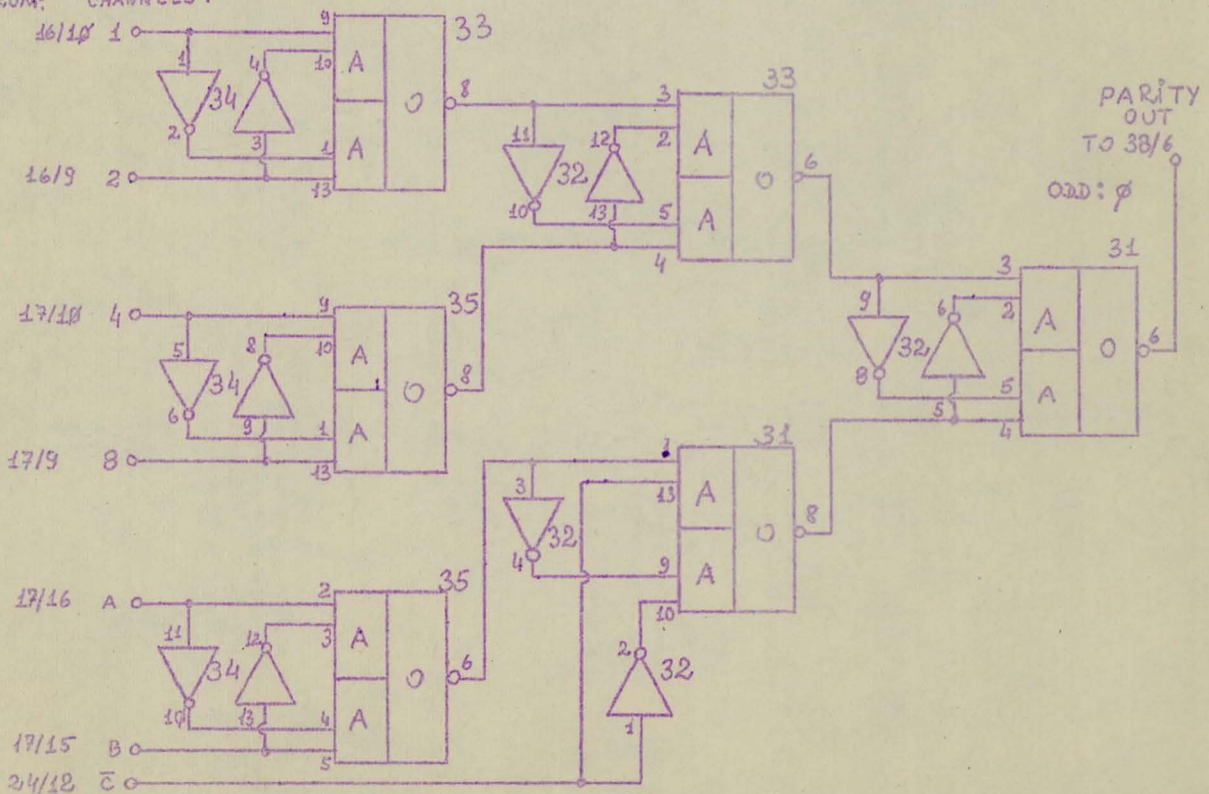


DATA OUTPUT

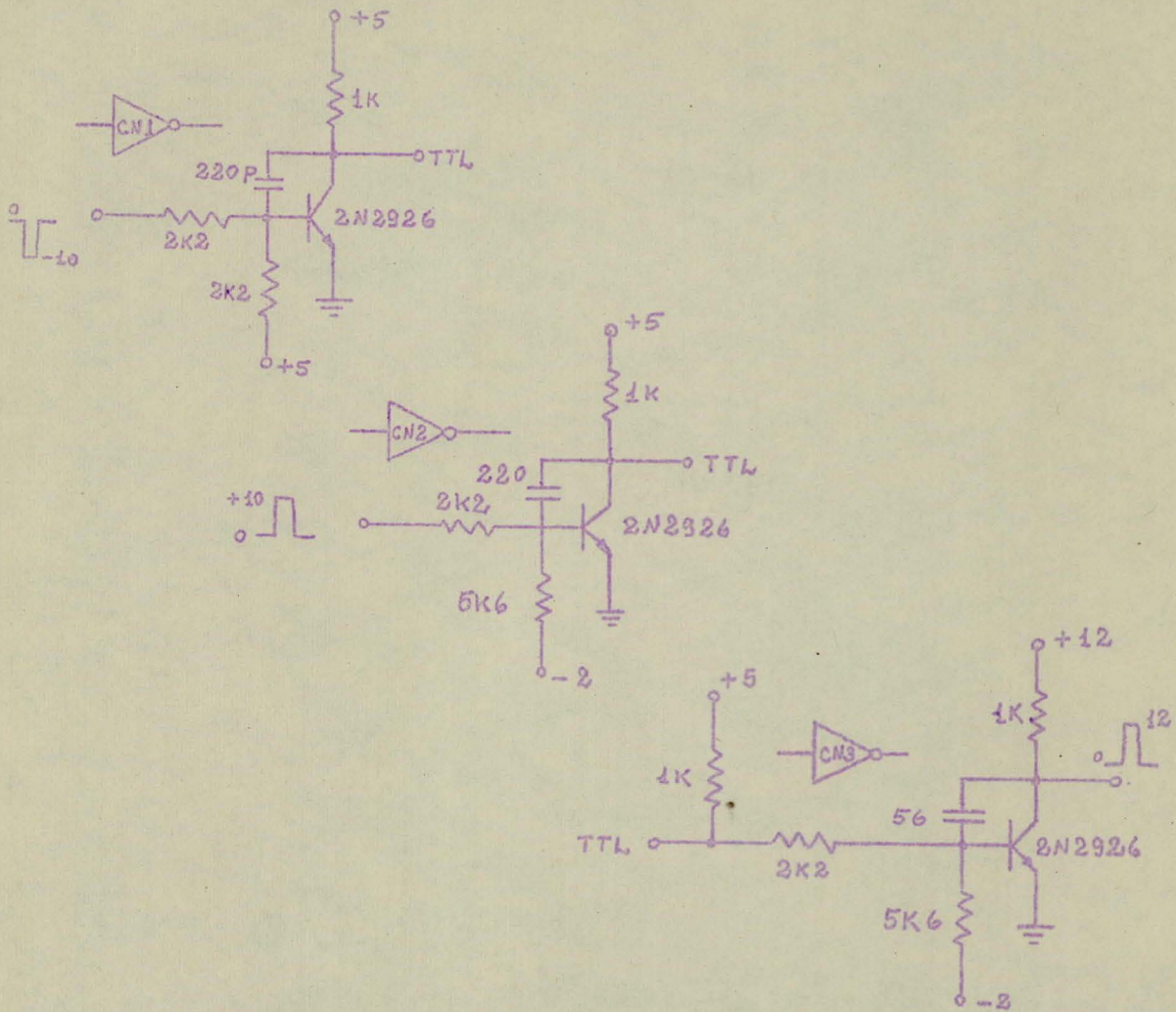


PARITY TREE

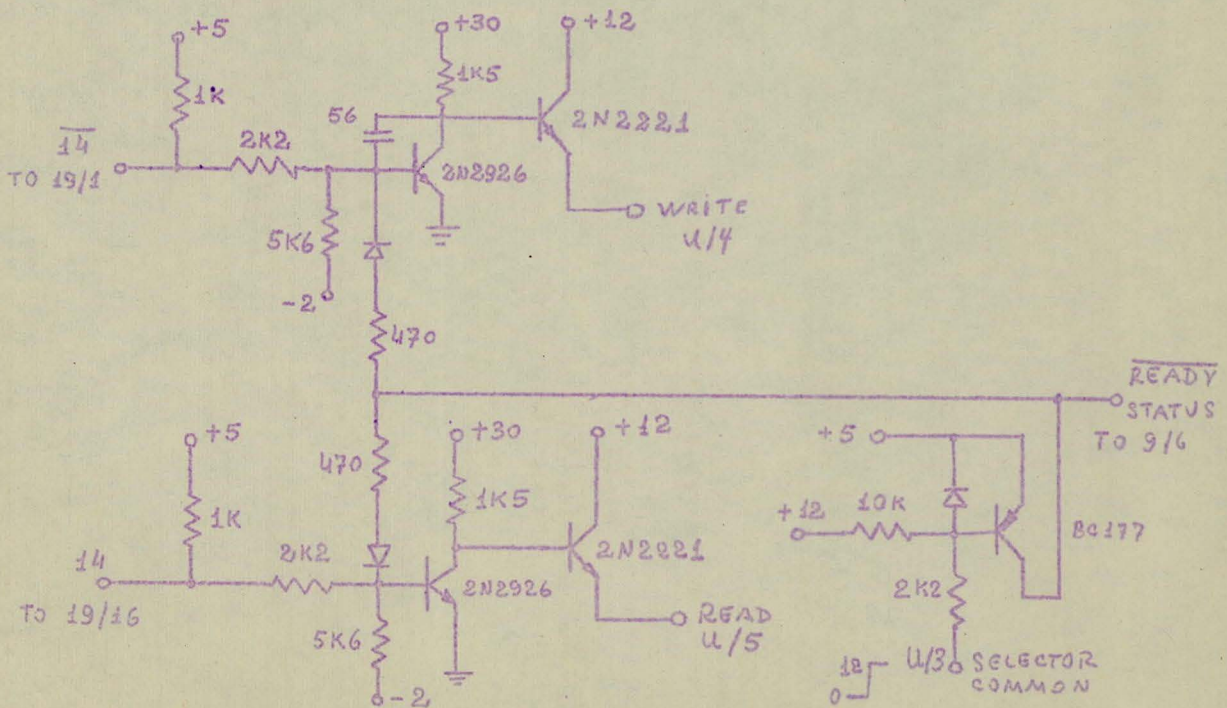
FROM: CHANNELS:



DISCRETE LEVEL CONVERTERS:



SELECTOR:



CÓDIGOS DE COMANDO

OPERAÇÕES	BITS				
	15	14	13	12	11
Read/ FPI	1	0	0	1	0
Read	1	0	0	0	0
Read/ REV	1	0	0	0	1
REWIND	1	1	1	1	1
Fast Fow.	1	1	1	1	0
Rec. Gap	1	1	1	0	1
File Gap	1	1	1	0	0
Write	1	1	0	x	x

