TESE
337

ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

DISSERTAÇÃO DE MESTRADO:

Projeto e Implementação de um Programador de Memórias usando Microprocessador

LUIZ EDUARDO BORGES DA SILVA

ITAJUBÁ - MINAS GERAIS
1982

ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

DISSERTAÇÃO DE MESTRADO:

"PROJETO E IMPLEMENTAÇÃO DE UM PROGRAMADOR DE MEMÓRIAS USANDO MICROPROCESSADOR"

Escola Federal de Engenharia
Biblioteca MAUÁ
Itajubá

LUIZ EDUARDO BORGES DA SILVA

ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ
30 JUL 1982
BIBLIOTECA MAUÁ

ITAJUBÁ - MINAS GERAIS

1982

CLASS. 681.327.2.001.63 : 681.322(043.2)
CUTT. S586p
TOMBO 337

"PROJETO E IMPLEMENTAÇÃO DE UM PROGRAMADOR DE MEMÓRIAS

USANDO MICROPROCESSADOR"

AGRADECIMENTOS

Ao professor Ignacio Sergio Miranda Ferreira, orientador desta dissertação, pela consciência profissional e boa vontade com que dirigiu a realização deste trabalho.

Somos sinceramente gratos aos professores Felicio Barbosa Monteiro e Ismael Noronha, pelo incentivo constante e decisivo apoio na realização deste trabalho.

A todos os funcionários da EFEI que de uma maneira ou de outra contribuíram na execução deste trabalho.

A meus pais, Filigênio e Maria Geraldina que me possibilitaram chegar a este ponto.

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

1 - Microprocessors and logic Design

Ronald L. Krutz

John Willey and Sons - 1980

2 - Microprocessors and digital systems

Douglas V. Hall

McGraw Hill book company - 1980

3 - Digital concepts using standard integrated circuits

Richard S. Sandige

McGraw Hill book company - 1978

4 - Microprocessor: Design and Aplications

Andrew Veronis

Reston Publishing Comapany Inc. - 1978

5 - Logic Design Projects

John F. Wakerly

J. Willey and Sons - 1976

6 - Eletronic Switching theory and Circuits

H. J. Beuscher, A. H. Budlong

Van Nostrand Reinhola Company - 1971

7 - Microprocessors Fundamentals

F. Halsall, P. F. Lister

Pitman Books Lim. - 1980

8 - TTL Data Book

Texas Instruments - 1978

9 - The Semiconductor Memory Book - INTEL

Nilley - Intercience - 1978

10 - The MOS Memory data Book for Engineers
Texas instruments - 1979

11 - SC/MP Techinical Description
National semiconductor corporation - 1976

12 - Microprocessor Assembly Language
Programing Manual
National Semiconductor - 1977

13 - SC/MP Microprocessor Applications Handbook
National semiconductor - 1976

14 - National's SC/MP Microprocessor Family Quick Reference
National semiconductor - 1976

15 - Memory Application Handbook
National semiconductor - 1976

16 - Sistemas Digitais
Apostila - EFEI- 1979

17 - Eletrônica Industrial Aplicada I
Apostila - EFEI - 1980

18 - Eletrônica Industrial Aplicada II
Apostila - EFEI - 1980

19 - Eletrônica Industrial III
Apostila - EFEI - 1980.

INTRODUÇÃO

O desenvolvimento da tecnologia de semicondutores nos últimos anos, aliado ao estímulo da necessidade do manuseio de um cada vez maior número de dados, propiciou o aparecimento de uma série de memórias semicondutores no mercado a baixo custo, alta densidade de armazenamento, altas frequências de trabalho e alta confiabilidade.

O presente trabalho está relacionado exatamente com um tipo especial destas memórias, isto é, as memórias que podem ser programadas de acordo com a necessidade do usuário.

Esta frase "de acordo com a necessidade do usuário" resume a premência de nosso trabalho.

Em continuidade a elaboração e montagem de nosso laboratório de eletrônica digital, nos deparamos com um entrave que é a necessidade de armazenar sequências lógicas que irão promover os ajustes das várias facções distintas de um processo complexo e dinamicamente interligado e controlado por um sistema digital.

Isto pode ser mais facilmente entendido através de um exemplo genérico.

Digamos que para o controle de um determinado processo, sejam necessárias operações aritméticas e lógicas em cima dos dados recebidos do próprio processo, que após serem manuseados, serão devolvidos ao processo na forma de sinais de controle.

Olhando assim o sistema, notamos que as partes referentes ao processo atitmético e lógico, interface de

entrada e interface de saída, existem no mercado disponíveis prontas, porém,a sequência das operações a serem executadas é uma variável do sistema em si, e depende exclusivamente deste.

Assim,devemos então inserir no sistema algo que contenha esta sequência, a fim de que atuando em conjunto com as outras partes, controle o processo.

Esta unidade de armazenamento é justamente esta memória programável pelo usuário, denominadas PROM (programmable read-only memory),este foi exatamente o problema que nos deparou, e foi necessário criar um circuito (viável) para programar estes dispositivos.

O mercado Internacional e mesmo nacional já possuem tais dispositivos, porém devido ao preço bastante elevado e a necessidade de uma distribuição de verba suprindo prioridades, resolvemos projetar e montar tal dispositivo com os componentes já disponíveis em nosso laboratório.

Como a complexidade do sistema exigiria um microprocessador, a escolha recaiu sobre o SC/MP,pois existia um protótipo doado por um convênio de pesquisa a EFEI, em desuso até a presente data.

Aliado a uma filosofia de nosso departamento,de que nada se perde tudo se transforma, resolvemos, então,usando materiais ociosos construir um programador de memórias semicondutoras, especialmente aquelas que podem ser apagadas com o uso do ultra-violeta.

As próximas páginas deste trabalho é o desenrolar dos estudos que nos levou ao protótipo final.

CAPÍTULO I

MEMÓRIAS SEMICONDUTORAS

INTRODUÇÃO:

As memórias semicondutoras se dividem em três categorias: RAM, ROM e SÉRIE de acordo com a devida capacidade funcional.

Com algumas exceções, quaisquer categorias podem ser implementadas com as duas maiores tecnologias em semicondutores: MOS e bipolar.

A figura 1 mostra esta divisão:

As exceções são as memórias EPROM (memórias ROM reprogramáveis) e CCD (dispositivo de carga-acoplada), as quais são implementadas apenas com tecnologia MOS.

MEMÓRIAS RAM:

As memórias de escrita e leitura se desenvolveram muito nos últimos tempos, auxiliadas pela aplicação da tecnologia MOS. Liderando as vantagens da aplicação desta tecnologia, apresenta-se o aumento de densidade de componentes MOS.

Em poucos anos a capacidade de integração de dispositivos MOS quadruplicou-se.

MEMÓRIAS ROM:

As memórias de leitura somente, do mesmo modo têm sido aperfeiçoadas e melhoradas enormemente nestes últimos anos.

A inovação fundamental está ligada ao aperfeiçoamento da capacidade de programar e poder reprogramar a gravação, isto é, as memórias PROM e EPROM.

Estes dois dispositivos MOS desenvolveram

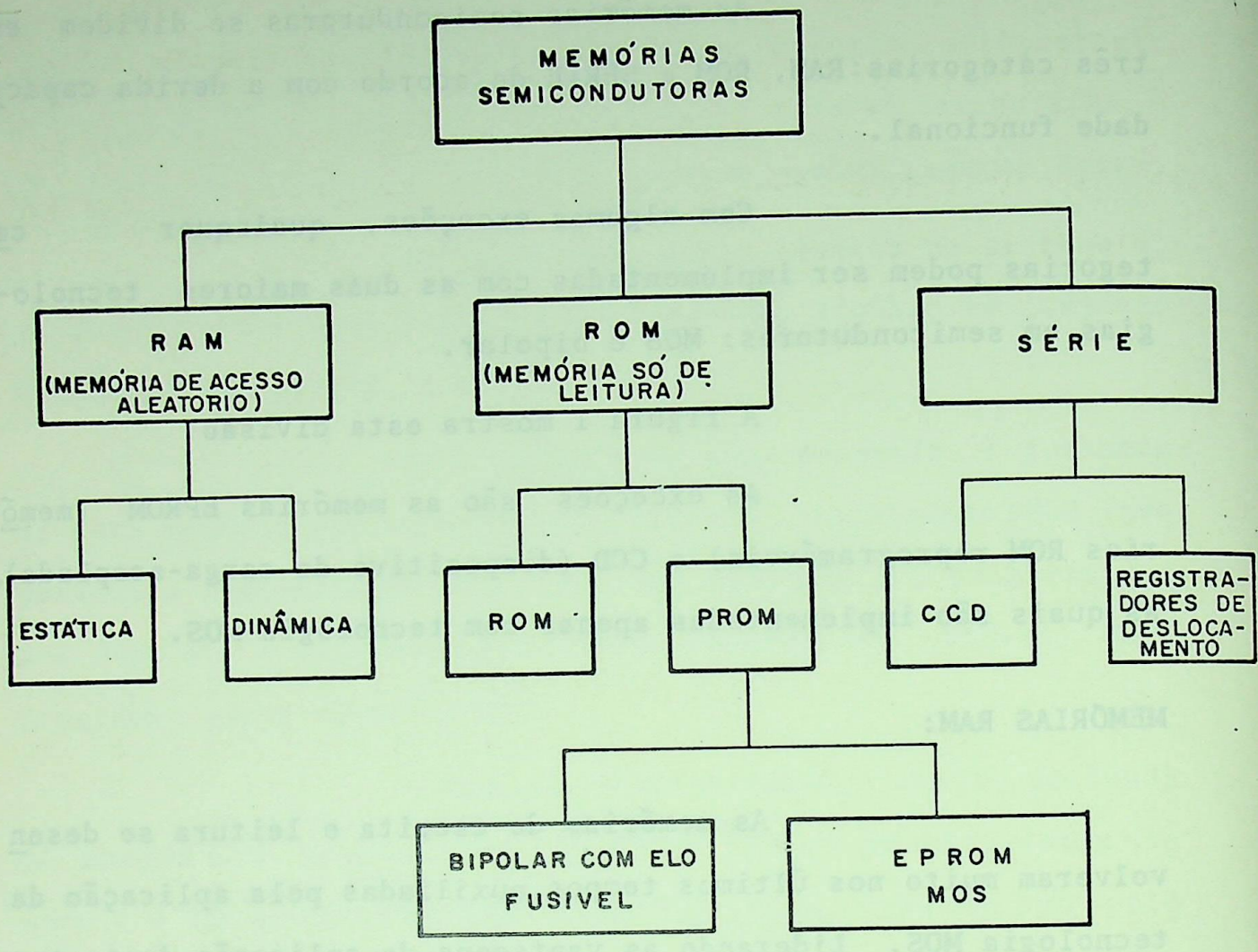

MEMÓRIAS SEMICONDUTORAS
RAM (MEMÓRIA DE ACESSO ALEATORIO)
ROM (MEMÓRIA SÓ DE LEITURA)
SÉRIE
ESTÁTICA
DINÂMICA
ROM
PROM
CCD
REGISTRADORES DE DESLOCAMENTO
BIPOLAR COM ELO FUSIVEL
EPROM MOS

FIG. 1

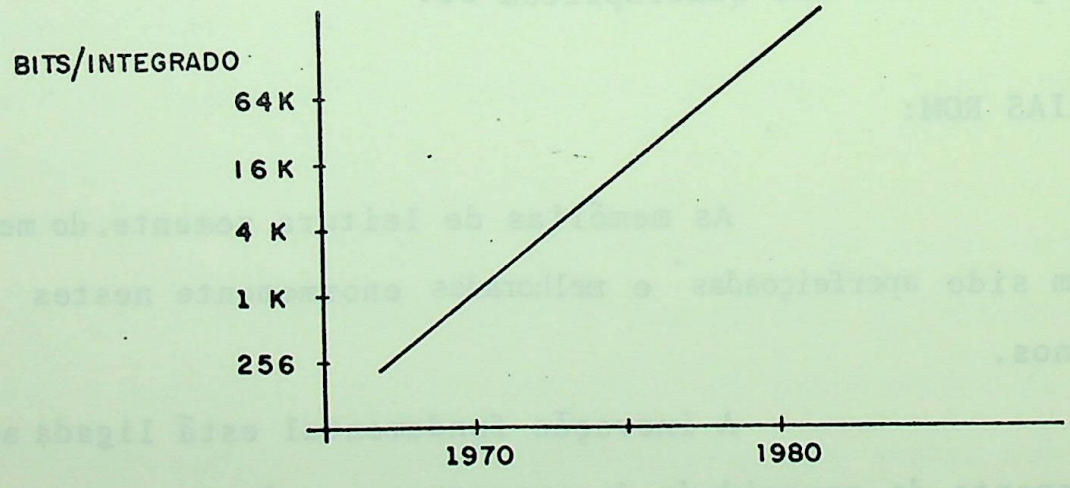

BITS/INTEGRADO
64K
16K
4K
1K
256
1970
1980

FIG. 2

a aceitação e aplicação das memórias de leitura somente.

Disponível no mercado existe uma série de disponibilidades como é o caso das memórias reprogramáveis com o uso da luz ultra-violeta.

Dentro destes dispositivos se encontra a memória UV EPROM 2708 pela qual dedicaremos um estudo mais detalhado posteriormente.

MEMÓRIAS SÉRIE

Um produto sensacional lançado recentemente no mercado diz respeito aos dispositivos de carga - acoplada (CCD).

A alta densidade e baixo custo, aumentaram sua aplicação nos sistemas com microprocessadores.

A este dispositivo iremos deixar o tempo e a necessidade nos coloquem na obrigação de dedicar horas de estudo e experiências na trilha da compreensão de seu comportamento.

CAPÍTULO II

A MEMÓRIA 2708

A memória 2708 é um dispositivo reprogramável de leitura somente (ROM), contendo 8192 bits (1024 x 8), chamado na literatura técnica de EPROM.

É encapsulada em um integrado de 24 pinos, o qual possui uma abertura transparente que permite a incidência da luz ultra violeta sobre a pastilha semicondutora, quando está para ser reprogramada.

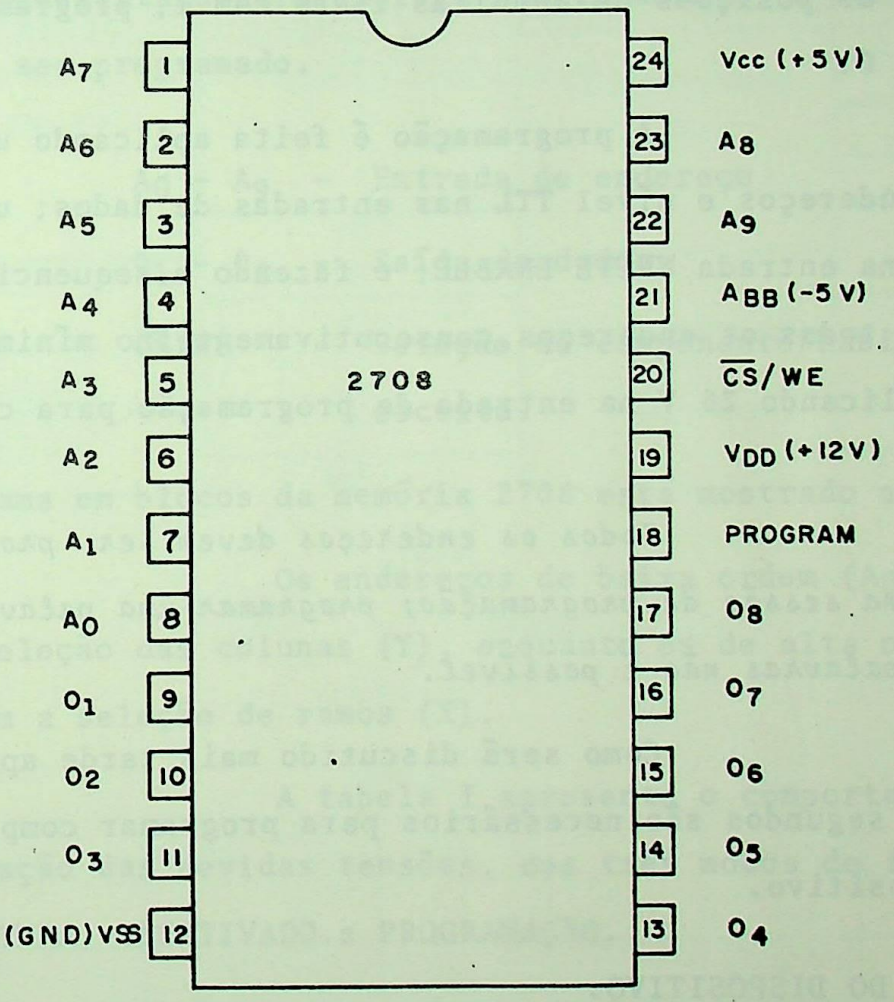


FIG. 1

O tempo máximo de acesso é 450 ns. O componente exige três fontes de alimentação ±5 V e +12 V.

Enquanto está sendo programado, um pulso de 26 V é necessário no pino de programação.

Os endereços de entrada e os dados de saída são compatíveis com a tecnologia TTL.

As saídas de dados são TRI-STATE para facilitar a expansão da memória.

Inicialmente, e após cada apagamento de programação, as posições de memórias ficam com 1, programando, é introduzido 0.

A programação é feita aplicando um nível TTL nos endereços e nível TTL nas entradas de dados; um sinal de +12 V na entrada WRITE ENABLE; e fazendo o sequenciamento através de todos os endereços, consecutivamente, no mínimo por 100 vezes, aplicando 26 V na entrada de programação para cada endereço.

Todos os endereços devem ser programados durante uma sessão de programação; programar uma palavra ou um bloco de palavras não é possível.

Como será discutido mais tarde aproximadamente 100 segundos são necessários para programar completamente o dispositivo.

DESCRIÇÃO DO DISPOSITIVO:

O componente é encapsulado numa embalagem de 24 pinos como mostra a figura 1.

O pino de programação (18) recebe 26 V du

rante o tempo de programação da memória, e durante o ciclo de leitura deve ser conectado a V_{SS} (GND), mantido no nível baixo (V_{IL}).

O pino 20 ($\overline{CS}$/WE), executa três funções: quando está no nível baixo V_{IL} (OV) o componente é selecionado para operação normal.

Quando está no nível alto V_{IH} (3 V min) o componente é desativado e as saídas permanecem em alta impedância.

Quando está em V_{IHW} (11,4 V min) o componente é habilitado para escrita (WRITE ENABLE) e permanece pronto para ser programado.

$A_0 - A_9$ - Entrada de endereço

$O_1 - O_8$ - Saída de dados

$\overline{CS}$/WE - Seleção do componente/habilitação de escrita.

O diagrama em blocos da memória 2708 está mostrado na fig. 2.

Os endereços de baixa ordem (A_0-A_3) executam a seleção das colunas (Y), enquanto os de alta ordem (A_4-A_9) executam a seleção de ramos (X).

A tabela I apresenta o comportamento com a aplicação das devidas tensões, dos três modos de funcionamento: LEITURA, DESATIVADO e PROGRAMAÇÃO.

DESCRIÇÃO DA CÉLULA:

A alma do CI-2708 é um simples transistor, implementado com duas camadas de poli-silicio.

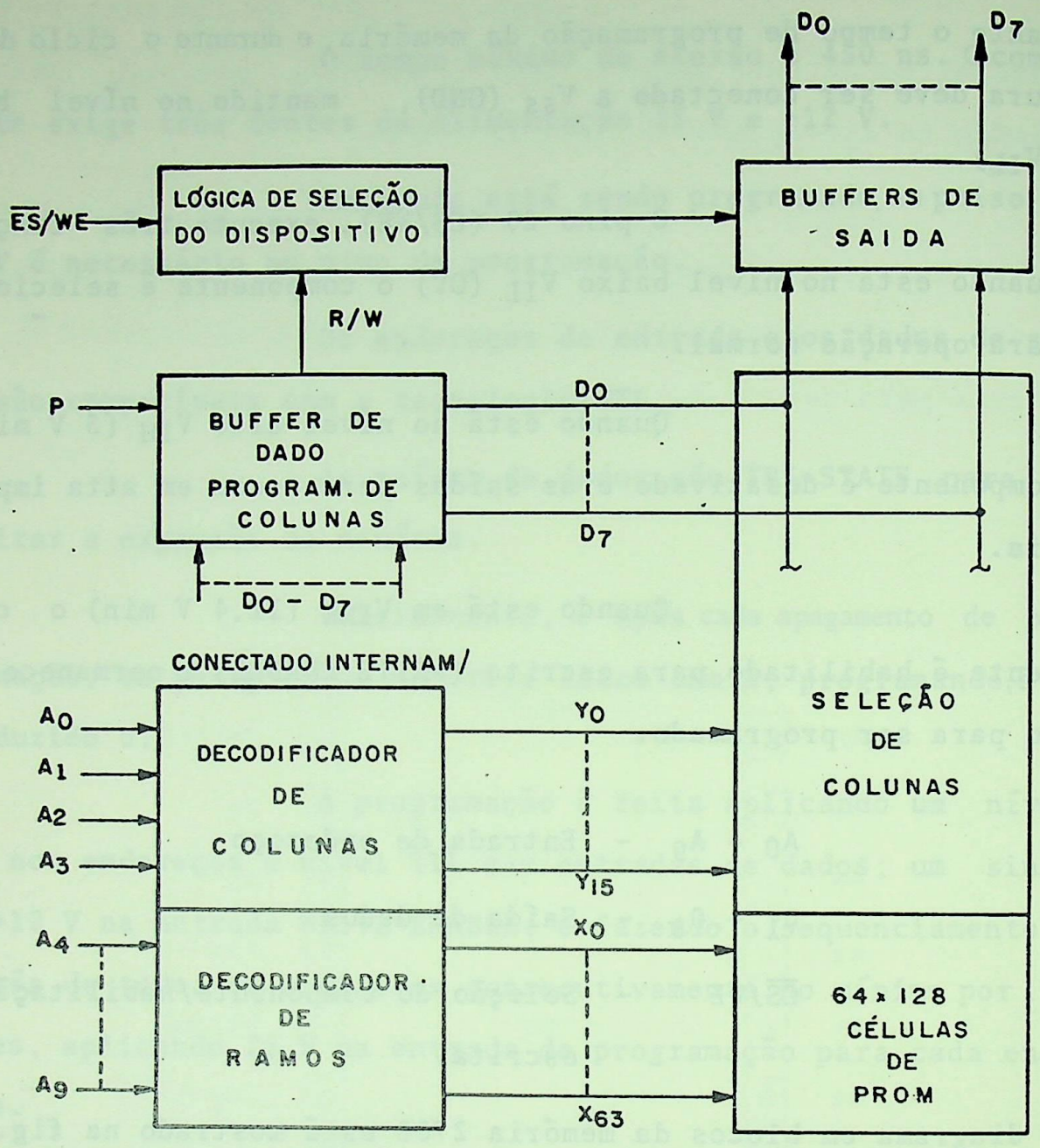
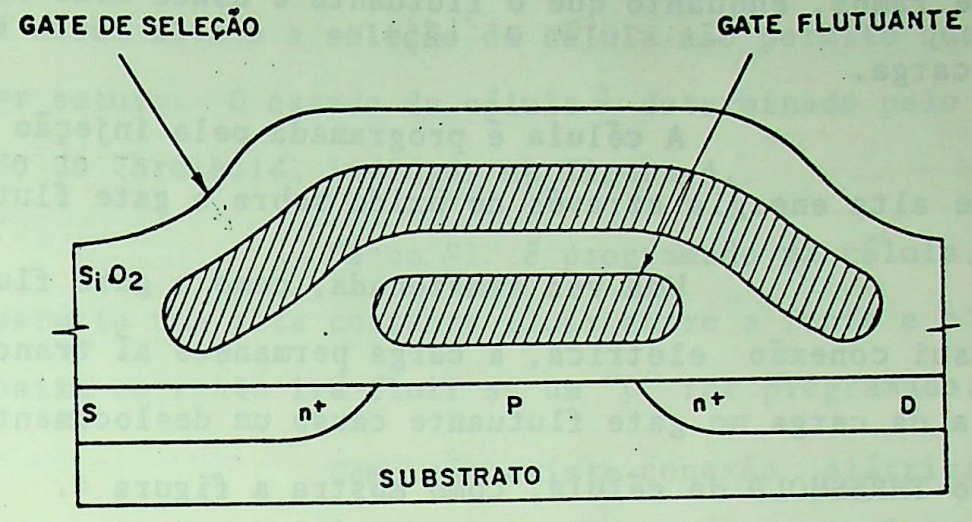


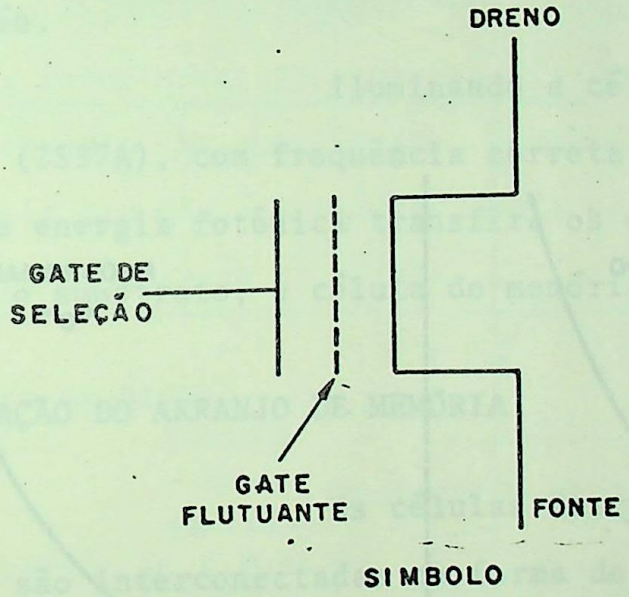
FIG. 2

TABELA I

FUNÇÃO	DADO ENT /SAÍDA	ENT. ENDER.	V_{SS} (GND)	PRGM.	V_{DD} (ALIMENTAÇÃO)	$\overline{CS}$/ WE	V_{BB} (ALIMENTAÇÃO)	V_{CC} (ALIMENT.)
PINO	9 - 11 13 - 17	1 - 7 23 , 22	12	18	19	20	21	24
MODO								
LEITURA	D_{OUT}	A_{IN}	GND	GND	+ 12V	V_{IL}	− 5V	+ 5V
DESATIVADO	ALTA IMPED.	—	GND	GND	+ 12 V	V_{IH}	− 5V	+ 5V
PROGRAMAÇÃO	D_{IN}	A_{IN}	GND	PULSO + 26V	+ 12V	V_{IHW}	− 5V	+ 5V



SESSÃO TRANSVERSAL



SIMBOLO

FIG. 3

A célula consiste de um gate flutuante como fundo e um gate por cima de seleção, como mostra a figura 3.

O gate de cima está conectado ao decodificador de ramos, enquanto que o flutuante é usado como memorizador de carga.

A célula é programada pela injeção de elétrons de alta energia através do óxido sobre o gate flutuante.

Uma vez armazenada, como o gate flutuante não possui conexão elétrica, a carga permanece aí trancada. A presença da carga no gate flutuante causa um deslocamento para a região THRESHOLD da célula, como mostra a figura 4.

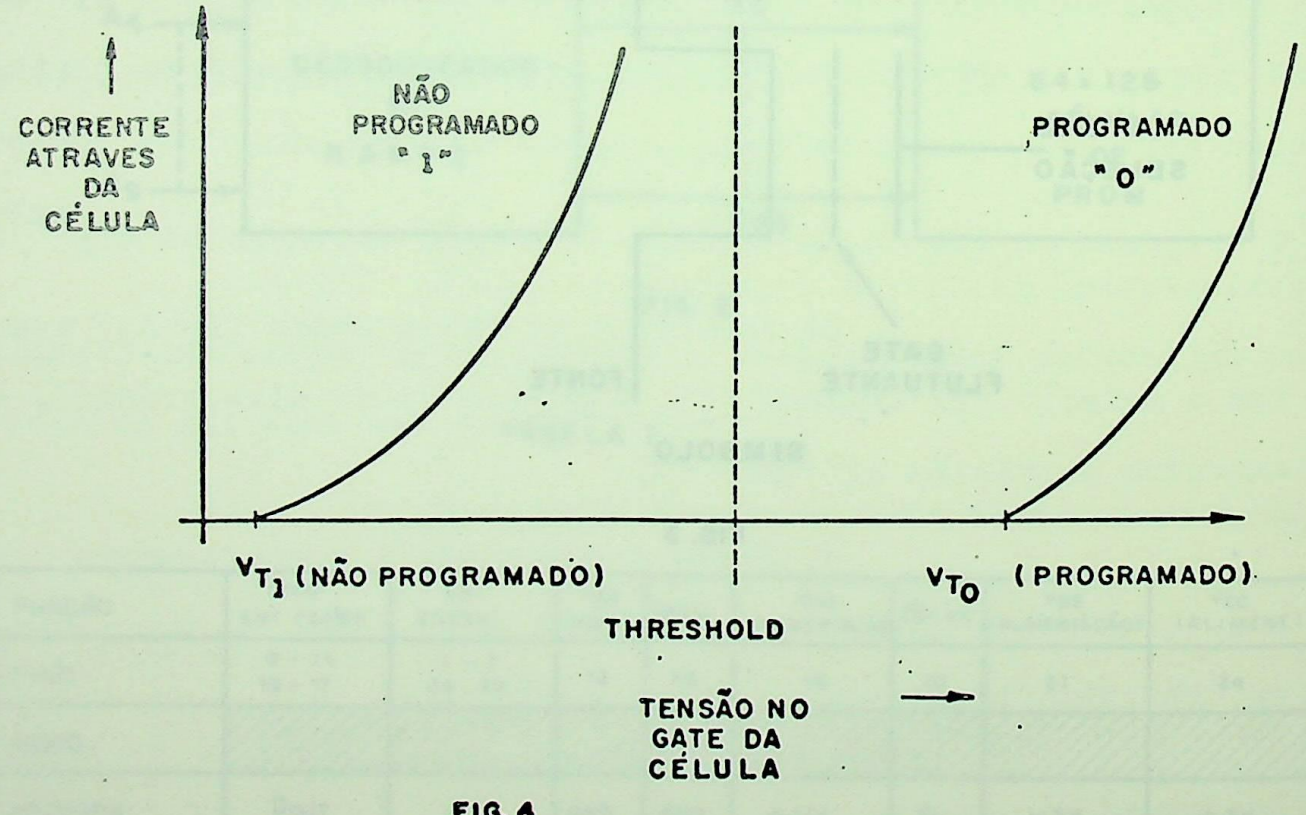


FIG. 4

Itajubá

No estado inicial, a célula possui um baixo nível Threshold e seleção da célula, fazendo o transistor saturar.

A programação desloca o nível Threshold para um nível alto e a seleção da célula não permite que o transistor sature. O estado da célula é determinado pelo exame do estado do Threshold, indicado na figura 4.

Se um "1" é programado na célula, a seleção permite uma alta corrente fluir entre a fonte e o dreno, e uma baixa corrente irá fluir se um "0" for programado.

Como não existe conexão elétrica no gate flutuante, o apagamento deve ser feito sem conexões elétricas também.

Iluminando a célula com uma luz ultra violeta (2537A), com frequência correta e duração suficiente para que a energia fotônica transfira os elétrons, através do óxido para o substrato, a célula de memória é apagada.

OPERAÇÃO DO ARRANJO DE MEMÓRIA

As células descritas no parágrafo anterior são interconectadas na forma de uma matriz 64 x 128, como mostra a figura 5. O acesso para uma célula particular é como segue:

Quando o endereço do ramo estabiliza, um ramo é selecionado, ativando a linha de ramos para todas as 128 células do ramo, O endereço da coluna conecta 8 das 128 colunas em seus respectivos amplificadores.

A linha de ramo providencia polarização pa

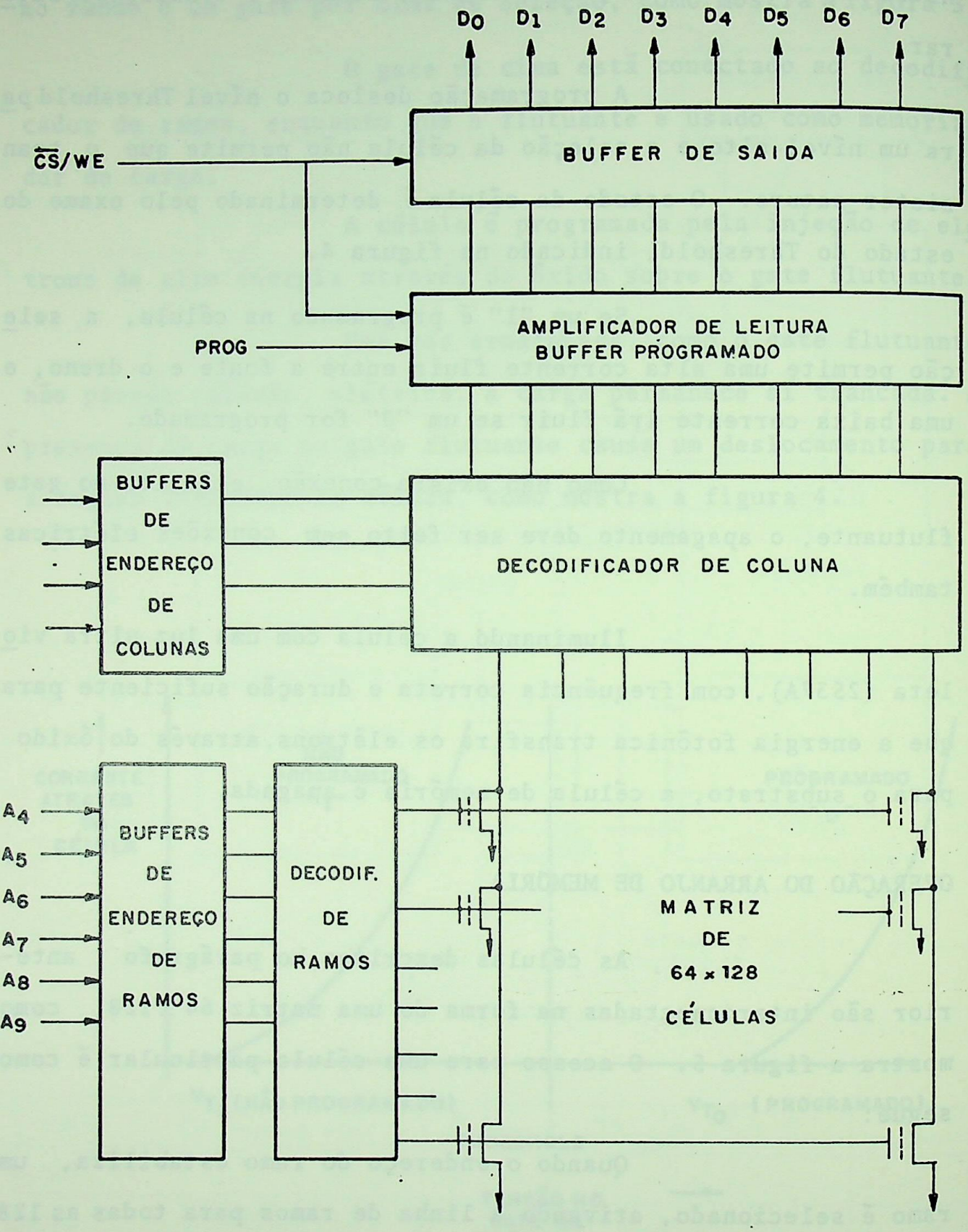

D0
D1
D2
D3
D4
D5
D6
D7
BUFFER DE SAIDA
CS/WE
AMPLIFICADOR DE LEITURA
BUFFER PROGRAMADO
PROG
BUFFERS
DE
ENDEREÇO
DE
COLUNAS
DECODIFICADOR DE COLUNA
A4
A5
A6
A7
A8
A9
BUFFERS
DE
ENDEREÇO
DE
RAMOS
DECODIF.
DE
RAMOS
MATRIZ
DE
64 x 128
CÉLULAS

FIG. 5

ra todos os gates, superiores numa particular seleção de ramos e, dependendo do estado da célula, as linhas de coluna estarão no nível pre-carregado (para um zero), ou descarregado, colocando a linha de coluna no nível baixo (para um 1).

Para uma rápida seleção do integrado até a saída (tempo de atraso t_{CO} = 120 ns), todos os amplificadores são ligados quando o dispositivo é selecionado, e então $\overline{CS}$/WE será baixo (V_{IL}); aqueles não selecionados são desligados e os remanecentes oito amplificadores convertem a carga da linha de coluna em nível TTL pelos buffers de saída.

Durante a programação a seleção de ramos e linhas de coluna são pulsadas a aproximadamente 26 V, e o gate flutuante é carregado. É a presença dos pulsos de 26 V nos gates interconectados que fazem: *Todos os endereços devem ser programados sequencialmente, programar uma palavra ou um grupo de palavras não é permitido, pois transitórios podem aparecer e alterar a carga da célula.*

BUFFER DE SAÍDA

O circuito equivalente do bufer de saída é mostrado na figura 6.

Consiste de um par de transistores MOS, conectados em push-pull.

$\overline{CS}$ habilita ambos os transistores quando baixo, quando alto a saída de ambos é desligada, providenciando uma saída TRI-STATE.

A saída provê: V_{OL} = 0,45 V com I_{OL} = 1,6 m A

V_{OH} = 2,4 V com I_{OH} = -1 m A

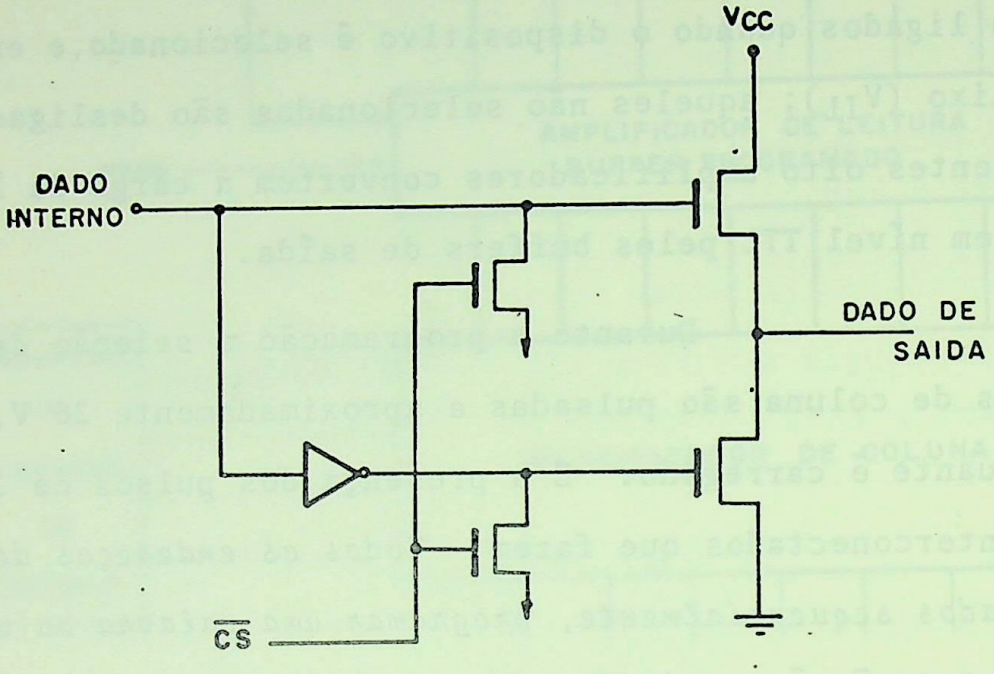


FIG.6

CARACTERÍSTICA DC:

Somente as características que requerem especial atenção serão tratadas.

A tabela II fornece estas características. A memória 2708 requer três fontes de alimentação, +12 V e ± 5V, mantendo as especificações com variações de até 5% no valor nominal destas fontes. Os valores máximos encontrados nas folhas de dados não devem ser excedidos em nenhuma fase da operação do componente incluindo a programação.

MODO LEITURA:

Os limites no valor das correntes estão mostradas na tabela II, apresentando a condição de pior caso, isto é, $T_A = 0°C$ e $\overline{CS}/WE = 5$ V.

As correntes I_{DD}, I_{CC} e I_{BB} indicam as máximas correntes drenadas das fontes.