BITS ESTADO

ESTADO	BITS	SIGNIFICADO
Parity	9	Erro de paridade
BOT	12	Marca sob sensor
FP	10	Escrever s/ anel
Ready	13	Máq. não está pronta
EOT	11	Marca sob sensor
Local	8	Máq. está local
RW/FF	14	Em RW ou FF
Gap	15	{Read: Lê um Gap {Write: Um Gap é escrito

OBSERVAÇÃO

Nos esquemas, tanto os circuitos integrados como os discretos, têm números que correspondem ao sequete da placa. Estes foram numerados da esquerda para a direita, e de baixo para cima, estando a placa voltada para o observador com a face onde se encaixam os componentes.

```

0001          ASMB,A,B,L
0003 16504          ORG 16504B
0004*
0005*          *ROTINAS DE EXECUCAO*
0006*
0007 16504 001310  REWND RAR,SLA          TESTA ESTADO READY
0008 16505 026510          JMP ++3          NAO ESTA: REJEITA.
0009 16506 067422          LDB CODE3          ESTA: EXECUTA
0010 16507 017442          JSB OUT
0011 16510 006400          CLB          RETORNA EM
0012 16511 127376          JMP MTR,1          P+2 COM B=0
0013*
0014 16512 066717  GAP  LDB CODE5          EM B O CODIGO DE FG.
0015 16513 002003          SZA,RSS          TESTA ESTADO P/ WRITE.
0016 16514 017442          JSB OUT          OK: EXECUTA.
0017 16515 026556          JMP END
0018*
0019 16516 007700  EOF  CCB,CCE          E=1: SIGNIFICA BIN.
0020 16517 077430          STB X3          TORNA O SIZE=-1
0021 16520 002001          RSS          SALTA P/ WRITE+1
0022 16521 016657  WRITE JSB AUX1          PRESETA OPERACAO
0023 16522 002002          SZA          TESTA ESTADO P/ WRITE
0024 16523 026556          JMP END          NAO TENDO CONDICoes VAI P/ END
0025 16524 037375          ISZ TAG          ESTANDO OK,INCREMENTA TAG
0026 16525 066715          LDB CODE2          PRESETA INTERFACE
0027 16526 017442          JSB OUT          EM WRITE E ESCREVE 1 BRANCO
0028 16527 062714          LDA PC55          ESCREVE MARCAS DE
0029 16530 016675          JSB UNPAC          INICIO DE REGISTRO
0030 16531 102101          STO          PRESETA ESCRITA DO TAG
0031 16532 002041          SEZ,RSS          TESTA E:
0032 16533 026540          JMP WR.1          E=0: REGISTRO BINARIO
0033 16534 063430          LDA X3          E=1: ASCII.PEGA TAMANHO
0034 16535 003004          CMA,INA          TROCA O SINAL
0035 16536 001727          ALF,ALF          GIRA 8 BITS
0036 16537 026543          JMP WR.2
0037 16540 162713  WR.1  LDA Z2,1          PEGA ELEMENTO DO BUFFER
0038 16541 036713          ISZ Z2          INCREMENTA INDEXADOR E
0039 16542 037425          ISZ Z3          CONTADOR
0040 16543 016675  WR.2  JSB UNPAC          ESCREVE
0041 16544 063375          LDA TAG
0042 16545 103201          SOC C          SE OVERFLOW=1
0043 16546 016675          JSB UNPAC          ESCREVE TAG.LIMPA OVERFLOW
0044 16547 037430          ISZ X3          ATUALIZA SIZE.TERMINOU?
0045 16550 026540          JMP WR.1          NAO
0046 16551 002400          CLA          SIM,ESCREVE
0047 16552 016675          JSB UNPAC          3 BRANCOS.
0048 16553 017451  RGAP JSB STATS          ESPERA GAP E PARA.
0049 16554 066716          LDB CODE4          FAZ UM
0050 16555 017442          JSB OUT          'RECORD GAP'
0051*
0052*          *ROTINA DE SAIDA*
0053*
0054 16556 017451  END  JSB STATS          ESPERA GAP E PARA.EM A O ESTADO
0055 16557 003031          CMA,SSA,SLA,RSS  CHECA BOT E EOT
0056 16560 026563          JMP M.OUT          SIM,SAI EM P+2
0057 16561 037376          ISZ MTR          NAO INCREMENTA
0058 16562 037376          ISZ MTR          2 VEZES O RETORNO