Estas entradas não podem drenar simultaneamente estes valores máximos de corrente.

Os endereços são compatíveis com a lógica TTL, requerendo V_{IL} entre V_{SS} e 0,65 V, e V_{IH} entre 3 V e $V_{CC}+1$. Um cuidado a ser tomado em usar níveis TTL apropriados ou resistores limitadores para V_{CC}.

TABELA II

Símbolo	Parâmetro	Min.	Tip.	Max	Unid.
I_{LI}	Corrente de entrada endereço e seleção de dispositivo		1	10	µA
I_{LO}	Corrente da saída		1	10	µA
I_{DD}	Corrente da Fonte V_{DD}		50	60	mA
I_{CC}	Corrente da Fonte V_{CC}		6	10	mA
I_{BB}	Corrente da Fonte V_{BB}		30	45	mA
V_{IL}	Tensão de entrada baixa	V_{ss}		0,65	V
V_{IH}	Tensão de entrada alta	3		$V_{CC}+1$	V
V_{OL}	Tensão de saída baixa (I_{OL} = 1,6 mA)			0,45	V
$V_{OH}(1)$	Tensão de saída alta (I_{OH} = -100µA)	3,7			V
$V_{OH}(2)$	Tensão de saída alta (I_{OH} = -1mA)	2,4			V
P_D	Dissipação de potência			800	mW

Durante o modo de leitura, a entrada $\overline{CS}$/WE (pino 20) é também compatível com lógica TTL.

As saídas também compatíveis com TTL produzem um V_{OL} de 0,45 V (máximo) com 1,6 mA e V_{OH} de 3,7 V com -100 µA de capacidade (ou 2,4 V com -1mA).

A figura 7 apresenta a condição típica de corrente absorvida como função da tensão de saída e da temperatura, para aplicações requerendo correntes maiores que I_{OL} normal.

A figura 8 ilustra vários pontos de funcionamento da memória 2708.

Primeiro como todo dispositivo MOS, a corrente da fonte diminui com o aumento da temperatura.

O segundo ponto é que a corrente da fonte aumenta quando ele não esta selecionado, isto é, $\overline{CS}$/WE = alto= V_{IH}. A razão para isto é que a fim de conseguir tempos rápidos de 120 n_s, todos os decodificadores e estágios de saída são ligados quando $\overline{CS}$/WE está em V_{IH}.

O gráfico também ilustra que a tensão V_{DD} é a mais indicada para uso no chaveamento reduzindo a potência.

Naturalmente, se o sistema permitir, $\overline{CS}$/WE pode ser ligado a V_{SS} para reduzir o consumo.

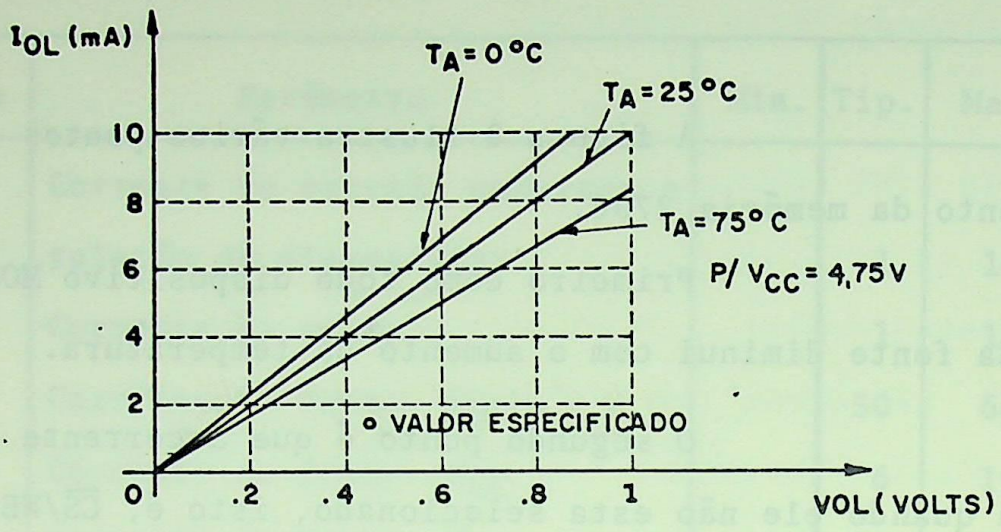

I_{OL} (mA)
T_A = 0 °C
T_A = 25 °C
T_A = 75° C
P/ V_{CC} = 4,75 V
10
8
6
4
2
0
.2
.4
.6
.8
1
o VALOR ESPECIFICADO
VOL (VOLTS)

FIG. 7

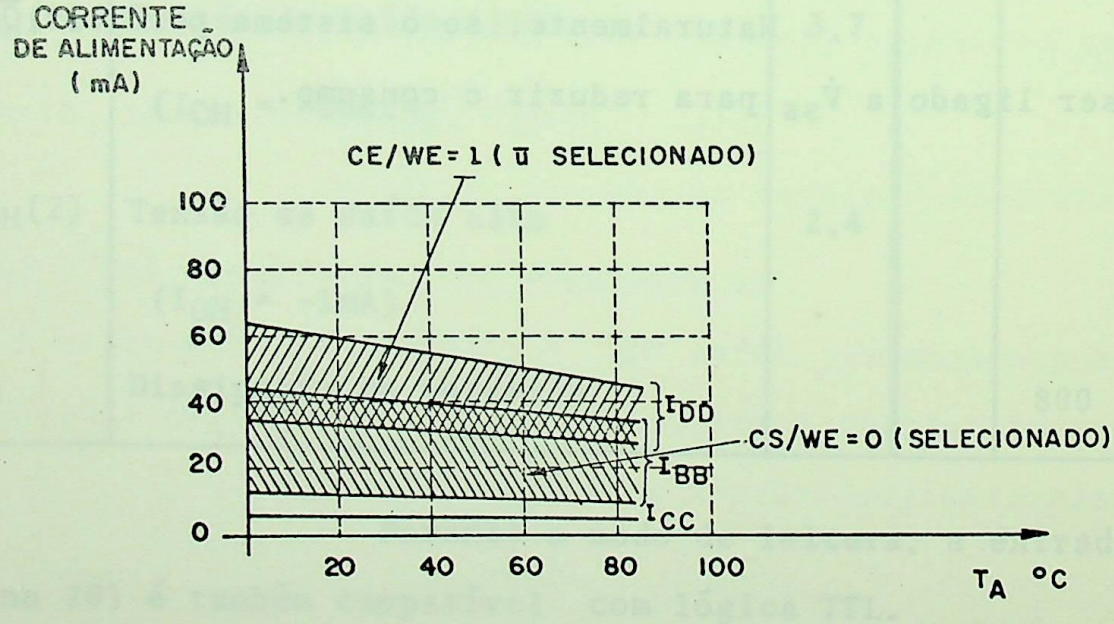

CORRENTE DE ALIMENTAÇÃO (mA)
CE/WE = 1 ($\overline{U}$ SELECIONADO)
100
80
60
40
20
0
I_{DD}
CS/WE = 0 (SELECIONADO)
I_{BB}
I_{CC}
20
40
60
80
100
T_A °C

FIG. 8

MODO DE PROGRAMAÇÃO

As entradas de endereço e dados são compatíveis com nível baixo durante a programação, com as mesmas condições de V_{IL} e V_{IH} do modo de leitura.

As características D.C. para programação são mostradas na Tabela III.

TABELA III

Símbolo	Parâmetros	Min	Típ.	Max	Unid.
I_{LI}	Entrada de corrente endereço e $\overline{CS}$/WE (absorvida)			10	µA
I_{IPL}	Pulso de corrente para program.	(fornecido)		3	mA
I_{IPH}	Pulso de corrente para program.	(absorvido)		20	mA
I_{DD}	Corrente de alimentação V_{DD}		50	65	mA
I_{CC}	Corrente de alimentação V_{CC}		6	10	mA
I_{BB}	Corrente de alimentação V_{BB}		30	45	mA
V_{IL}	Entrada p/nível baixo	V_{ss}		0,65	V
V_{IH}	Entrada p/nível alto (dados e endereços)	3		V_{cc}+1	V
V_{IHW}	Nível alto p/ $\overline{CS}$/WE	11,4		12,6	V
V_{IHP}	Pulso de Program. nível alto	25		27	V
V_{ILP}	Pulso de prog.nível alto	V_{ss}		1	V

Para habilitar o comportamento para programação, o pino $\overline{CS}/WE$ deve ser colocado em V_{IHW} (11,4 V). Se o sistema exigir que o componente permaneça na mesma localização tanto para programação quanto para leitura, este pino exigirá três níveis de trabalho:

- V_{IL} de V_{SS} para 0,65 V afim de selecionar o componente para operação de leitura.
- V_{IH} de 3 V para $+V_{CC}+1$ para desativar o componente e colocar as saídas em alta impedância.
- V_{IHW} de 11,4 para 12,6 V habilitando escrever, no componente, ou seja permitir a programação do dispositivo.

Os circuitos para gerar estes três níveis lógicos são encontrados no final deste capítulo.

Durante a operação de programação, as saídas tornam-se os dados de entrada e podem ser tratadas como um barramento TRI-STATE.

O mesmo nível V_{IL} e V_{IH} aplicado aos pinos I/O de dados, pode ser aplicado aos pinos de endereço.

O pulso de programação, aplicado ao pino 18 durante a programação, deve possuir as características de V_{ILP} (V_{SS} até 1 V) e V_{IHP} (26 V ±1 V).

A fonte do pulso de programação deve ser capaz de fornecer o máximo de 20 mA por componente quando um nível alto (V_{IHP}), e absorver a corrente de 3 mA (I_{IPL}).

Esta corrente deverá ser levada em conta' quando projetar o circuito fornecedor do pulso de programação, se um resistor para a terra for usado, a tensão sobre o resis

tor pode superar a tensão V_{ILP} (max) requerida isto é 1 V.

Pode também ser notado que o pulso de programação não apresenta especificação caso V_{IHP} é tomado no valor mínimo (25 V) e V_{ILP} no máximo (1 V), como $V_{IHP} - V_{ILP}$, deve ser no mínimo igual a 25 V.

Existem vários circuitos apresentados no final deste capítulo para prover estes 25 V mínimos necessários entre $V_{IHP} - V_{ILP}$.

CARACTERÍSTICAS A.C.

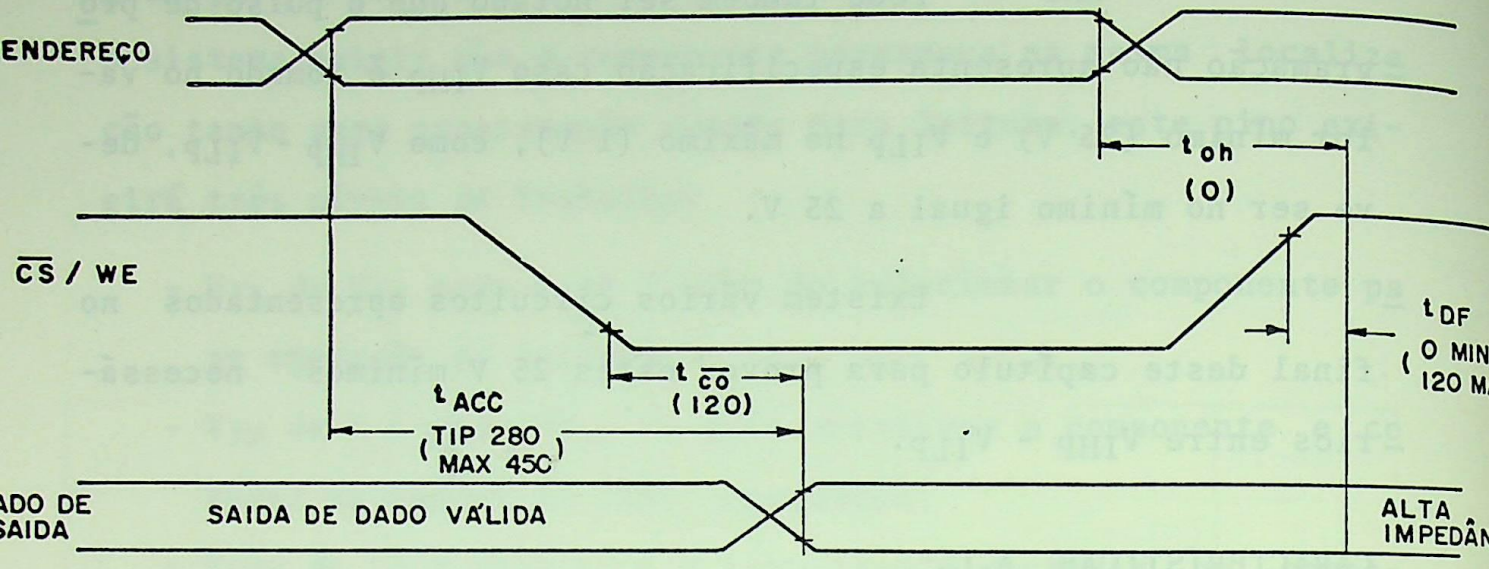
MODO DE LEITURA

A figura 9 apresenta a temporização do modo de Leitura, indicando os tempos máximos e mínimos e vários outros parâmetros de tempo. Uma atenção particular deve ser dispensada ao tempo t_{DF}, desativamento do componente para saídas flutuantes.

Este tempo indica que os BUFFERS de saída da 2708 não são garantidos estarem em alta impedância menos que 120 ns após $\overline{CS}/WE$ alcançar o ponto de 2,8 V.

Se tiver um outro dispositivo ligado a este nó durante este tempo (120 ns), irá drenar uma alta corrente I_{CC}, gerando ruído na fonte e possivelmente reduzindo o nível V_{CC} de modo a tornar outros componentes inoperantes.

O tempo T_{DF} deve também ser considerado quando uma alimentação chaveada (V_{DD}) for usada.



NOTA: OS TEMPOS SÃO EM ns

FIG. 9

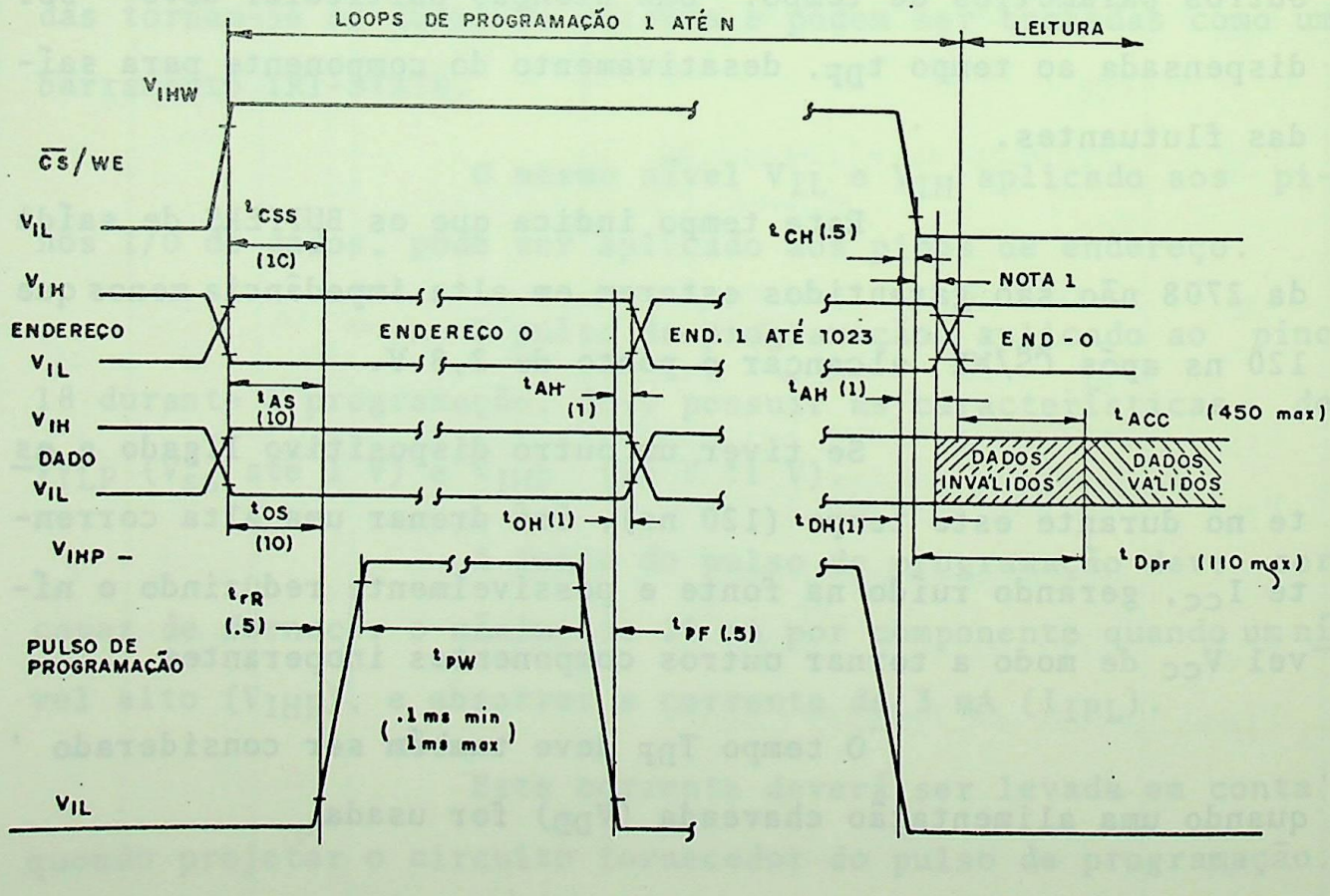


FIG. 10

MODO DE PROGRAMAÇÃO

A figura 10 mostra a temporização de programação, enquanto que a tabela IV apresenta os vários parâmetros AC.

NOTA 1 - A transição de $\overline{CS}$/WE deve ocorrer após a transição do pulso de programação e antes da transição do endereço.

NOTA 2 - Os tempos são em µs.

TABELA IV

Símbolo	Parâmetro	Min.	Tip.	Max	Unid.
T_{AS}	Setup Time de Endereço	10			µs
T_{CSS}	Setup Time de $\overline{CS}$/WE	10			µs
T_{DS}	Setup Time de Dado	10			µs
T_{AH}	Hold Time de Endereço	1			µs
T_{CH}	Hold Time de $\overline{CS}$/WE	.5			µs
T_{DH}	Hold Time de Dado	1			µs
T_{DF}	Atraso entre seleção e Alta Imp.	0		120	ns
T_{DPR}	Atraso entre program. e leitura			10	µs
T_{PW}	Largura do pulso de Prog.	.1		1	ms
T_{PR}	Rise Time do pulso de Program.	.5		2	µs
T_{PF}	Fall Time do pulso de Program.	.5		2	µs

Várias opções de programação são disponíveis na memória 2708.

As formas de onda da figura 10 apresentam a mais eficiente maneira de programação.

Os vários parâmetros são auto-explicativos. Faremos uma explanação dos dois mais importantes somente; o RISE e FALL TIME do pulso de programação (T_{PR} e T_{PF}), respectivamente, que devem ser mantidos dentro do limite de 0,5 a 2µs, a fim de minimizar o efeito transitório de acoplamento. Isto, normalmente, exige uma série de circuitos RC na saída do pulso de programação para modificar o RISE-TIME.

Outro parâmetro importante é a transição do modo programação para o de leitura.

Se a transição do $\overline{CS}$/WE não ocorrer após o final da transição do pulso de programação e antes da transição do endereço, nós internos não estarão descarregados, fazendo a saída indicar dados falsos por vários milisegundos, e mostrado na figura 10 com o nome T_{DPR} (10 µs).

Se a temporização de $\overline{CS}$/WE é difícil de ajustar, providenciando o complemento binário do primeiro endereço para ser verificado, faz com que seja descarregado os nós internos.

PROGRAMAÇÃO

Vários tipos de programadores da memória ROM 2708 são disponíveis no mercado internacional.

Porém, devido as dificuldades de importação, faremos a opção de projetar um. Isto é possível com o uso de um microprocessador, e seguindo a descrição:

- O dispositivo é colocado na operação de

programação, pela colocação da entrada $\overline{CS}/WE$ (pino 20) em V_{IHW} (+12 V). A palavra de endereço é então selecionada do mesmo modo que no modo de leitura.

O dado a ser programado é apresentado na forma de 8 bits, em paralelo, aos pinos de dado de saída (O_1-O_8).

Os níveis lógicos para endereços, dados, e tensões de alimentação são os mesmos que no modo de leituras.

Após os tempos de SET UP de endereço e dado (T_{AS} e T_{DS} Figura 10), um pulso de programação com largura T_{PW} é aplicado ao pino de programação (pino 18). Esta sequência é então repetida para os próximos endereços.

Uma passada através dos 1024 endereços é definida como um LOOP DE PROGRAMAÇÃO. O número de LOOPS de Programação (N) requerido é função da largura do pulso de programação (T_{PW}), de acordo com a fórmula:

$$N \times t_{pw} \geq 100 \text{ ms}$$

onde:

N - número de LOOPS de Programação

t_{pw} - largura do pulso de programação.

A largura do pulso de programação pode variar de 0,1 até 1,0 ms. O número de LOOPS (N) pode variar de no mínimo 100 (t_{pw} = 1,0 ms) e no máximo 1000 (t_{pw} = 0,1 ms) dependendo do valor escolhido para t_{pw}.

Não pode ser aplicado N pulso de programação em um endereço, e então, mudar para o próximo endereço e aplicar de novo N pulsos de programação, Deve ser executado N LOOPS sucessivos através dos 1024 endereços.

De acordo com o diagrama de tempo da figura 10, a programação ótima é conseguida quando:

$t_{css} = t_{as} = t_{ds} = 10\ \mu s$

$t_{pw} = 1,0\ ms$

$t_{ah} = t_{dh} = 1,0\ \mu s$

$t_{pr} = t_{pf} = 0,5\ \mu s$

e o tempo para 1 endereço será:

$$t_{as} + t_{pr} + t_{pw} + t_{pf} + t_{ah} = 1012\ ms$$

ou, para 100 LOOPS e 1024 endereços, o tempo total para programar completamente a memória será

1012 ms/endereço x 100 loops x 1024 endereços ou seja:

103,6 segundos

Notar que o ciclo de trabalho do pulso de programação é aproximadamente 99%. Para qualquer que seja a largura do pulso de programação, as sucessivas passagens por todos os endereços não podem ser eliminadas.

PROGRAMADOR TÍPICO:

A figura 11 mostra o diagrama em blocos de um programador típico, que apresenta todos os requerimentos necessários a programação da memória 2708, assim como facilmente adaptável a um sistema de barramento de um microprocessador, que irá funcionar como fonte de dados. Fazer a entrada de dados via teclado também é possível, embora a programação se torne deve-

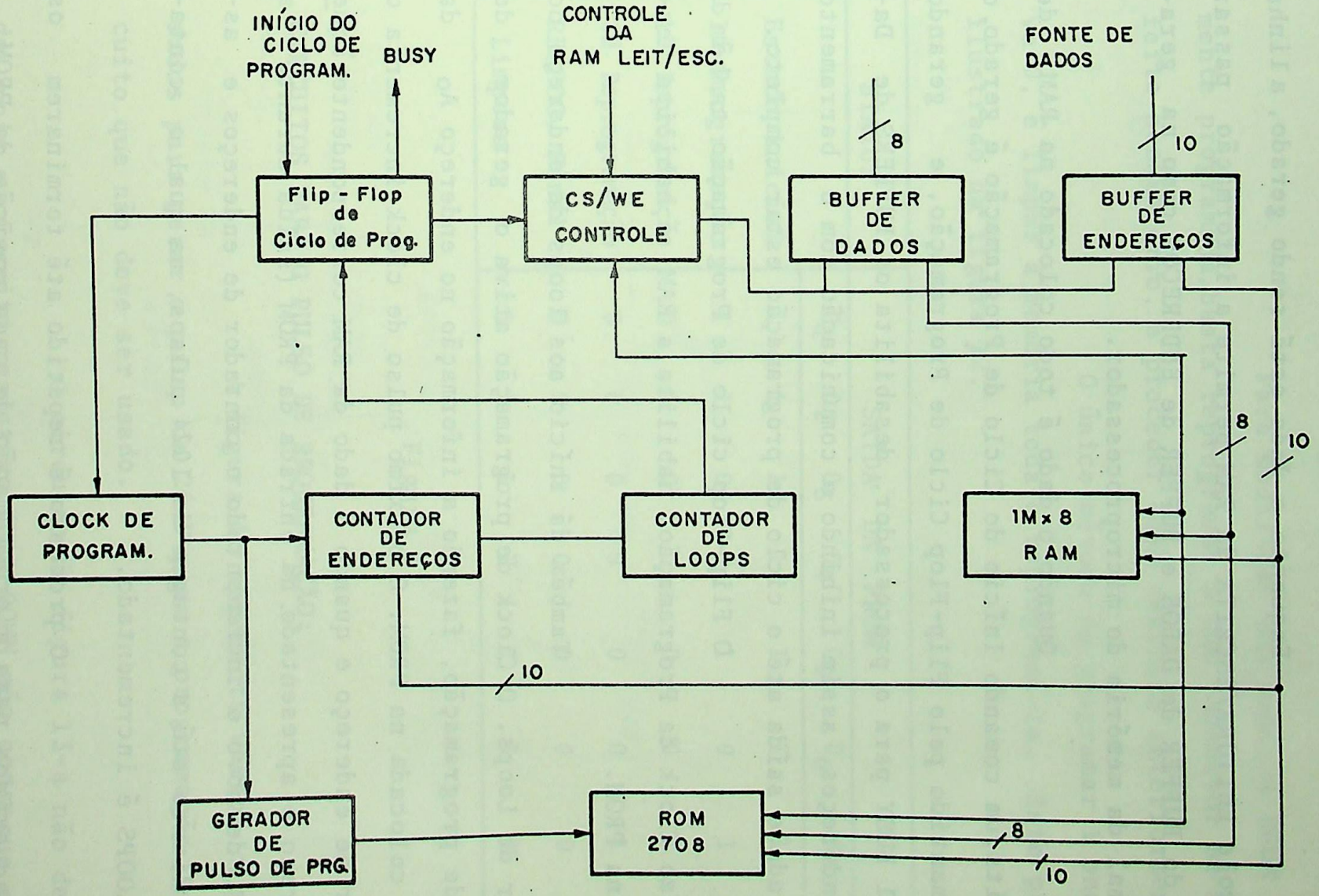


FIG. 11

Escola Federal de Engenharia de Itajubá
Biblioteca
MAUÁ

ras tediosa. A operação do programador é a seguinte:

Enquanto o dado está sendo gerado, a linha de conttole LEITURA/ESCRITA da RAM permite a informação passar através do BUFFER de DADOS e BUFFER de ENDEREÇOS como na geração normal da memória do microprocessador.

Quando o dado é todo colocado na RAM de 1K x 8 bits, um comando Início do Ciclo de Programação é gerado, o qual é mantido pelo Flip-Flop Ciclo de Programação, e gerando um sinal BUSY para o processador, desabilita os BUFFERS de Dados e endereços, assim inibindo a comunicação com o barramento de entrada e saída até o ciclo de programação estar completo.

O Flip-Flop ciclo de Programação também dá início ao Clock da Programação, habilita a RAM, e habilita escrever na PROM.

Também dá início aos loops de endereço do contador de Loops. O Clock de programação ativa o gerador de Pulsos de programação, fazendo a informação no endereço A_0 da RAM ser colocada na PROM. O próximo pulso de clock incrementa o contador de endereço e quando o dado da RAM correspondente àquele endereço e apresentado na entrada da PROM (saída durante a leitura), de novo é incrementado o contador de endereços e assim por diante até a contagem de 1024 pulsos, na qual o contador de LOOPS é incrementado.

O processo é repetido até terminarem os pulsos requeridos para a programação de cada posição da PROM.

Ao final da contagem do contador de Loops é resetado o Flip-Flop de ciclo de Programação.

A memória PROM não pode ser lida ou verificada durante o ciclo de Programação.

Para modificar uma memória PROM parcialmente programada, basta reprogramar a RAM, observar se não foi feito nenhum erro, e proceder como descrito anteriormente.

O único método de programar 1 onde estava 0, é limpar a memória toda e reprogramá-la. Este processo é ilustrado na figura 12.

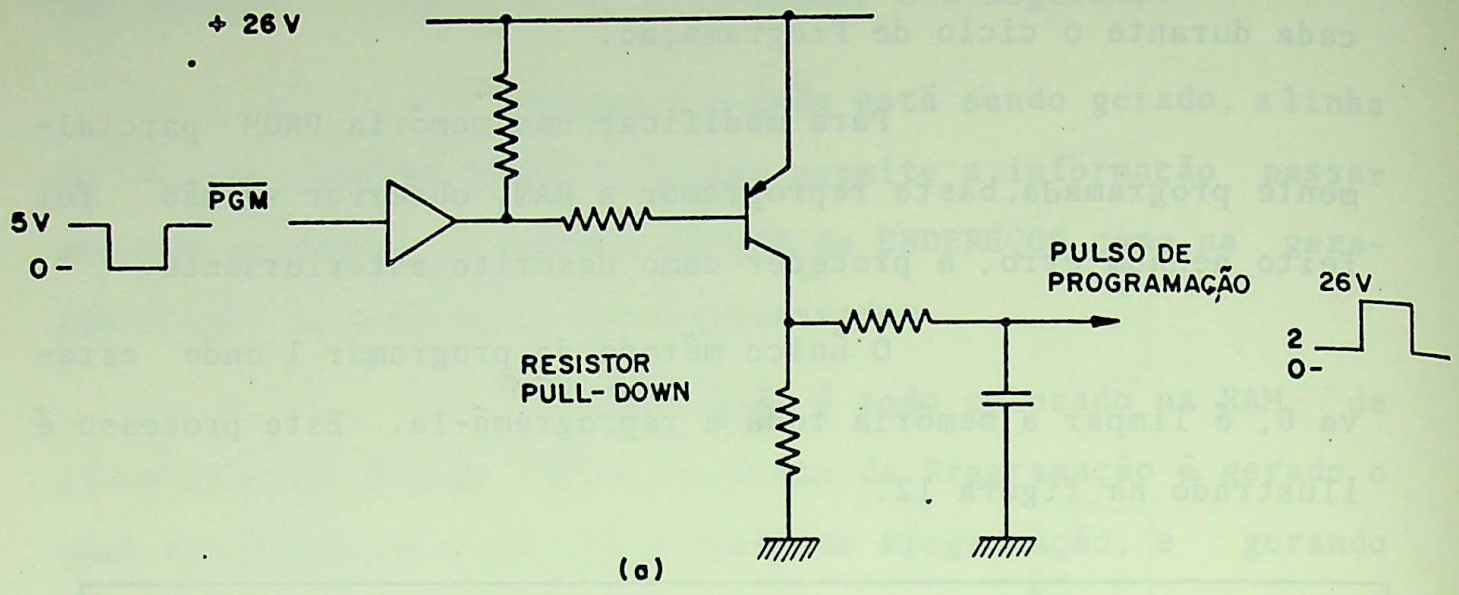
STATUS	SAÍDA DA PROM							
	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
Estado inicial	1	1	1	1	1	1	1	1
1ª Programação	1	1	0	0	1	0	1	0
1ª Reprogramação	0	1	0	0	0	0	1	0
2ª Reprogramação	0	0	0	0	0	0	1	0
REPROG. Final	0	0	0	0	0	0	0	0
Limpeza	1	1	1	1	1	1	1	1

Figura 12

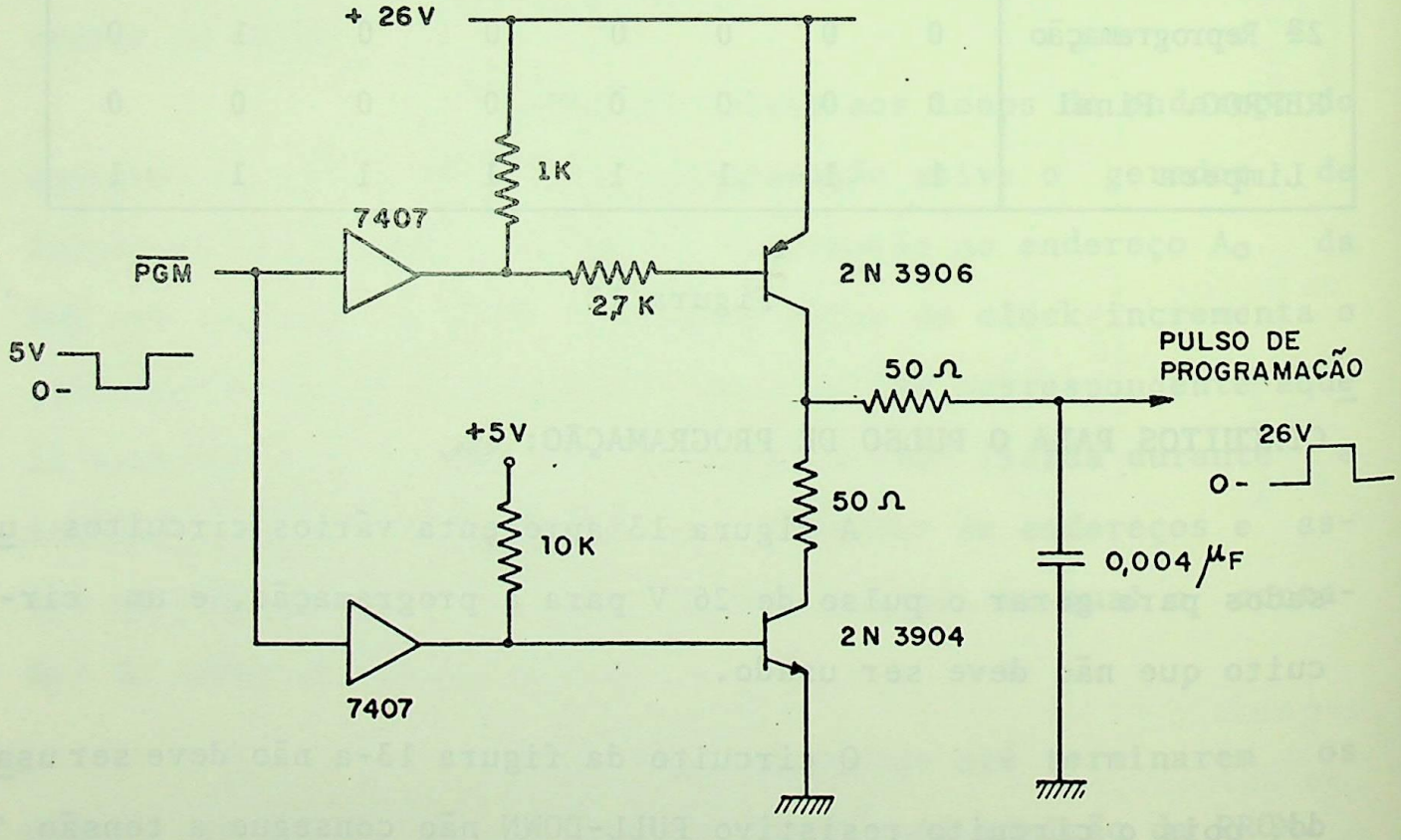
CIRCUITOS PARA O PULSO DE PROGRAMAÇÃO:

A figura 13 apresenta vários circuitos usados para gerar o pulso de 26 V para a programação, e um circuito que não deve ser usado.

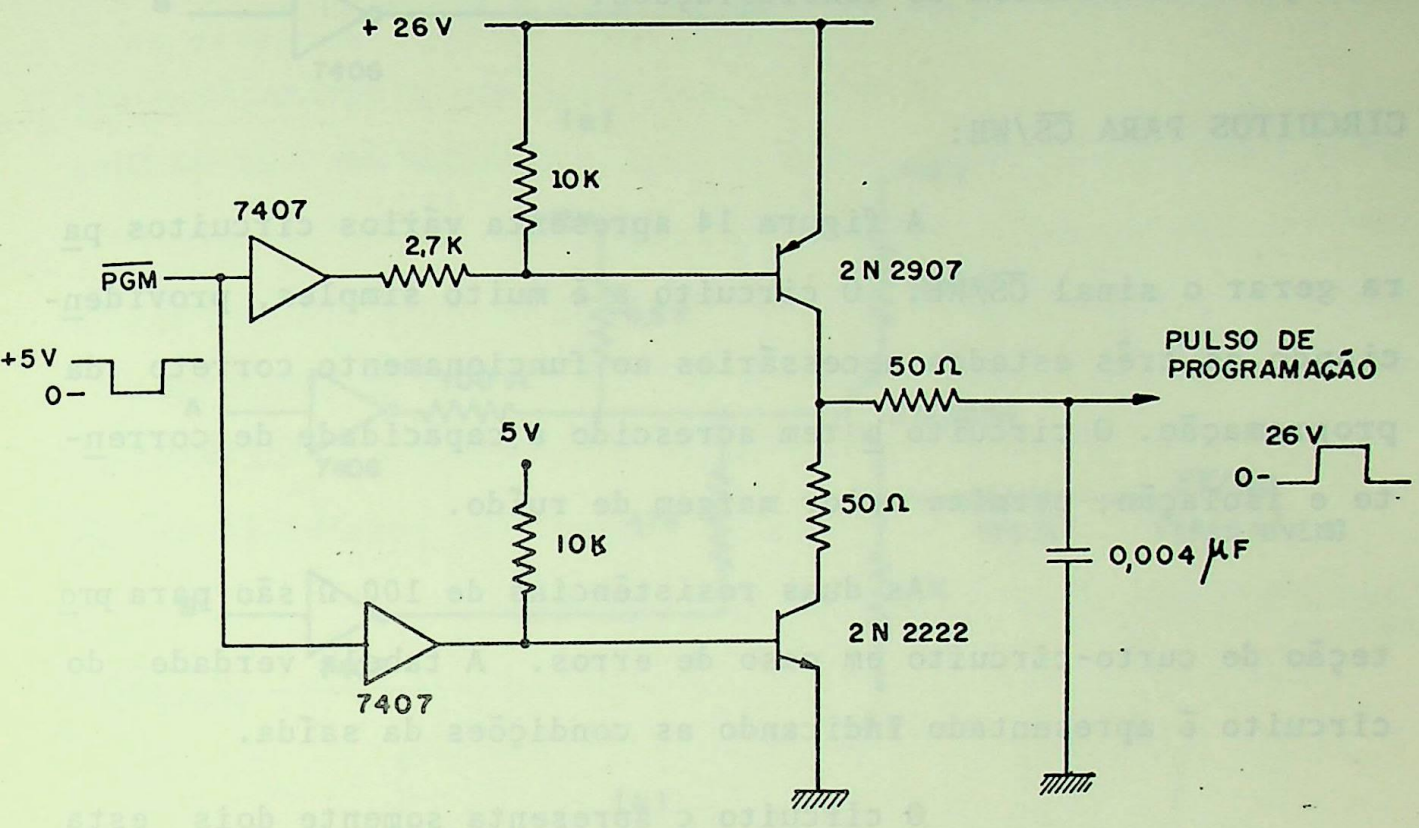
O circuito da figura 13-a não deve ser usado, pois o circuito resistivo PULL-DOWN não consegue a tensão V_{ILP} requerida de 1 V máx, não permitindo assim $V_{IHP} - V_{ILP}$ ser igual ou maior que 25 V.


+ 26 V
PGM
5V
0-
RESISTOR
PULL- DOWN
PULSO DE
PROGRAMAÇÃO
26V
2
0-

(a)


+ 26 V
1K
7407
PGM
2,7 K
2 N 3906
5V
0-
50 Ω
PULSO DE
PROGRAMAÇÃO
+5V
26V
0-
50 Ω
10 K
0,004 μF
2 N 3904
7407

(b)


+ 26 V
10K
7407
2,7K
PGM
2 N 2907
+5V
0-
50 Ω
PULSO DE
PROGRAMAÇÃO
5V
26 V
0-
50 Ω
10K
0,004 μF
2 N 2222
7407

(c)

FIG. 13

Como já mensionado, a razão para isto é que o pino de programação (pino 18) fornece I_{ILP} de 2 mA quando o pulso de programação é nível baixo e $\overline{CS}/WE$ está em 12 V.

Os outros circuitos b e c podem ser usados, pois satisfazem as considerações.