```

0059	16563	017451	M.OUT	JSB STATS	EM A O ESTADO
0060	16564	070001		STA 01	POE EM B
0061	16565	005700		BLF	E GIRA 4 BITS
0062	16566	063425		LDA Z3	EM A O NUMERO DE PALAVRAS
0063	16567	002040		SEZ	TRANSMITIDAS. SE E=1, ENTAO
0064	16570	001000		ALS	E ASCII, MULTIPLICA-SE POR 2
0065	16571	127376		JMP MTRD, I	SAIDA NORMAL DO DRIVER
0066*					
0067	16572	006007	PGST.	INB, SZB, RSS	CHECA SE B=-1 (POSICIONA P/
0068	16573	026576		JMP ++3	ESCREVER)
0069	16574	006400		CLB	B=-1, ENTAO TORNA-SE X2=0
0070	16575	077427		STB X2	
0071	16576	067430		LDB X3	EM B O TAG PEDIDO
0072	16577	026602		JMP ++3	
0073	16600	067375	READ	LDB TAG	POE TAG ATUAL EM B
0074	16601	006004		INB	E INCREMENTA P/ LER SEGUINTE
0075	16602	012720		AND P110K	MASCARA OS BIT ESTADO LEITURA
0076	16603	002002		SZA	TESTA
0077	16604	026556		JMP END	NAO HA CONDICOOES.
0078	16605	006002		SZB	TESTA SE NOVO TAG E ZERO
0079	16606	026614		JMP ++6	NAO
0080	16607	017902		JSB BACK	SIM: PROVIDENCIA O RETROCESSO
0081	16610	003400		CCA	DECREMENTA
0082	16611	043375		ADA TAG	TAG
0083	16612	073375		STA TAG	ATUAL
0084	16613	026556		JMP END	
0085	16614	017512		JSB SEARK	POSICIONA NO TAG DADO EM B
0086	16615	127376		JMP MTRD, I	RETORNO DE ERRO: SAI EM P+3
0087	16616	063427		LDA X2	CHECA X2
0088	16617	002003		SZA, RSS	
0089	16620	026556		JMP END	SE FOR ZERO ERA POSIC. P/ LER
0090	16621	002020		SSA	CHECA SINAL DE X2, SE FOR<0, ERA
0091	16622	026553		JMP RGAP	POSIC. P/ ESCREVER: FAZ RGAP
0092	16623	016657		JSB AUX1	X2>0, E LEITURA:
0093	16624	063502		LDA TEMPI	EM A O SIZE
0094	16625	002040		SEZ	CHECA E.
0095	16626	026632		JMP ++4	E=1: E ASCII
0096	16627	172713		STA Z2, I	E=0: BINARIO, POE SIZE NO BUFFER
0097	16630	036713		ISZ Z2	INCREMENTA INDEXADOR
0098	16631	037425		ISZ Z3	E CONTADOR
0099	16632	001727		ALF, ALF	ANALISA O SIZE:
0100	16633	012721		AND P377	MASCARA 8 BITS
0101	16634	043430		ADA X3	SOMA COM O DADO (NEGATIVO)
0102	16635	003004		CMA, INA	TROCA O SINAL E CHECA
0103	16636	002020		SSA	SE A<0: O LIDO>DADO,
0104	16637	002400		CLA	ENTAO USA O DADO
0105	16640	043430		ADA X3	SE A>=0: O LIDO<=DADO,
0106	16641	072657		STA Z1	USA-SE O LIDO.
0107	16642	036657		ISZ Z1	O SIZE LIDO E 1?
0108	16643	026650		JMP R.01	NAO
0109	16644	017451		JSB STATS	SIM, E EOF. ESPERA GAP E PARA.
0110	16645	006404		CLB, INB	APRONTA INDICADOR DE
0111	16646	005727		BLF, BLF	ESTADO DE EOF EM B.
0112	16647	127376		JMP MTRD, I	SAI EM P+2
0113	16650	017462	R.01	JSB PAC	LE UMA PALAVRA 16BITS
0114	16651	172713		STA Z2, I	GUARDA NO BUFFER
0115	16652	036713		ISZ Z2	ATUALIZA INDEXADOR,

JUL/20/72

0116	16653	037425		ISZ Z3	CONTADOR E
0117	16654	036657		ISZ Z1	CONTROLADOR
0118	16655	026650		JMP R.01	
0119	16656	026556		JMP END	TERMINANDO, EXECUTA ROTINA SAIDA.
0120*					
0121*					
0122*					
0123*					
0124*					
0125*					
0126	16657	000000	AUX1	NOP	
0127	16660	067427		LDB X2	PRESETA INDEXADOR
0128	16661	076713		STB Z2	
0129	16662	006400		CLB	PRESETA CONTADOR
0130	16663	077425		STB Z3	
0131	16664	067430		LDB X3	CHECA TAMANHO DADO
0132	16665	006003		SZB,RSS	SE FOR ZERO, REJEITA.
0133	16666	026556		JMP END	
0134	16667	006124		CLE,SSB,INB	LIMPA E, TESTA SINAL
0135	16670	126657		JMP AUX1,1	SE FOR <0, E BINARIO: SAI, E=0
0136	16671	005100		BRS	SE FOR >0, E ASCII: INCREMENTA
0137	16672	007300		CMB,CCE	O TAMANHO, DIVIDE POR 2, E
0138	16673	077430		STB X3	COMPLEMENTA, E=1
0139	16674	126657		JMP AUX1,1	
0140*					
0141*					
0142*					
0143*					
0144	16675	000000	UNPAC	NOP	
0145	16676	001200		RAL	PREPARA A PALAVRA
0146	16677	073462		STA TEMP3	E SALVA
0147	16700	063441		LDA M3	PREPARA CONTROLE LOOP
0148	16701	073426		STA X1	PARA 3.
0149	16702	063462	UN.1	LDA TEMP3	GIRA A PALAVRA
0150	16703	001722		ALF,RAL	5 VEZES E
0151	16704	073462		STA TEMP3	SALVA
0152	16705	013433		AND P77	PEGA OS 6 BITS MENOS
0153	16706	070001		STA 01	SIGNIFICATIVOS E DA SAIDA
0154	16707	017442		JSB OUT	ATRAVES DO REGISTRADOR B
0155	16710	037426		ISZ X1	CONTROLA O LOOP:
0156	16711	026702		JMP UN.1	RETORNA PARA CARACTER SEGUINTE.
0157	16712	126675		JMP UNPAC,1	RETORNO DA SUBROTINA
0158*					
0159*					
0160*					
0161	16713	000000	Z2	NOP	INDEXADOR
0162	16714	132643	PC55	OCT 132643	CONSTANTE QUE GERA MARCAS
0163	16715	140000	CODE2	OCT 140000	CODIGO PARA WRITE
0164	16716	164000	CODE4	OCT 164000	CODIGO PARA RGAP
0165	16717	160000	CODE5	OCT 160000	CODIGO PARA FGAP
0166	16720	110000	CODE7	OCT 110000	CODIGO PARA READ/FP RESET
0167	16721	000377	P377	OCT 377	MASCARA 8 BITS
0168	16504		A0	EGU REWND	MARCADOR INICIO DO BLOCO
0169	16721		A1	EGU P377	MARCADOR FIM DO BLOCO
0170	00107			ORG 107B	
0171	00107	017376		DEF MTRD	LINK DO DRIVER
0172*					A INSTRUCAO SEGUINTE NAO TEM SIGNIFICADO