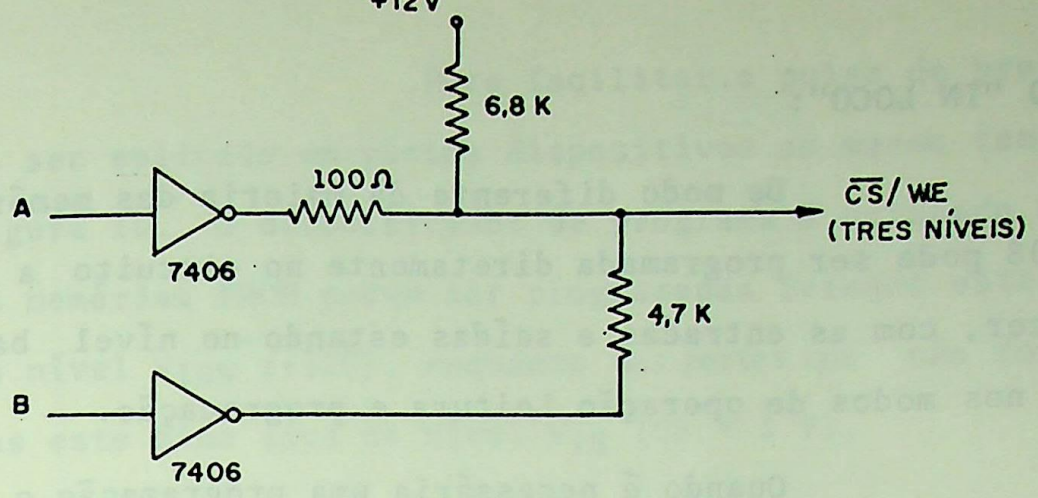
CIRCUITOS PARA $\overline{CS}/WE$:

A figura 14 apresenta vários circuitos para gerar o sinal $\overline{CS}/WE$. O circuito a é muito simples, providenciando os três estados necessários ao funcionamento correto da programação. O circuito b tem acrescido a capacidade de corrente e isolação; permite maior margem de ruído.

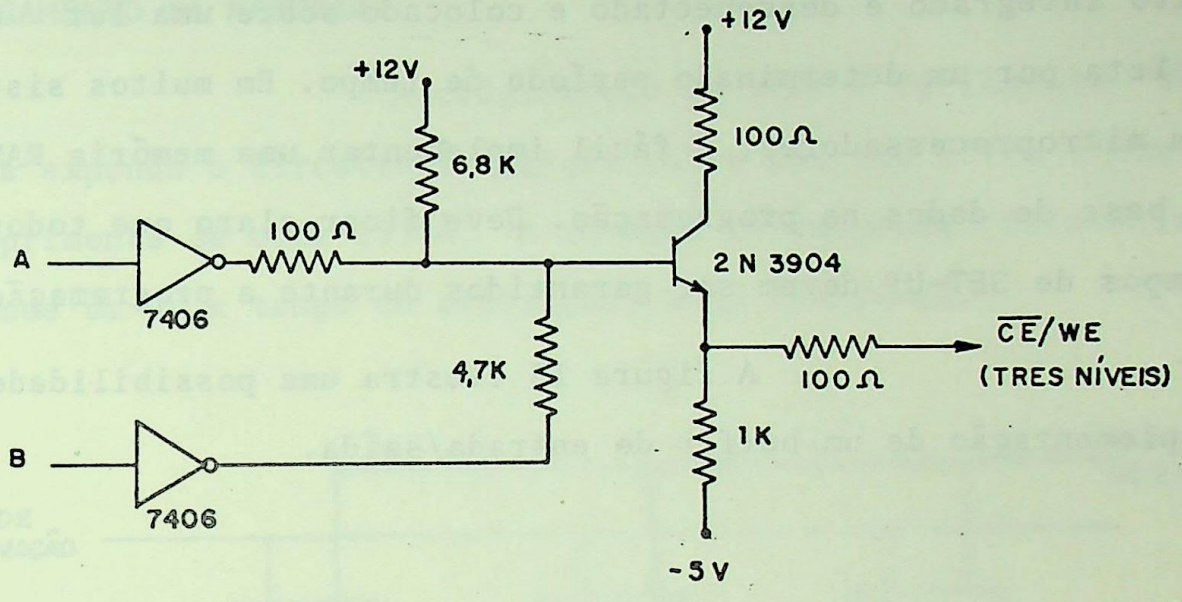
As duas resistências de 100 Ω são para proteção de curto-circuito em caso de erros. A tabela verdade do circuito é apresentado indicando as condições da saída.

O circuito c apresenta somente dois estados de saída, V_{IL}(0v) e V_{IHW}(+12V), para uso na programação e verificação somente, a memória PROM não pode ser desativada com este circuito.

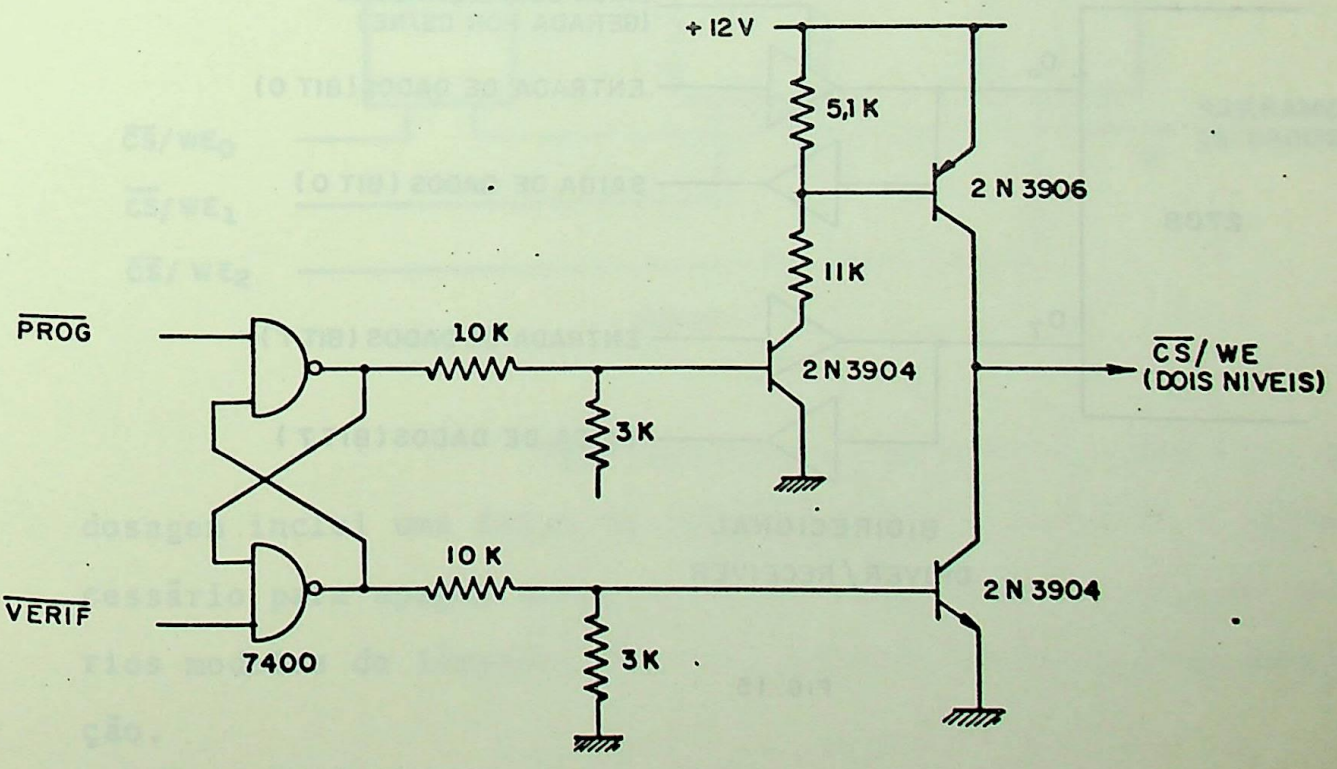
ENTRADA		SAÍDA	FUNÇÃO
A	B	$\overline{CS}/WE$	
1	0	+12 V	PROM habilitada para escrita
1	1	+ 5 V	PROM desativada
0	1	0	PROM ativada


+12V
6,8 K
100 Ω
A
$\overline{CS}$/WE
(TRES NÍVEIS)
7406
4,7 K
B
7406

(a)


+12 V
+12V
100 Ω
6,8 K
100 Ω
A
2 N 3904
7406
$\overline{CE}$/WE
(TRES NÍVEIS)
4,7K
100 Ω
1 K
B
7406
-5 V

(b)


+12 V
5,1 K
2 N 3906
11K
PROG
10 K
2 N3904
$\overline{CS}$/WE
(DOIS NIVEIS)
3K
10 K
2 N 3904
VERIF
7400
3K

(c)

PROGRAMAÇÃO "IN LOCO":

De modo diferente da maioria das memórias PROM, a 2708 pode ser programada diretamente no circuito a que vai pertencer, com as entradas e saídas estando no nível baixo, compatível nos modos de operação Leitura e programação.

Quando é necessária uma programação, o circuito integrado é desconectado e colocado sobre uma luz ultra violeta por um determinado período de tempo. Em muitos sistemas com microprocessadores, é fácil implementar uma memória RAM como base de dados na programação. Deve ficar claro que todos os tempos de SET-UP devem ser garantidos durante a programação.

A figura 15 ilustra uma possibilidade de implementação de um buffer de entrada/saída.

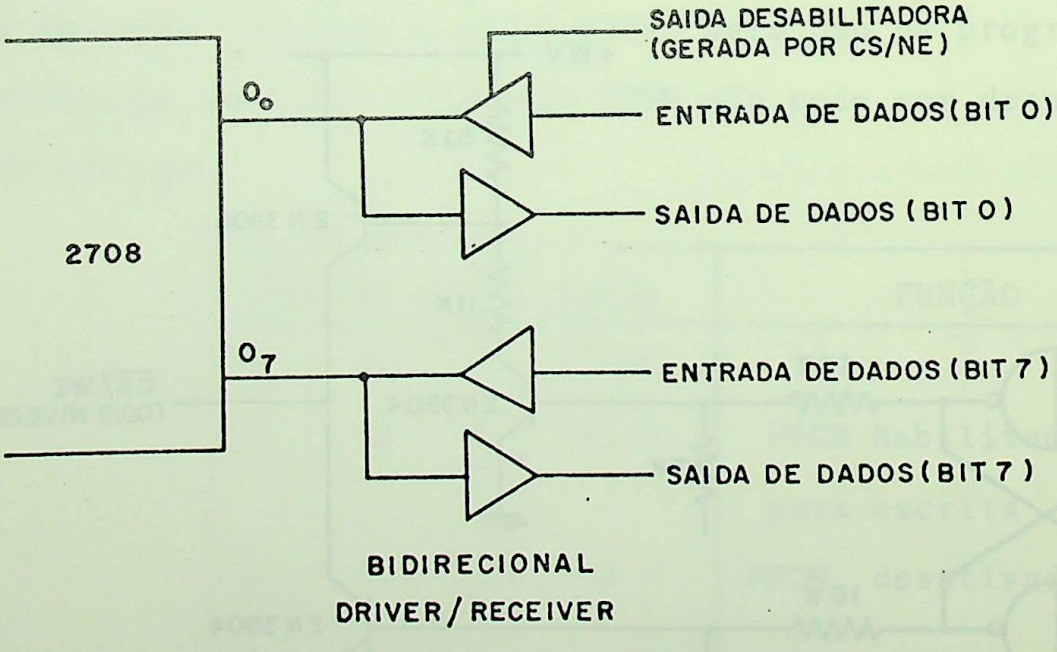


FIG. 15

Para facilitar,o pulso de programação pode ser aplicado em vários dispositivos ao mesmo tempo, conforme figura 16. O decodificador de programa é colocado no pino $\overline{CS}$/WE. As memórias PROM podem ser programadas fazendo este pino ficar no nível V_{IHW} (+12V), enquanto nas partes que não forem progamadas este pino fica no nível V_{IH} (CS = 3 V).

APAGAMENTO DA MEMÓRIA:

A programação da 2708 é facilmente eliminada expondo o circuito integrado sobre uma luz ultra violeta de comprimento de onda 2537Å. A dosagem recomendada (isto é,intensidade de UV x tempo de exposição) é 15 W-seg/cm2.

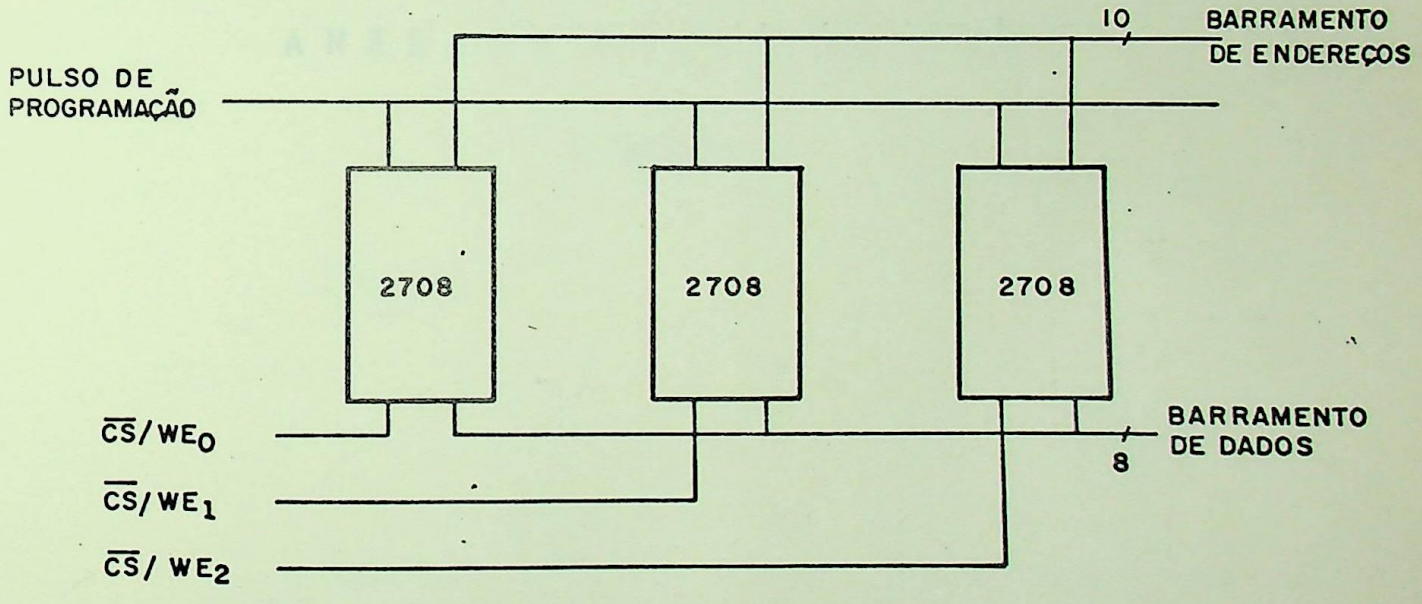


FIG. 16

Para garantir o apagamento completo, esta dosagem inclui uma faixa de segurança de 3 a 4 vezes o tempo necessário para apagar todos os bits. A tabela V apresenta os vários modelos de lâmpadas UV e os respectivos tempos de exposição.

TABELA V

MODELO	POTÊNCIA	TEMPO TÍPICO
S-68	12.000 $\mu W/cm^2$	15 minutos
S-52	12.000 $\mu W/cm^2$	15 minutos
UVS-54	5.700 $\mu W/cm^2$	45 minutos
R-52	13.000 $\mu W/cm^2$	15 minutos
UVS-11	5.500 $\mu W/cm^2$	45 minutos

Para outras lâmpadas o tempo pode ser calculado impiricamente , ou através de medidores de intensidade de UV.

CAPÍTULO III

ANÁLISE GERAL DA CPU

SC/MP

O SC/MP é um microprocessador, encapsulado em uma embalagem de 40 pinos, cujo Lay-out aparece na fig.1.

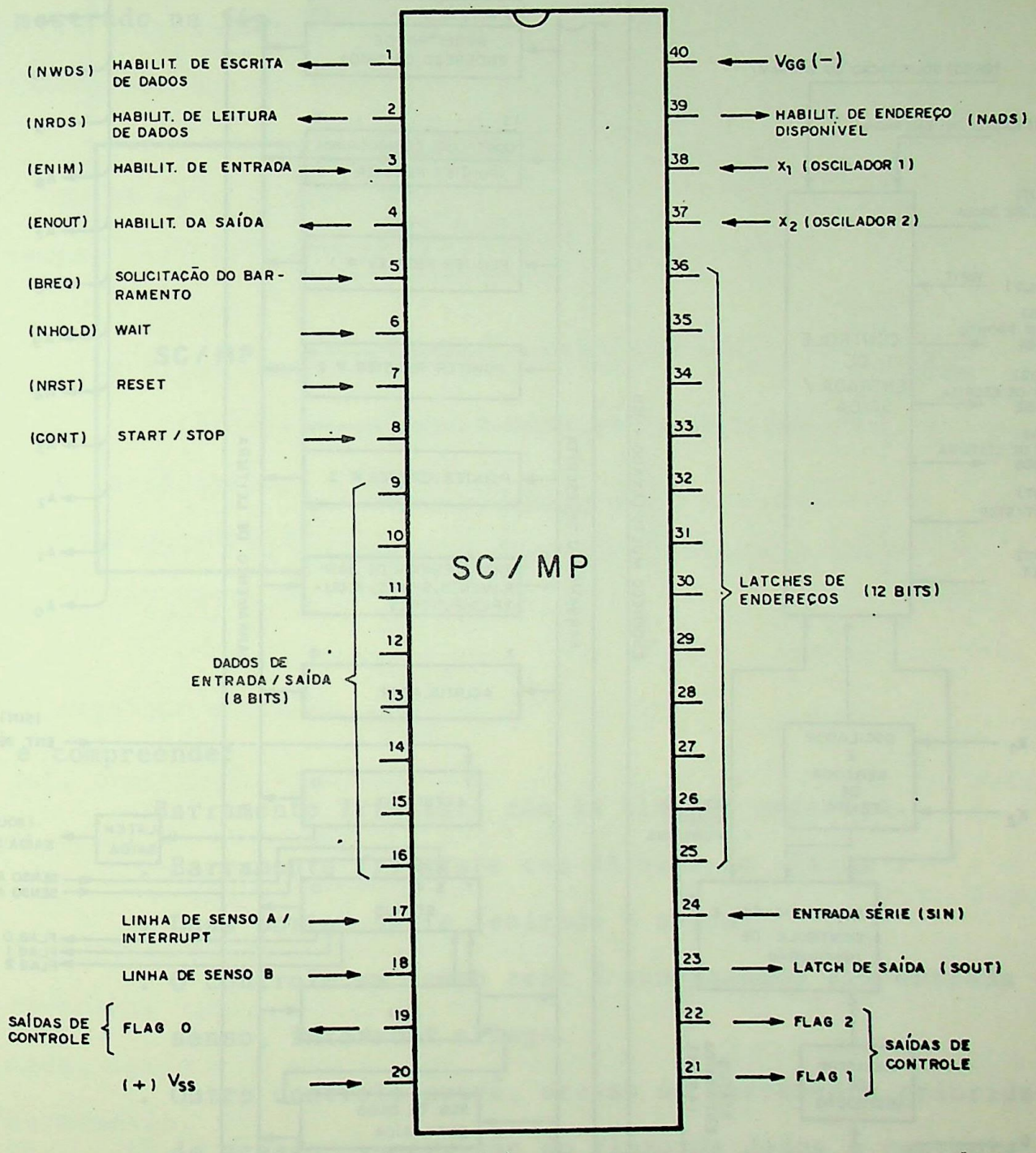


FIG. 1

O diagrama em blocos é mostrado na fig.2, isto é, a arquitetura do SC/MP.

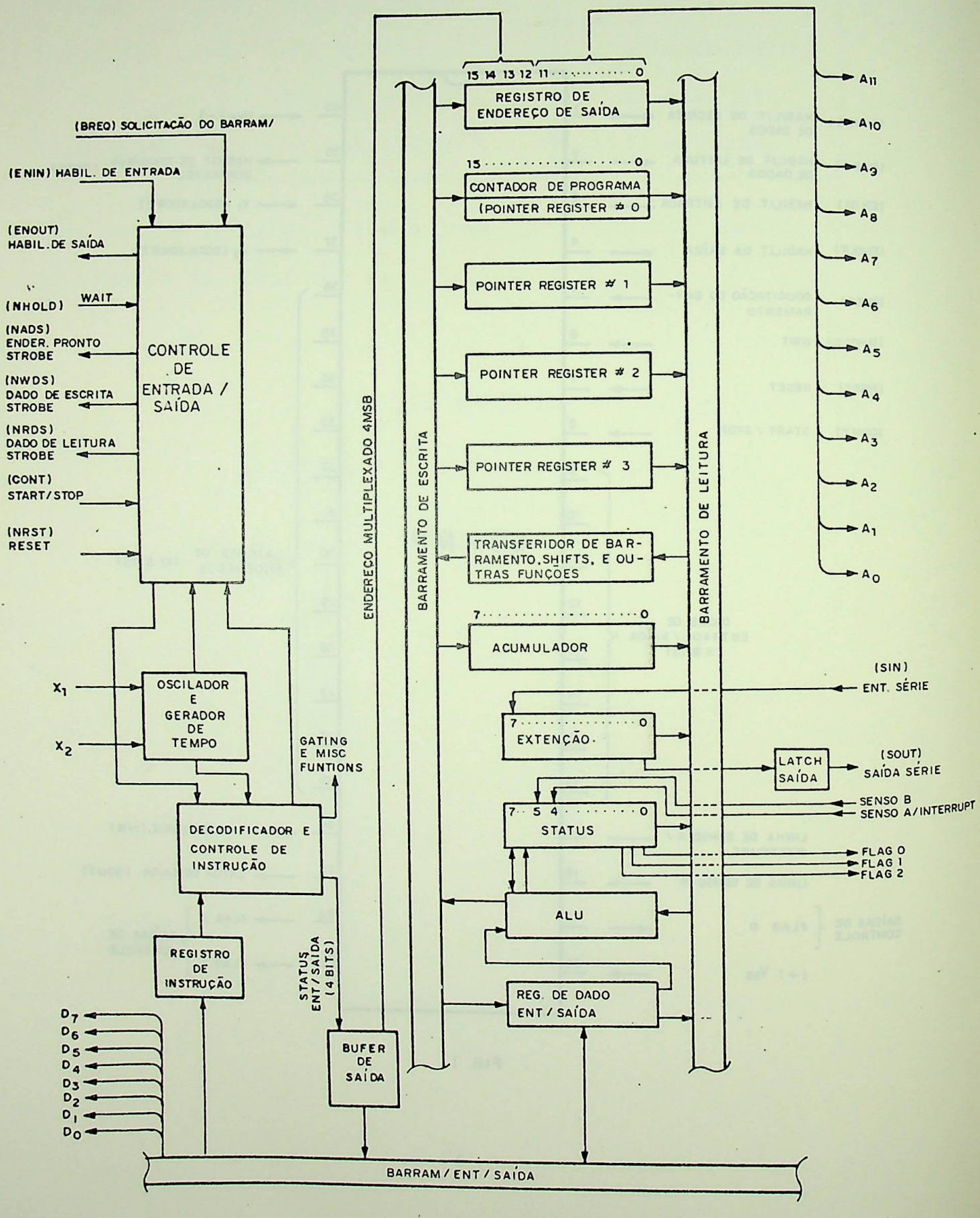


FIG. 2

A CPU do SC/MP pode ser usada em controle com uma configuração mínima, que inclui algumas chaves para controle, uma ROM implementando as instruções e uns poucos indicadores para proposição de monitoração.

A capacidade de ENTRADA/SAIDA do SC/MP é mostrado na fig. 03.

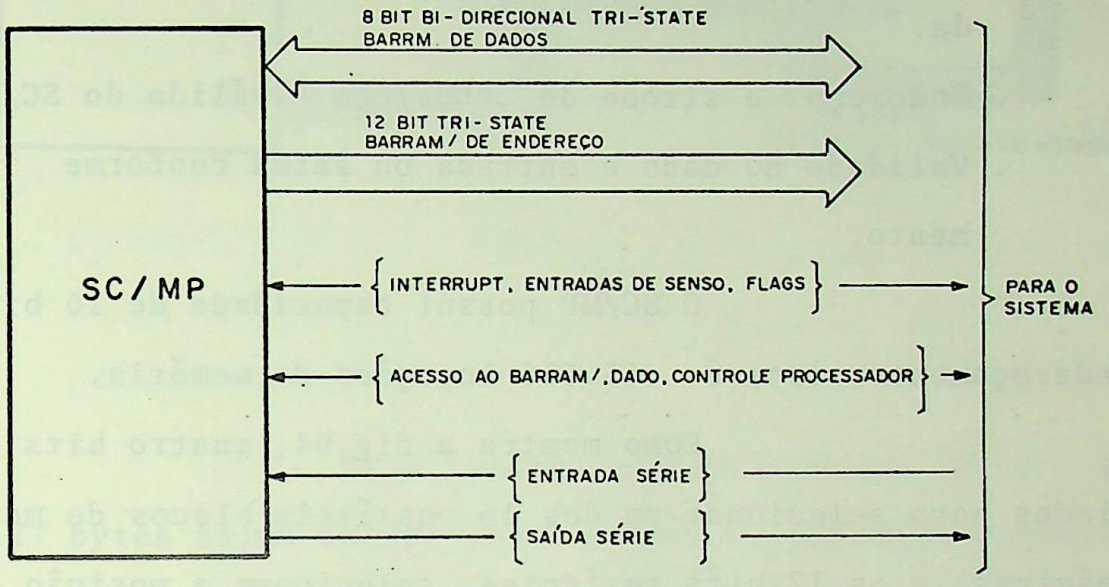


FIG. 3

e compreende:

.Barramento Tri-state com 12 bits de endereço.

. Barramento Tri-state com 08 bits de dados .

. Duas portas série (entrada e saida)

. O controle em tempo real é acompanhado via entrada de senso, interrupt eFlags.

. Outro controle provê, acesso ao barramento,prioridade de acesso, supervisão do fluxo de dados e controle geral do processador.

As linhas, acesso ao barramento e priori

dade de acesso podem ser usados como linhas de controle para fazer cascata de vários processadores.

A sequência de controle de entrada/saida do SC/MP pode ser resumida como:

. Solicitação ao barramento pelo SC/MP

. Solicitação garantida ou negada - Se negada, a solicitação ao barramento permanece ativada até ser garantida.

. Endereço e strobe de endereço - válido do SC/MP

. Validade do dado e entrada ou saida conforme requerimento.

O SC/MP possui capacidade de 16 bits de endereçamento, isto é, 65.536 locações de memória.

Como mostra a fig.04, quatro bits são enviados para selecionar um dos 16 possíveis blocos de memória (páginas) e os 12 bits restantes, selecionam a posição da memória ROM, RAM ou PROM .

O SC/MP usa cinco modos de montagem do endereço:

. Contador de Programa relativo

. Imediato

. Indexado

. Auto-indexado

. Subentendido

A seguir, mostraremos na fig. 04, os cinco modos de montagem do endereço.

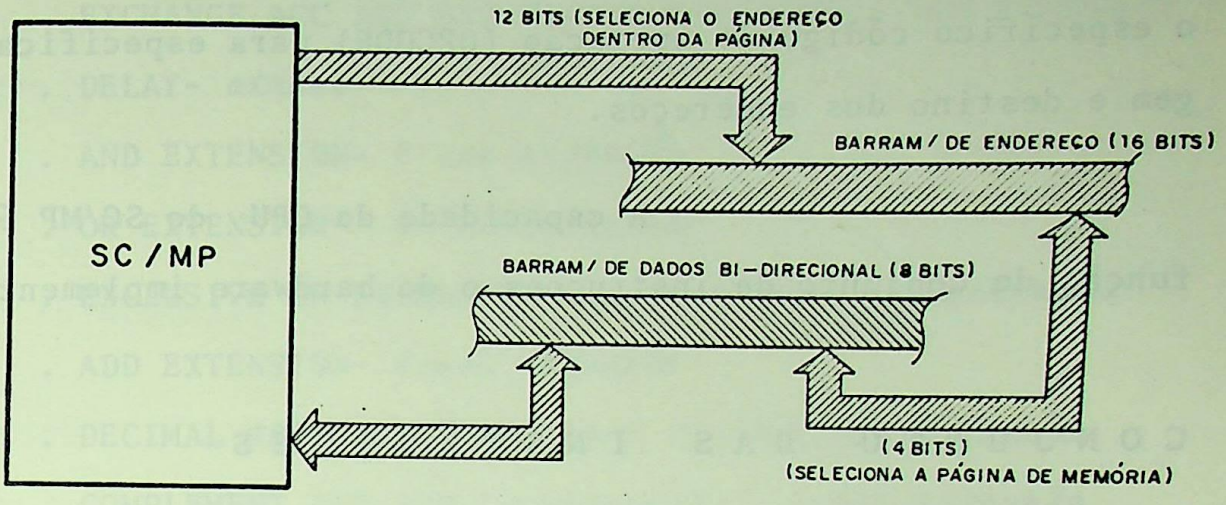


FIG. 4

Se o limite de endereçamento não é mais que 127 bytes acima ou 128 bytes abaixo do endereço especificado pelo contador de programa, o modo de endereçamento PC-relativo.

Nèste modo, um endereço efetivo é formado pela combinação dos conteúdos do contador de programa (PC), com o campo de deslocamento (segundo byte da instrução).

O modo imediato simplesmente usa o segundo byte da instrução como o dado da memória; este modo é relativamente rápido, pois um endereço de dado adicional não é formado, assim, não existe um outro ciclo da busca de informação, na memória.

Para o endereçamento indexado, o campo de deslocamento é combinado com o conteúdo do POINTER REGISTER a fim de formar o endereço efetivo.

O endereçamento auto-indexado é semelhante

ao indexado, somente o conteúdo do *Pointer Register* é trocado pelo endereço efetivo. O modo de endereçamento escolhido usa o específico código de operação (OPCODE) para especificar origem e destino dos endereços.

A capacidade da CPU do SC/MP é uma função do conjunto de instruções e do hardware implementado.

C O N J U N T O D A S I N S T R U Ç Õ E S

- LOAD - *Carregar*
- STORE- *Armazenar*
- AND - *E*
- EXCLUSIVE-OR- *ou-exclusivo*
- DECIMAL ADD - *Soma decimal*
- ADD - *Somar*
- COMPLEMENT AND ADD - *Complementar e somar*
- LOAD IMMEDIATE - *Somar imediatamente*
- AND IMMEDIATE - *E imediato*
- OR IMMEDIATE - *ou imediato*
- EXCLUSIVE OR IMMEDIATE *ou-exclusivo imediato*
- ADD IMMEDIATE- *Somar imediatamente*
- DECIMAL ADD IMMEDIATE- *Soma decimal imediata*
- COMPLEMENT AND ADD IMMEDIATE - *Complementar e somar*
- JUMP - *Salto*
- JUMP IF POSITIVE- *Saltar se positivo*
- JUMP IF ZERO - *saltar se zero*
- JUMP IF NOT ZERO - *saltar se não for zero*
- INCREMENT AND LOAD- *Incrementar e carregar*

. DECREMENT AND LOAD- *Decrementar e carregar*

. LOAD ACC FROM EXT- *Carregar acumulador com extensão*

. EXCHANGE ACC AND EXT-*Trocar acumulador com extensão*

. DELAY- *atraso*

. AND EXTENSION- *E com extensão*

. OR EXTENSION- *Ou com extensão*

. EXCLUSIVE OR EXTENSION- *Ou-exclusivo com extensão*

. ADD EXTENSION- *Somar extensão*

. DECIMAL ADD EXTENSION- *Soma decimal com extensão*

. COMPLEMENT AND ADD-*Complementar e somar extensão*

. EXCHANGE POINTER LOW-*Trocar menos significativos do Pointer*

. EXCHANGE POINTER HIGH- *Trocar mais significat. do Pointer*

. EXCHANGE POINTER WITH- *Trocar POinter com PC*

. SERIAL I/O- *Entrada /saída-série*

. SHIFT RIGHT- *Deslocamento à direita*

. SHIFT RIGHT WITH LINK-*Deslocamento à direita em anel*

. ROTATE RIGHT- *Rotação à direita*

. ROTATE RIGHT WITH LINK-*Rotação à direita em anel*

. HALT- *Parar*

. CLEAR CARRY LINK- *Limpar carry*

. SET CARRY LINK- *Setar carry*

. COPY ACC TO STATUS- *Copiar acumulador para status*

. COPY STATUS TO ACC-*Copiar status para acumulador*

. ENABLE INTERRUPTS- *Habilitar interrupções*

. DISABLE INTERRUPTS-*Desabilitar interrupções*

. NO OPERATION- *Não fazer nada*

REGISTROS E BLOCOS DE CONTROLE

OSCILADOR E GERADOR DE TEMPO

Gera a temporização básica para o controle das funções do SC/MP.

CONTROLE DE ENTRADA/SAIDA

Gera os sinais de controle da entrada e saída.

CONTROLE E DECODIFICAÇÃO DE INSTRUÇÕES:

Decodifica as instruções e providencia todo o controle e funções de gatilhamento, requeridas para execução da operação específica.

REGISTRO DE INSTRUÇÃO

Armazenar o byte um da instrução durante o ciclo de execução.

REGISTRO DE ENDEREÇO DE SAÍDA

Armazena os 16 bits de endereço no tempo de habilitação de endereço.

Os 04 MSB (bits mais significativos) são habilitados para o barramento de dados, e os 12 LSB(bits menos significativos) são colocados no barramento de endereço na saída.

REGISTRO DE DADO DE ENTRADA/SAÍDA:

Durante o ciclo de entrada, recebe informação do barramento de dados (08 bits); durante o ciclo de saída para o conteúdo do registro é transferido para o barramento de dados.

REGISTRO ACUMULADOR

Registro de 08 bits que serve como ligação entre a memória e todos os outros registros controlados pelo Software.

Usado na execução de operações aritméticas e lógicas, para memorizar os resultados destas operações, transferência de dados, deslocamentos e rotações.

REGISTRO EXTENSÃO

Registro de 08 bits que apoia o acumulador nas operações aritméticas e lógicas, transferências.

Pode também implementar operações de entrada/saída série no registro de extensão.

REGISTRO STATUS

Registro de 08 bits que faz a memorização para o Status aritmético, controle de status e software de status.

UNIDADE ARITMÉTICA E LÓGICA

Executa operações lógicas e aritméticas

CONTADOR DE PROGRAMA (REGISTRO POINTER Nº 0)

Executa o sequenciamento do programa, isto é, endereça sequencialmente as instruções do programa.

REGISTRO POINTER Nº 01

Registro de 16 bits usado para memória temporária e outras funções diferentes.

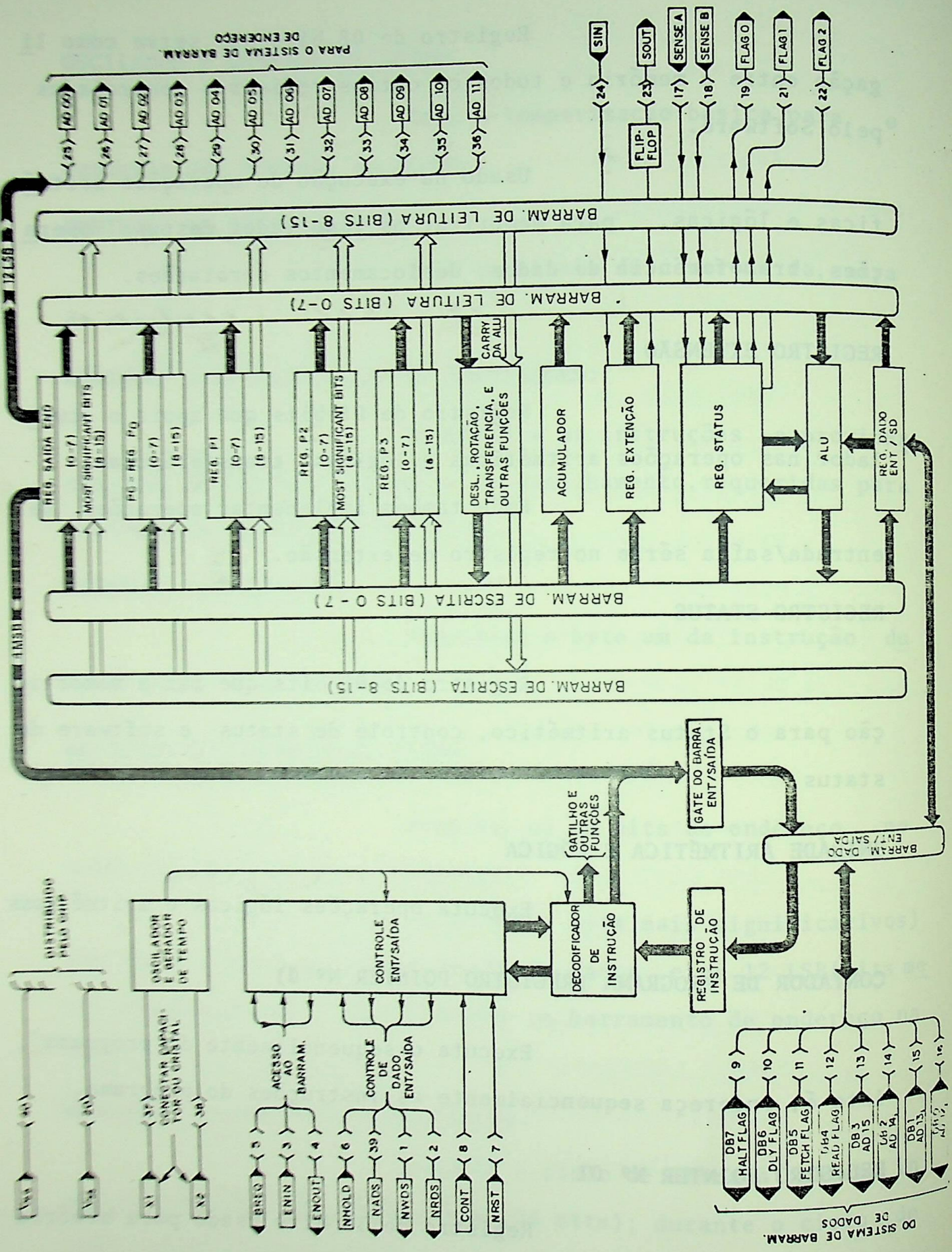

PARA O SISTEMA DE BARRAM. DE ENDEREÇO
AD 00
AD 01
AD 02
AD 03
AD 04
AD 05
AD 06
AD 07
AD 08
AD 09
AD 10
AD 11
SIN
SOUT
SENSE A
SENSE B
FLAG 0
FLAG 1
FLAG 2
FLIP-FLOP
BARRAM. DE LEITURA (BITS 8-15)
BARRAM. DE LEITURA (BITS 0-7)
REG. SAÍDA END
REG. P1
REG. P2
REG. P3
DESL., ROTAÇÃO, TRANSFERENCIA, E OUTRAS FUNÇÕES
CARRY DA ALU
ACUMULADOR
REG. EXTENÇÃO
REG. STATUS
ALU
REG. DADO ENT / SD
BARRAM. DE ESCRITA (BITS 0-7)
BARRAM. DE ESCRITA (BITS 8-15)
GATE DO BARRA ENT/SAÍDA
GATILHO E OUTRAS FUNÇÕES
BARRAM. DADO ENT / SAÍDA
DECODIFICADOR DE INSTRUÇÃO
REGISTRO DE INSTRUÇÃO
CONTROLE ENT/SAÍDA
OSCILADOR E GERADOR DE TEMPO
DISTRIBUIDO PELO CHIP
CONECTAR CAPACITOR OU CRISTAL
ACESSO AO BARRAM.
CONTROLE DE DADO ENT/SAÍDA
BREQ
ENIN
ENOUT
NHOLD
NADS
NWDS
NRDS
CONT
NRST
DB7 HALT FLAG
DB6 DLY FLAG
DB5 FETCH FLAG
DB4 READ FLAG
DB3 AD 15
DB2 AD 14
DB1 AD 13
DO SISTEMA DE BARRAM. DE DADOS

FIG. 5

REGISTRO POINTER Nº 02

Ìdem ao anterior

REGISTRO POINTER Nº 03

Ìdem ao anterior

DESLOCAMENTO,ROTAÇÃO,TRANSFERÊNCIA E OUTRAS FUNÇÕES LÓGICAS

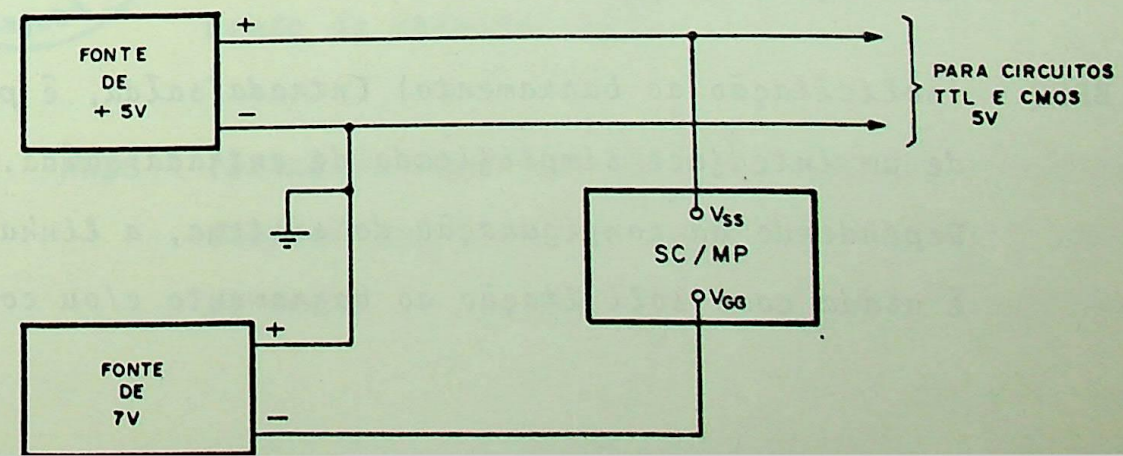
Este circuito implementa as funções indicadas e também providencia várias economias de tarefas.