JUL/20/72

0173*

0174*

0175 00110 000000

NO PROGRAMA, APENAS PERMITE O FRACIONAMENTO
DA FITA OBJETO. SEGUE UM TRAILER
BSS I

JUL/20/72

```

0002*                *CONFIGURADOR DAS INSTRUCOES I/O*
0003*
0004 02000           ORG 2000B
0005 02000 126001   JMP **1,I      O CONFIGURADOR ESTA
0006 02001 017357   DEF CONF      DE ACORDO COM OS
0007 17357           ORG 17357B      SIO DRIVERS ORIGINAIS,
0008 17357 106501   CONF LIB 01      SENDO USADO NA MONTAGEM.
0009 17360 007004   CMB,INB     SOBRE ESTA AREA E CARREGADO
0010 17361 037443   C.01 ISZ C.2      O DRIVER DA TELETIPO,
0011 17362 037444   ISZ C.3      POSTERIORMENTE
0012 17363 037445   ISZ C.4
0013 17364 037447   ISZ C.5
0014 17365 037452   ISZ C.6
0015 17366 037456   ISZ C.7
0016 17367 037470   ISZ C.8
0017 17370 037601   ISZ C.9
0018 17371 006006   INB,SZB
0019 17372 027361   JMP C.01
0020 17373 102077   HLT 77B
0021 17374 027373   JMP *-1
0022*
0023*                *INICIALIZACAO DO SIO DRIVER*
0024*
0025*
0026 17375 000000   TAG NOP      CONTEM O ULTIMO TAG TRANSMITIDO
0027 17376 000000   MTD R NOP    ENTRY POINT DO MGTP SIO DRIVER
0028 17377 077427   STB X2       SALVA B
0029 17400 073430   STA X3       SALVA A
0030 17401 163376   LDA MTD R,I  PEGA P+1 DA CHAMADA
0031 17402 037376   ISZ MTD R    PREPARA RETORNO EM P+2
0032 17403 013432   AND P7       MASCARA BITS DE COMANDO
0033 17404 043410   ADA ADDR     SOMA INICIO TABELA DE
0034 17405 072713   STA Z2       TRANSFERENCIA E SALVA.
0035 17406 017451   JSB STATS    POE BITS ESTADO EM A
0036 17407 126713   JMP Z2,I     TRANSFERE PARA EXECUCAO.
0037*                A CONTEM BITS DE ESTADO
0038*                B NAO FOI ALTERADO
0039 17410 117411   ADDR DEF **1,I TABELA DE TRANSFERENCIA.
0040 17411 016600   DEF READ
0041 17412 016521   DEF WRITE
0042 17413 016516   DEF EOF
0043 17414 016504   DEF REWND
0044 17415 016572   DEF POST.
0045 17416 016504   DEF REWND
0046 17417 016512   DEF GAP
0047 17420 016563   DEF M.OUT
0048*
0049*                *AREA DE CONSTANTES E ARMAZENAGEM DO LOADER*
0050*
0051 17421 100000   CODE1 OCT 100000 CODIGOS: READ
0052 17422 174000   CODE3 OCT 174000 REWIND
0053 17423 170000   CODE6 OCT 170000 FAST FORWARD
0054 17424 104000   CODE8 OCT 104000 BACKSPACE
0055 17425 000000   FLAG NOP     ACUSA REGISTRO INEXISTENTE
0056 17426 000000   X1 NOP      CONTROLE LOOP
0057 17427 000000   X2 NOP      SALVA BUFFER
0058 17430 000000   X3 NOP      SALVA TAMANHO

```

JUL/20/72

0059	17431	000000	XX3	NOP	CONTROLA NO. RELEITURAS
0060	17432	000007	P7	OCT 7	MASCARAS : 3 BITS
0061	17433	000077	P77	OCT 77	6 BITS
0062	17434	000043	P43	OCT 43	IDENTIFICADORES DE
0063	17435	000055	P55	OCT 55	MARCAS DE REGISTRO
0064	17436	150003	MASK	OCT 150003	MASCARA PARA BITS ESTADO ESCRITA
0065	17437	177400	STMSK	OCT 177400	MASCARA PARA BITS ESTADO.
0066	17440	017375	MAX	DEF TAG	INICIO DO BLOCO P/ LOADER
0067	17441	177775	M3	OCT -3	CONSTANTE -3

0068*
0069* *SUBROTINAS USADAS PELO DRIVER E PELO LOADER*

0070*
0071* *SUBROTINA DE TRANSFERENCIA (OUT)*
0072* ENVIA UM DADO OU COMANDO AO INTERFACE
0073* ENQUANTO LE UM DADO. USA:
0074* REG. B PARA SAIR
0075* REG. A PARA ENTRAR
0076*

0077	17442	000000	OUT	NOP	
0078	17443	106600	C.2	OTB 00	COMANDA I/O
0079	17444	103700	C.3	STC 00.C	NO MODO SEM
0080	17445	102300	C.4	SFS 00	INTERRUPCAO
0081	17446	027445		JMP *-1	DE MANEIRA
0082	17447	102500	C.5	LIA 00	CONVENCIONAL
0083	17450	127442		JMP OUT,I	

0084*
0085* *SUBROTINA DE ESTADO*
0086* ESPERA CONDIÇÃO DE GAP E REWIND PARA
0087* DESLIGAR O ENCODE FF. LE BITS DE ESTADO
0088* E OS PREPARA EM A.
0089*

0090	17451	000000	STATS	NOP	
0091	17452	102500	C.6	LIA 00	LE BITS ESTADO
0092	17453	001210		RAL,SLA	TESTA GAP (BIT 15)
0093	17454	001210		RAL,SLA	TESTA REWIND (BIT 14)
0094	17455	027452		JMP C.6	
0095	17456	106700	C.7	CLC 00	SE 15=1 E 14=0, DESLIGA A FM.
0096	17457	001222		RAL,RAL	PREPARA ESTADO EM BITS:
0097	17460	013436		AND MASK	0-3 E 12-15. ELIMINA BITS:
0098	17461	127451		JMP STATS,I	2,3 E 13. (GAP, REW E PARITY)

0099*
0100* *SUBROTINA DE LEITURA DE PALAVRA (PAC)*
0101*

0102	17462	000000	PAC	NOP	
0103	17463	063441		LDA M3	PREPARA 3
0104	17464	073426		STA X1	LOOPS
0105	17465	002400		CLA	LIMPA TEMP2
0106	17466	001722	P.1	ALF,RAL	GIRA 5 BITS
0107	17467	073451		STA TEMP2	SALVA EM TEMP2
0108	17470	102500	C.8	LIA 00	LE REG. DO INTERFACE
0109	17471	013437		AND STMSK	MASCARA BITS ESTADO.
0110	17472	002002		SZA	CHECA
0111	17473	027624		JMP PARIT	OCORRENDO QUALQUER UM, VAI RELER
0112	17474	006400		CLB	APAGA B PARA NAO ALTERAR COMANDO
0113	17475	017442		JSB OUT	LE UM CHARACTER
0114	17476	033451		IOR TEMP2	MONTA EM A
0115	17477	037426		ISZ X1	CONTROLA LOOP