NOTA :*Todos os sinais iniciados por "N" ou seguidos de (*) asterísco denotam sinal complementar,isto é, ativado com um nível lógico zero, de outro modo, o sinal é ativado com um nível lógico Um.*

ALIMENTAÇÃO E CONTROLE DE TEMPO

O Circuito integrado SC/MP pode ser alimentado por uma simples fonte de tensão contínua de -12volts com respeito a V_{ss}.

Para interface direto com TTL e CMOS um arranjo é executado como mostra a figura 6.



Toda a temporização é fornecida pelos sinais X_1 e X_2, e se deve usar um cristal, se houver necessidade de precisão nesta temporização.

DESCRIÇÃO DOS SINAIS DE ENTRADA/SAÍDA

X_1, X_2: *Conectar um capacitor ou um cristal entre os pinos X_1 e X_2 do CI para a devida temporização.*

V_{ss} - *Tensão de alimentação positiva*

V_{gg} - *Tensão de alimentação negativa*

NRST - *(Reset) Entrada, quando tem um nível baixo, interrompe a operação do processo, quando retorna ao nível alto faz a zeragem de todos os registros e a próxima instrução é buscada na posição $(0001)_{16}$ da memória.*

CONT - *(continue) Entrada, quando tem um nível alto, a instrução é buscada no endereço especificado no contador de programa e executada.*
Quando está em nível baixo, a operação do processador é suspensa anteriormente à próxima instrução a ser buscada. Consequentemente ciclos de busca e execução podem ser implementados manualmente no modo instrução, simples.

BREQ - *(solicitação ao barramento) Entrada/saída, é parte de um interface simplificado de entrada/saída.*
Dependendo da configuração do sistema, a linha BREQ, é usada como solicitação ao barramento c/ou como um

sinal de barramento ocupado (busy). A linha BREQ usada como lógica E por fios, exige um resistor de carga para V_{gg} ou terra.

ENIN - (habilitação de entrada) Entrada, quando em nível alto, é garantido o acesso do processador ao barramento de entrada/saída. Quando baixo, o acesso é inibido.

ENOUT - (habilitação de saída) Saída, quando em nível alto, indica que ENIN é alto e que a CPU não tem acesso ao barramento. Quando baixo, indica que a CPU tem acesso ao barramento ou que ENIN é baixo.

NADS - ("strobe" de endereço) Saída, quando em nível baixo, os 04 bits de status de ENTRADA/SAÍDA e os 04 mais significativos bits dos 16 bits de endereço são disponíveis no barramento de dados do sistema.

NRDS - ("strobe" de leitura) Saída, quando em nível baixo, dados são colocados do barramento de dados do sistema para dentro do processador, quando alto, o processador não faz leitura no barramento de entrada/saída.
Esta linha é uma carga de alta impedância (circuito aberto), quando o SC/MP não está acessando o barramento de entrada/saída.

Biblioteca MAUÁ Itajubá

NWDS - (strob de escrita) Saída, quando em nível baixo, são disponíveis dados do processador para o barramento de dados de entrada/saída; quando o processador não tem

acesso ao barramento de dados de entrada/saída, a linha NWDS é uma carga de alta impedância (circuito aberto).

NHOLD - (manter ou estender) Entrada; quando em nível baixo, estende o ciclo de entrada/saída até o sinal ir para alto, isto é, atrasando o edge negativo do pulso NRDS, para permitir o interface com periféricos de entrada/saída não sejam tão rápidos quanto o SC/MP.

SENSE A- (Entrada de senso e solicitação de interrupção). Esta linha é amostrada pela análise do quarto bit do registro de STATUS; esta linha é testada de modo síncrono; Serve também para linha de solicitação de interrupção caso seja solicitado pelo software.

SENSE B- (Entrada de senso); esta linha está ligada ao quinto bit do registro de status. Não tem função de interrupção.

SIN - (Entrada série para o registro E) sob o controle do software, os dados nesta linha são deslocados para dentro do registro. E de modo série, pela instrução SIO.

SOUT- (Saída série do registroE) sob o controle do software, o conteúdo do registro E é deslocado de modo série, pela linha SOUT, pela instrução SIO.
A saída é armazenada por cada intervalo entre dois comandos SIO.

FLAG 0, FLAG 1, FLAG 2 - (Saídas do Flag). Os flags 0,1 e 2, correspondem respectivamente aos bits 0,1e 2 do registro de status, estes bits são disponíveis para o usuário executar funções específicas com eles.

AD000-AD01-....AD11 - (Bits de endereço de 00 até 11); são doze linhas TRI-STATE, ao comando NADS (Strobe de Endereço), sinais de endereço são disponíveis nestas linhas. Os endereços permanecem válidos até o Edge negativo do strobe da leitura (NRDS) ou escrita (NWDS).

NOTA:"AS LINHAS DE ENDEREÇO SÃO CARGAS DE ALTA IMPEDÂNCIA (CIRCUITOABERTO) QUANDO O SC/MP NÃO TEM ACESSO AO BARRAMENTO DE ENTRADA/SAÍDA."

DB0 - (AD-12-Bit de endereço nº 12) quarto bit mais significativo dos 16 bits de endereço. Entrada quando acontece NRDS
Saída quando acontece NWDS

DB1 - (AD-13 - Bit de endereço número 13) terceiro mais significativo bit dos 16 bits de endereço. Saida em NWDS entrada em NRDS.

DB2 - (AD-14 - Bit de endereço número 14) segundo mais significativo bit dos 16 bits de endereço. Saída em NWDS, entrada em NRDS.

DB4 - (RFLG -R-FLAG); quando em nível alto, o ciclo de entrada de dados é iniciado; quando em baixo, o ciclo de saída de dados é iniciado. Saída em NWDS, entrada em NRDS.

DB5 - (IFLG-I-FLAG); quando nível alto, o primeiro byte da instrução é buscado. Saída em NWDS, entrada em NRDS.

DB6 - (DFLG-D-FLAG); quando nível alto, indica que o ciclo de atraso é iniciado, isto é, o segundo byte da instrução DLY está sendo buscado. Saída em NWDS, entrada em NRDS.

DB7 - (HFLG-H-FLAG); quando nível alto, indica que a instrução HALT está sendo executada. Em algumas configurações, a saída H-FLAG é memorizada, e em conjunção com a entrada CONTinue, prove uma parada programada; Saída em NWDS, entrada em NRDS.

NOTA: TODAS ESTAS SAÍDAS (DBO ATÉ DB7) SÃO ALTA-IMPEDÂNCIA, QUANDO O SC/MP NÃO TEM ACESSO AO BARRAMENTO DE ENTRADA/SAÍDA.

ACESSO AO BARRAMENTO

Antes do SC/MP poder transferir ou receber dados da memória ou de outro periférico, ele deve ter acesso ao sistema de endereço de controle e sistema de barramento de dados.

O acesso ao barramento é controlado por três

sinais:

. Solicitação ao barramento (BREQ)

. Habilitação de entrada (ENIN)

. Habilitação de saída (ENOUT)

Para um sistema simples, BREQ e ENOUT não precisam ser usados, e ENIN pode ser permanentemente habitada (conectar em V_{SS}).

Em sistemas onde a troca de informações deve ser rápida, é usado o método DMA (Direct Memmory Access), acesso direto à memória, isto pode ser facilmente implementado, com o SC/MP como mostra a figura 7.

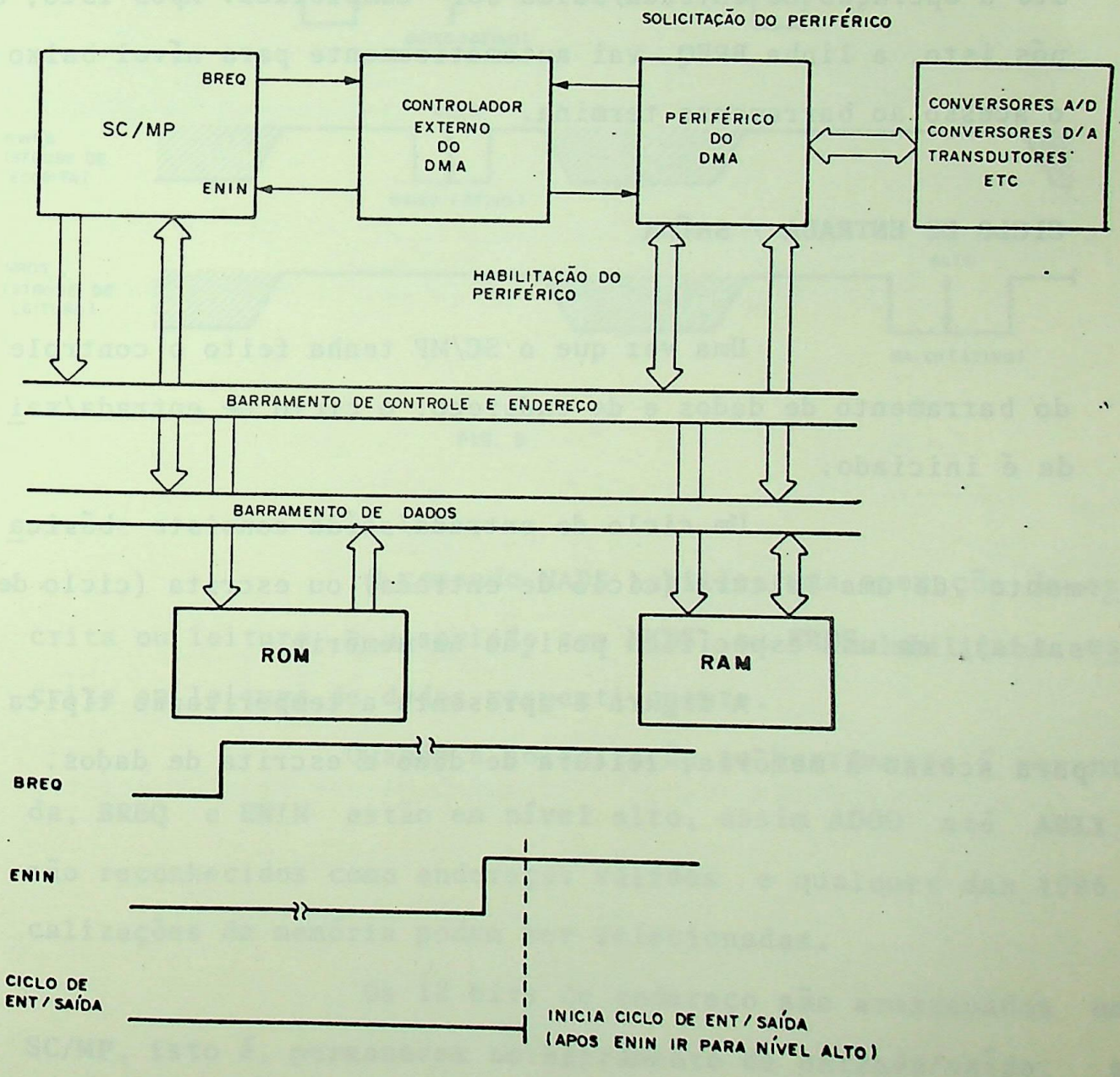


FIG. 7

Na figura 7,o acesso ao barramento é controlado por um circuito lógico externo.

Como mostra a carta de tempo, o SC/MP solicita acesso ao barramento fazendo BREQ ir para nível alto.

Este sinal atua no controlador de DMA, e se o barramento está disponível, o sinal de habilitação de entrada ENIN vai para nível alto, e o acesso ao barramento é garantido.

Com BREQ e ENIN em nível alto, o SC/MP , pode usar os barramentos de dados e endereços sem interrupções até a operação de entrada/saída ser completada. Após isto, a pós isto, a linha BREQ vai automáticamente para nível baixo e o acesso ao barramento termina.

CICLO DE ENTRADA / SAÍDA

Uma vez que o SC/MP tenha feito o controle do barramento de dados e de endereço, o ciclo de entrada/saida é iniciado.

Um ciclo de entrada/saída consiste básicamente de uma leitura (ciclo de entrada) ou escrita (ciclo de saida), em uma específica posição na memória.

A figura 8 apresenta a temporização típica para acesso à memória, leitura de dado e escrita de dados.

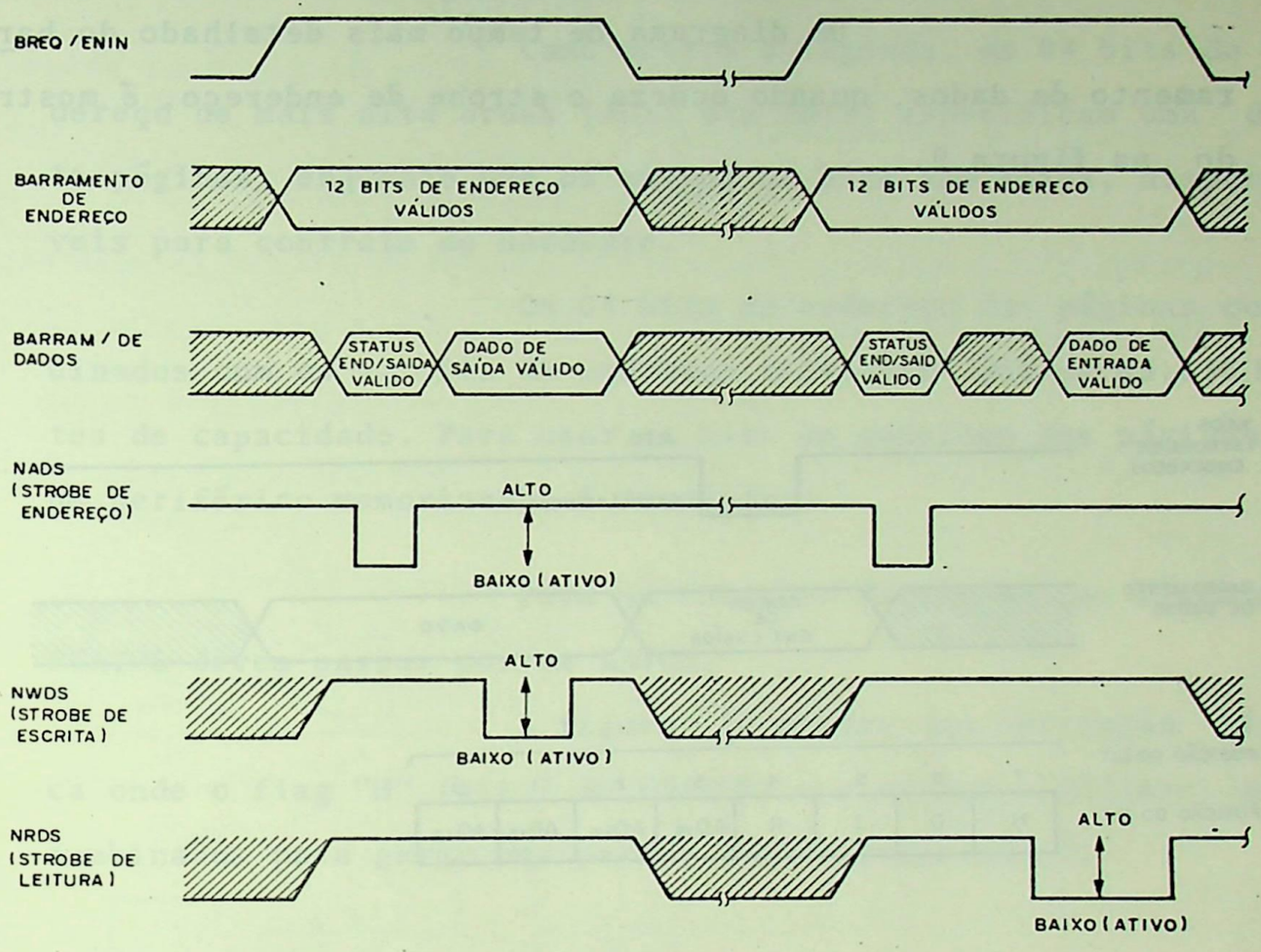


FIG. 8

O comando **NADS** habilita uma operação de escrita ou leitura e associado com **NWDS** ou **NRDS**, habilita a escrita ou leitura de dados respectivamente.

Quando a solicitação ao barramento é garantida, **BREQ** e **ENIN** estão em nível alto, assim **AD00** até **AD11** são reconhecidos como endereços válidos e qualquer das 4096 localizações de memória podem ser selecionadas.

Os 12 bits de endereço são armazenados no SC/MP, isto é, permanecem no barramento de entrada/saída; o

As formas de onda da figura 10 apresentam a mais eficiente maneira de programação.

Os vários parâmetros são auto-explicativos. Faremos uma explanação dos dois mais importantes somente; o RISE e FALL TIME do pulso de programação (T_{PR} e T_{PF}), respectivamente, que devem ser mantidos dentro do limite de 0,5 a 2μs, a fim de minimizar o efeito transitório de acoplamento. Isto, normalmente, exige uma série de circuitos RC na saída do pulso de programação para modificar o RISE-TIME.

Outro parâmetro importante é a transição do modo programação para o de leitura.

Se a transição do $\overline{CS}$/WE não ocorrer após o final da transição do pulso de programação e antes da transição do endereço, nós internos não estarão descarregados, fazendo a saída indicar dados falsos por vários milisegundos, e mostrado na figura 10 com o nome T_{DPR} (10 μs).

Se a temporização de $\overline{CS}$/WE é difícil de ajustar, providenciando o complemento binário do primeiro endereço para ser verificado, faz com que seja descarregado os nós internos.

PROGRAMAÇÃO

Vários tipos de programadores da memória ROM 2708 são disponíveis no mercado internacional.

Porém, devido as dificuldades de importação, faremos a opção de projetar um. Isto é possível com o uso de um microprocessador, e seguindo a descrição:

- O dispositivo é colocado na operação de

programação, pela colocação da entrada $\overline{CS}/WE$ (pino 20) em V_{IHW} (+12 V). A palavra de endereço é então selecionada do mesmo modo que no modo de leitura.

O dado a ser programado é apresentado na forma de 8 bits, em paralelo, aos pinos de dado de saída (O_1-O_8).

Os níveis lógicos para endereços, dados, e tensões de alimentação são os mesmos que no modo de leituras.

Após os tempos de SET UP de endereço e dado (T_{AS} e T_{DS} Figura 10), um pulso de programação com largura T_{PW} é aplicado ao pino de programação (pino 18). Esta sequência é então repetida para os próximos endereços.