JUL/20/72

0116	17500	027466		JMP P.1	
0117	17501	127462		JMP PAG.1	SAIDA NORMAL
0118*					
0119*					
0120*					
0121*					
0122	17502	000000	BACK	NOP	
0123	17503	017451		JSB STATS	ESPERA GAP E PARA.
0124	17504	067424		LDB CODE8	COMANDA 0
0125	17505	017442		JSB OUT	RETROCESSO
0126	17506	002400		CLA	ESPERA PARA RETROCEDER
0127	17507	034000		ISZ 00	0 MAIOR REGISTRO
0128	17510	027507		JMP *-1	(TAMANHO 100)
0129	17511	127502		JMP BACK.1	RETORNA


```

0131*
0132*
0133*
0134*
0135*
0136*
0137*
0138*
0139*
0140*
0141*
0142*
0143*
0144*
0145 17512 000000 SEARX NOP
0146 17513 003400 CCA PRESETA
0147 17514 073425 STA FLAG FLAG P/ 1 TENTATIVA
0148 17515 001020 ALS,ALS PRESETA
0149 17516 073431 STA XX3 XX3 P/ 4 LEITURAS.
0150 17517 017451 JSB STATS LE ESTADO
0151 17520 001325 RAR,ERA POE BIT 1 EM E (READY)
0152 17521 063375 LDA TAG SALVA EM A O TAG ATUAL
0153 17522 077375 STB TAG GUARDA NOVO EM TAG
0154 17523 002041 SEZ,RSS TESTA SE ESTAVA READY
0155 17524 027553 JMP CHECK SIM, CALCULA COM O TAG ATUAL
0156* NAO, LE-SE UM REGISTRO AFIM DE
0157* ATUALIZAR O TAG.
0158 17525 01451 REC JSB STATS INICIO DA LEITURA DE REGISTROS:
0159 17526 067421 LDB CODE1 ESPERA GAP E PARA. TESTA SE
0160 17527 003400 CCA X2=-1. CASO AFIRMATIVO O CODIGO
0161 17530 053427 CPA X2 E 7 (LEITURA C/ FP RESET), CASO
0162 17531 066720 LDB CODE7 CONTRARIO E 1 (LEITURA).
0163 17532 002101 RESET CLE,RSS ESTA ROTINA TENTA ACHAR AS MARCAS
0164 17533 002300 WAIT CCE DE INICIO DE REGISTRO QUE SAO
0165 17534 017442 JSB OUT 3 CARACTERES CONSECUTIVOS:
0166 17535 006400 CLB 55, 55 E 43
0167 17536 053435 CPA P55 ACHANDO UM 55 SEGUIDO DE 43, A
0168 17537 027533 JMP WAIT ROTINA TERMINA, DE OUTRO MODO
0169 17540 053434 CPA P43 VOLTA EM RESET A PROCURAR
0170 17541 002041 SEZ,RSS DO INICIO
0171 17542 027532 JMP RESET
0172 17543 017462 JSB PAC LEITURA DA PRIM. PAL=SIZE
0173 17544 073502 STA TEMPI SALVA EM TEMPI
0174 17545 017462 JSB PAC LEITURA DO ENDERECO
0175 17546 102601 OTA 01 FAZ DISPLAY NO SR.
0176 17547 067512 LDB SEARX PREPARA EM B O RETORNO
0177 17550 006004 INB EM Q+2
0178 17551 053375 CPA TAG VERIFICA TAGS: LIDO=PEDIDO?
0179 17552 124001 JMP 01,1 SIM, RETORNA EM Q+2.
0180* SE NAO E, PASSA A PROCURA-LO:
0181 17553 073462 CHECK STA AC2 PRIMEIRAMENTE O NO. DE ARQUIVO
0182 17554 001722 ALF,RAL E TESTADO (6 BITS MAIS SIGNF.).
0183 17555 001200 RAL O TAG ATUAL ESTA EM AC2 PARA
0184 17556 013433 AND P77 POSTERIOR VERIFICACAO
0185 17557 070001 STA 01 EM B E COLOCADO O ARG. ATUAL.
0186 17560 063375 LDA TAG AGORA O ARG. PEDIDO E
0187 17561 001722 ALF,RAL POSTO EM POSICAO

```

0188	17562	001200		RAL	
0189	17563	013433		AND P77	EM A, O ARQ. PEDIDO
0190	17564	003004		CMA,INA	EM A, O ARQ. PEDIDO NEGATIVO.
0191	17565	044000		ADB 00	EM B, ATUAL -PEDIDO
0192	17566	006003		SZB,RSS	CHECA SE SAO IGUAIS
0193	17567	027612		JMP BLOCK	SIM, VAI CHECAR OS REGISTROS
0194*					NAO, PROCEDE-SE O
0195*					POSICIONAMENTO DE ARQUIVOS.
0196	17570	017451		JSB STATS	ESPERA GAP E PARA.
0197	17571	060001		LDA 01	SALVA SOMA EM A
0198	17572	067423		LDB CODE6	EM B, CODIGO FAST FORWARD
0199	17573	002020		SSA	TESTA ATUAL-PEDIDO
0200	17574	027577		JMP **3	<0, PEDIDO ESTA NA FRENTE (FF)
0201	17575	003000	REV	CMA	>0, ESTA ATRAS:REWINDS(N+1).
0202	17576	067422		LDB CODE3	CODIGO REWIND
0203	17577	073462		STA XXI	EM XXI NUMERO DE REV OU FF
0204	17600	017442	REP	JSB OUT	COMANDA OPERACAO
0205	17601	102500	C.9	LIA 00	LE ESTADO
0206	17602	001700		ALF	PREPARA BOT BIT 0 E EOT 1.
0207	17603	002020		SSA	CHECA EOT
0208	17604	127512		JMP SEARX,1	SIM, SAI EM 0+1 (ERRO)
0209	17605	002011		SLA,RSS	CHECA BOT
0210	17606	027601		JMP C.9	NAO, ESPERA
0211	17607	037462		ISZ XXI	SIM, CONTROLA LOOP
0212	17610	027600		JMP REP	CONTINUA
0213	17611	027525		JMP REC	TERMINA, VAI LER UM REGISTRO
0214*					
0215*					CHEQUE DO NO. DE REGISTROS.
0216	17612	063462	BLOCK	LDA AC2	PEGA TAG ATUAL
0217	17613	003004		CMA,INA	TROCA SINAL
0218	17614	043375		ADA TAG	EM A, PEDIDO-ATUAL
0219	17615	002021		SSA,RSS	CHECA
0220	17616	027525		JMP REC	A=0=ESTA A FRENTE, LE SEGUINTE
0221	17617	017451		JSB STATS	A<0=ESTA ATRAS, PARA NO GAP
0222	17620	002400		CLA	PREPARA 1 REWIND
0223	17621	037425		ISZ FLAG	TESTA FLAG
0224	17622	102010		HLT 10B	ERRO: O REGISTRO NAO EXISTE!
0225	17623	027575		JMP REW	VAI FAZER REWIND
0226*					
0227*				*ROTINA PARIT*	
0228*				ERROS NA LEITURA	
0229*					
0230	17624	017502	PARIT	JSB BACK	COMANDA RETROCESSO
0231	17625	037431		ISZ XX3	CHECA NO. RELEITURAS
0232	17626	027525		JMP REC	<4, VAI LER DE NOVO
0233	17627	017451		JSB STATS	=4, ESPERA GAP E PARA.
0234	17630	037376		ISZ MTRD	PREPARA SAIDA DO DRIVER EM P+3
0235	17631	127512		JMP SEARX,1	SAI EM 0+1 (ERRO)
0236*					

```

0238*                               *MAG TAPE LOADER*
0239*
0240*                               CARREGA PROGRAMAS ABSOLUTOS
0241*
0242*                               CHAMADA:
0243*                               OPERADOR: END. 17632. TAG NO SR.
0244*                               PROGRAMA: END. 17636. TAG EM B.
0245*                               SAIDA: TRANSFERE VIA LINK EM 100.
0246*
0247 17632 002400 LDR CLA          INICIO OPERADOR:
0248 17633 102601 OTA 01          APAGA SR
0249 17634 102011 HLT 11B          E PARA.
0250 17635 106501 LIB 01          LE TAG DO SR
0251 17636 077427 MTLDR STB X2     INICIO VIA PROGRAMA: EM B O TAG.
0252 17637 103100 CLF 00          EM X2 PARA EVITAR FPRESSET
0253 17640 017512 JSB SEARX      PROCURA REGISTRO
0254 17641 027632 JMP LDR        EM CASO DE ERRO, RETORNA OPER.
0255 17642 063502 LDA TEMP1      PEGA TAMANHO,
0256 17643 001727 ALF,ALF        PREPARA CONTROLE DE LOOP.
0257 17644 043441 ADA M3
0258 17645 002021 SSA,RSS      CHECA SE TAM-3<0
0259 17646 027653 JMP **5       NAO, VAI CARREGAR
0260 17647 017451 JSB STATS    SIM, E EOF= PARA NO GAP
0261 17650 002400 CLA          APAGA
0262 17651 102601 OTA 01          SR
0263 17652 124100 JMP 100B,I   SAI VIA LINK EM 100
0264 17653 003004 CMA,INA     EM A: 3-TAM.
0265 17654 073376 STA X4       POE EM X4
0266 17655 017462 JSB PAC      LE END INICIAL, COLOCANDO-O
0267 17656 073427 STA X2       NO INDEXADOR E
0268 17657 073425 STA SUM      NA SOMA
0269 17660 017462 L.02 JSB PAC  CARREGA O RESTANTE
0270 17661 067427 LDB X2       DO REGISTRO. SE O
0271 17662 057440 CPB MAX      ENDERECO ATINGIR
0272 17663 102055 HLT 55B     MAX, PARA EM HLT 55.
0273 17664 173427 STA X2,I    FAZ A SOMA DURANTE
0274 17665 043425 ADA SUM      A CARGA
0275 17666 073425 STA SUM
0276 17667 037427 ISZ X2      ATUALIZA INDEXADOR
0277 17670 037376 ISZ X4      LOOP
0278 17671 027660 JMP L.02    CONTINUA
0279 17672 017462 JSB PAC     LE CHECKSUM
0280 17673 067375 LDB TAG     EM B O TAG ATUAL
0281 17674 006004 INB        EM B O TAG SEGUINTE
0282 17675 053425 CPA SUM     SE CHECKSUM=SUM, ENTAO VAI LER O
0283 17676 027636 JMP MTLDR   SEGUINTE
0284 17677 027624 A4 JMP PARIT SE FOR DIFERENTE EXECUTA PARIT
0285*
0286*                               *EQUIVALENCIAS USADAS NO PROGRAMA*
0287 17375          A3 EQU TAG      USADO PARA MARCAR INICIO
0288 16657          Z1 EQU AUX1
0289 17425          Z3 EQU FLAG
0290 17502          TEMP1 EQU BACK
0291 17451          TEMP2 EQU STATS
0292 17462          TEMP3 EQU PAC
0293 17425          SUM EQU FLAG
0294 17376          X4 EQU MTRD

```

0295	17462	XX1	EQU PAC
0296	17462	AC2	EQU PAC
0297	16720	P110K	EQU CODE7
0298			END

** NO ERRORS*

BIBLIOGRAFIA

- HEWLETT PACKARD : Pocket Guide to HP Computers.
- HEWLETT PACKARD : Pocket Guide to Interfacing HP Computers.
- HEWLETT PACKARD : Driver Manual.
- KENNEDY : Operation and Maintenance Manual - Model 1400R.
- M. ALGRANATI, T. NUNAMAKER and M. SACHS : Use of an IBM - Compatible Incremental Magnetic Tape Recorder as an Input/Output Device for a Multi-channel Analyzer.
- C. W. GEAR : Computer Organization and Programming.