Uma passada através dos 1024 endereços é definida como um LOOP DE PROGRAMAÇÃO. O número de LOOPS de Programação (N) requerido é função da largura do pulso de programação (T_{PW}), de acordo com a fórmula:

$$N \times t_{pw} \geq 100 \text{ ms}$$

onde:

N - número de LOOPS de Programação

t_{pw} - largura do pulso de programação.

A largura do pulso de programação pode variar de 0,1 até 1,0 ms. O número de LOOPS (N) pode variar de no mínimo 100 (t_{pw} = 1,0 ms) e no máximo 1000 (t_{pw} = 0,1 ms) dependendo do valor escolhido para t_{pw}.

Não pode ser aplicado N pulso de programação em um endereço, e então, mudar para o próximo endereço e aplicar de novo N pulsos de programação, Deve ser executado N LOOPS sucessivos através dos 1024 endereços.

Quando a chave START vai para zero(0), o circuito anti-bouce gera um pulso positivo de clock, que seta a entrada CONT na saída Q do FLIP-FLOP.

Estando DB 7 (Flag H) no nível baixo, o processador corre normalmente.

Quando o flag H vai para nível alto, a entrada CLR é ativada com o aparecimento do strobe de endereço NADS, assim a entrada CONT é feita baixa e o processador pára.

Uma outra forma de controlar o SC/MP é implementada como mostra a figura 11. Operação com ciclo- simples ou Instrução-Simples.

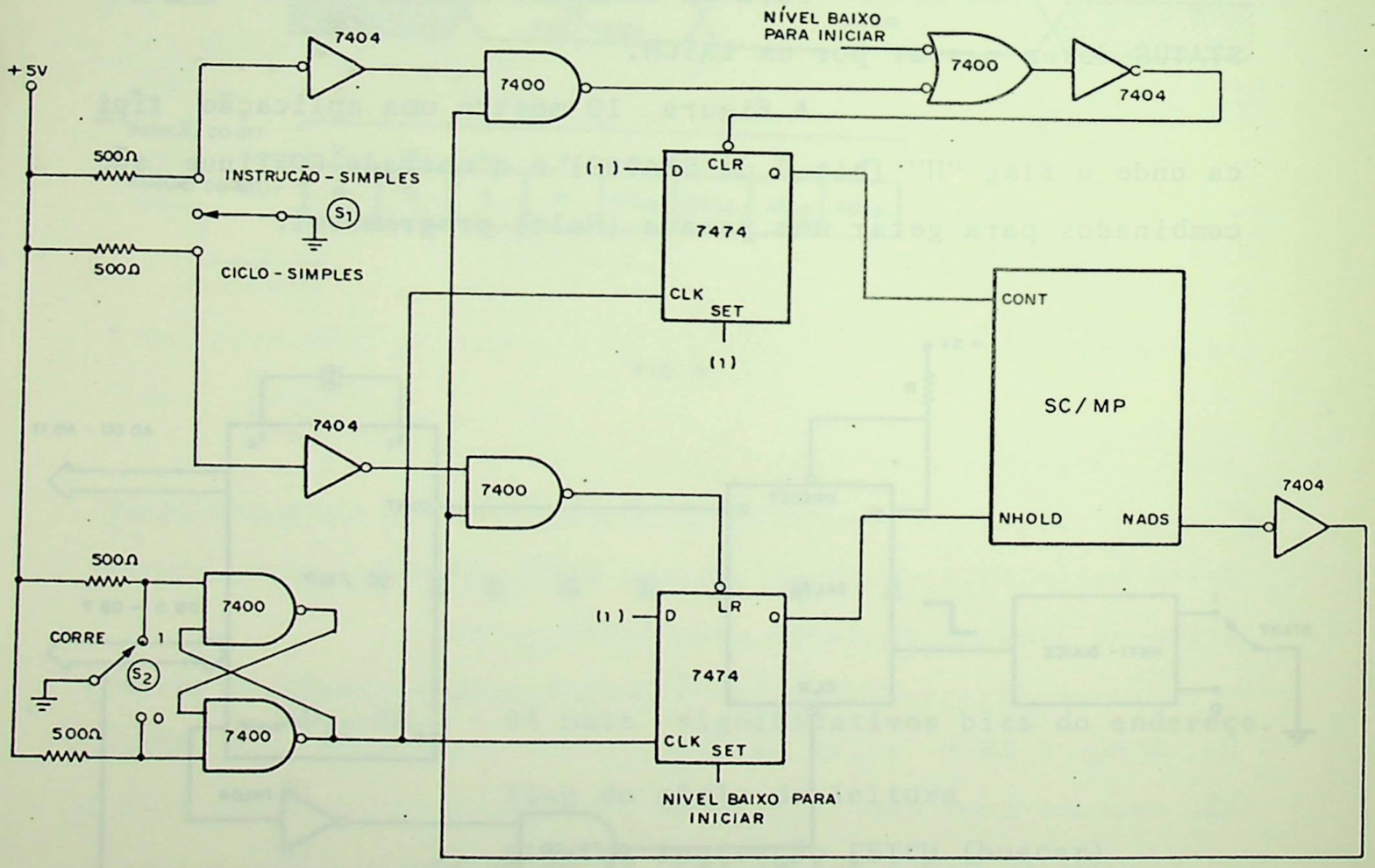


FIG. 11

A chave S_1 é colocada no modo escolhido e a chave S_2 é momentaneamente levada a zero.

Estes eventos causam um nível alto na entrada CONT e a entrada NHOLD vai para baixo; então a instrução é buscada e executada se S_1 está em instrução-simples, ou uma operação busque e espere, ocorre se S_1 está no modo Ciclo-simples.

Na ocorrência do strobe de endereço, NADS, os Flip-Flops, são zerados para iniciar nova operação.

Ao concluir o strobe de endereços, o processador está apto a iniciar um ciclo de escrita ou leitura, que são sincronizados pelos strobes de leitura (NRDS) e de escrita (NWDS).

As transferências de dados entre o SC/MP e dispositivos periféricos não necessitam maiores complicações que os mostrados na figura 12 e 13.

A sequência de entrada/saída se processa como segue:

- O barramento solicita ao SC/MP.
- A solicitação é garantida por lógica interna ou externa.
- Endereços válidos, (e guardados se a memória exceder 4K).
- Dados válidos, introduzidos ou retirados, conforme seja o caso.

O dado válido aparece no barramento antes do strobe de leitura executar uma transição de baixo para

alto. Com o uso de novas memórias rápidas, o dado válido está sempre disponível antes do strobe de leitura ser inativo. Assim, não há necessidade de estender o intervalo de atuação do strobe de leitura. Comparar as figuras 12 e 13.

A sequência de entrada/saída é idêntica, porém,o tempo de acesso à memória é um pouco mais longo, e o strobe de leitura pode terminar seu tempo antes dos dados estarem disponíveis.

Para prevenir isto, um sinal da memória NHOLD é gerado no periférico da memória lenta e aplicado no controle do SC/MP.

Como mostra a carta de tempo, o strobe de leitura ou escrita é estendido pelo NHOLD para satisfazer as características da memória.

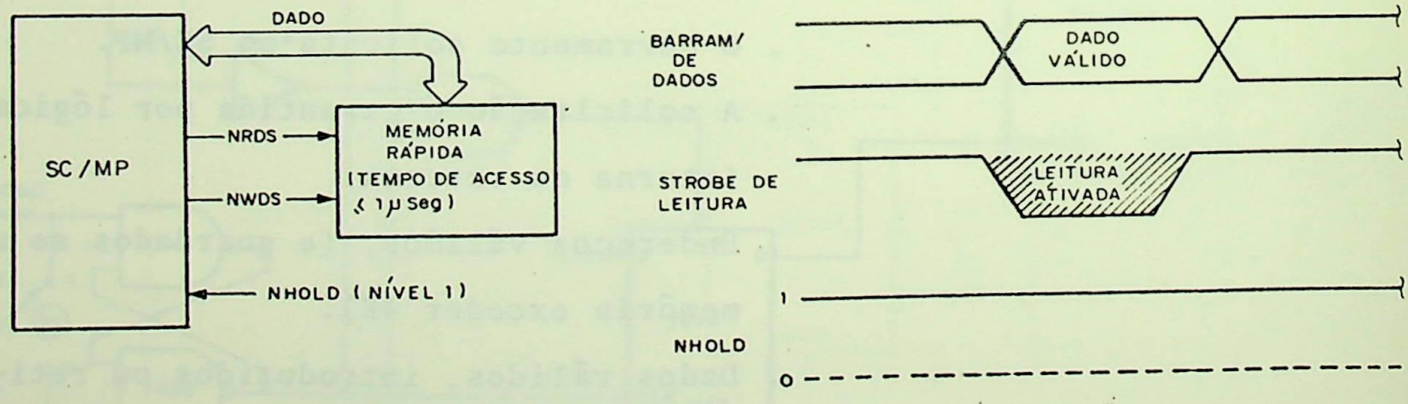


FIG. 12

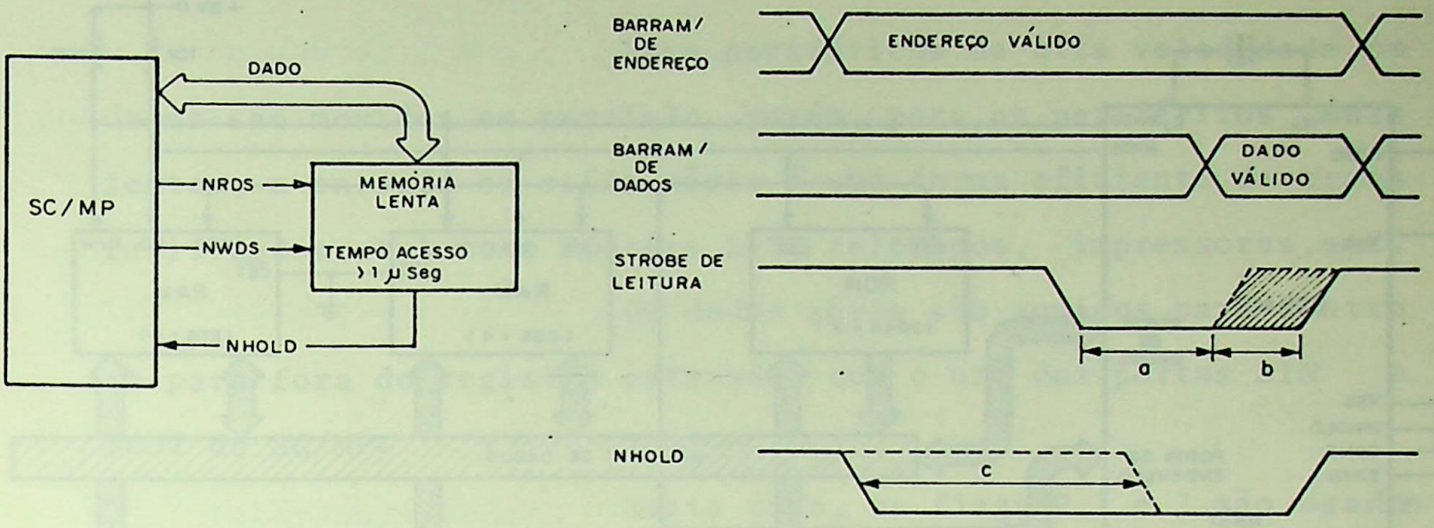


FIG. 13

Da figura 13, temos que:

a - Tempo de **NRDS** normal (tempo de acesso $\leq$1μseg)

b - Tempo de **NRDS** - estendido pelo **NHOLD** (tempo de acesso > 1 μseg)

c - Como mostra a linha tracejada, o sinal **NHOLD** pode ser ativo depois do Edge negativo do strobe de leitura.

COLOCANDO BUFFER NOS BARRAMENTOS:

Em pequenos sistemas com um mínimo de memória, buffers nos barramentos de endereços e de dados não são necessários como o da figura 14.

Com o aumento dos periféricos e requerimentos de memória, os buffers podem ser usados como mostra a figura 15.

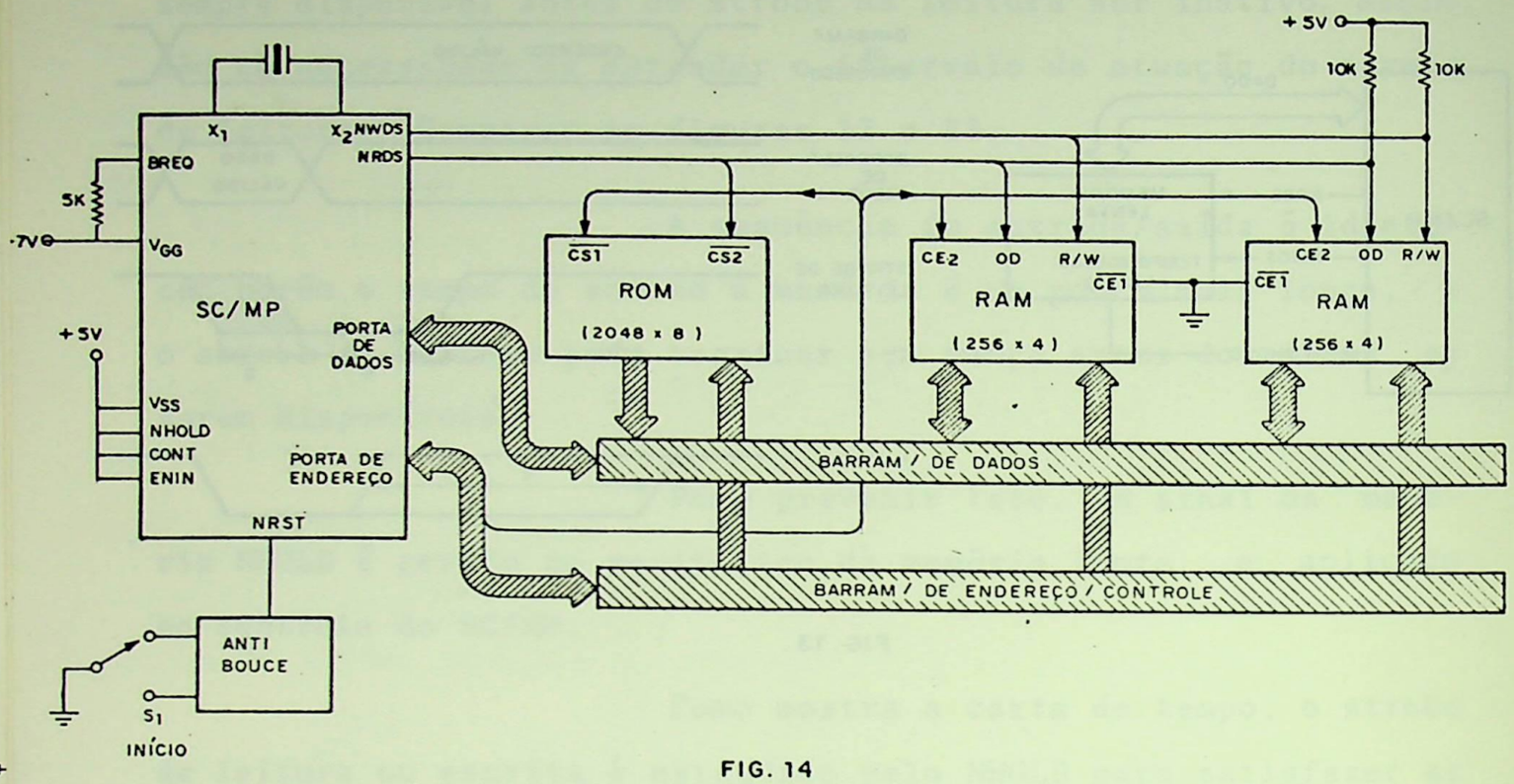

+5V
10K
10K
X1
X2
NWDS
NRDS
BREQ
5K
-7V
VGG
SC/MP
+5V
PORTA DE DADOS
VSS
NHOLD
CONT
ENIN
PORTA DE ENDEREÇO
NRST
CS1
CS2
ROM
(2048 x 8)
CE2
OD
R/W
CE1
RAM
(256 x 4)
CE1
CE2
OD
R/W
RAM
(256 x 4)
BARRAM / DE DADOS
BARRAM / DE ENDEREÇO / CONTROLE
ANTI BOUCE
S1
INÍCIO

FIG. 14

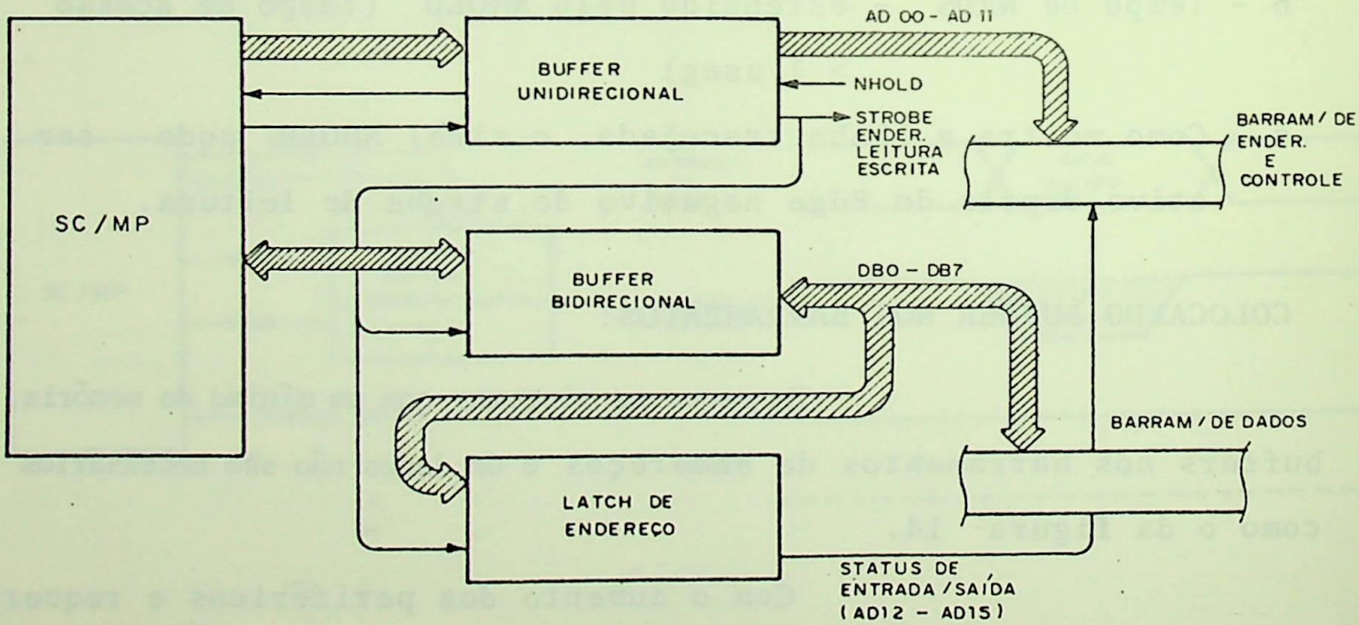

SC/MP
BUFFER UNIDIRECIONAL
AD 00 - AD 11
NHOLD
STROBE ENDER. LEITURA ESCRITA
BARRAM / DE ENDER. E CONTROLE
BUFFER BIDIRECIONAL
DB0 - DB7
BARRAM / DE DADOS
LATCH DE ENDEREÇO
STATUS DE ENTRADA / SAÍDA (AD12 - AD15)

FIG. 15

TRANSFERÊNCIA SÉRIE DE DADOS (ENTRADA/SAIDA)

Para periféricos de alta velocidade, os dados são movidos em paralelo porém, para os periféricos mais lentos, a entrada ou saída série é uma forma eficiente de transferir dados; tal como Ploters X-Y, teletipos, impressoras, etc.

Os dados série são movidos para dentro ou para fora do registro extensão, com o uso das portas SIN e SOUT do SC/MP.

Neste caso, os flags 0,1 e 2 são usados para o controle dos registradores de deslocamento.

Em outras aplicações, os flags e as duas entradas de senso podem ser usados para introduzir e retirar os dados série.

Na figura 16, as linhas SIN e SOUT podem ser expandidas para servir oito ou mais dispositivos de entrada/saída com o uso apropriado de multiplexadores.

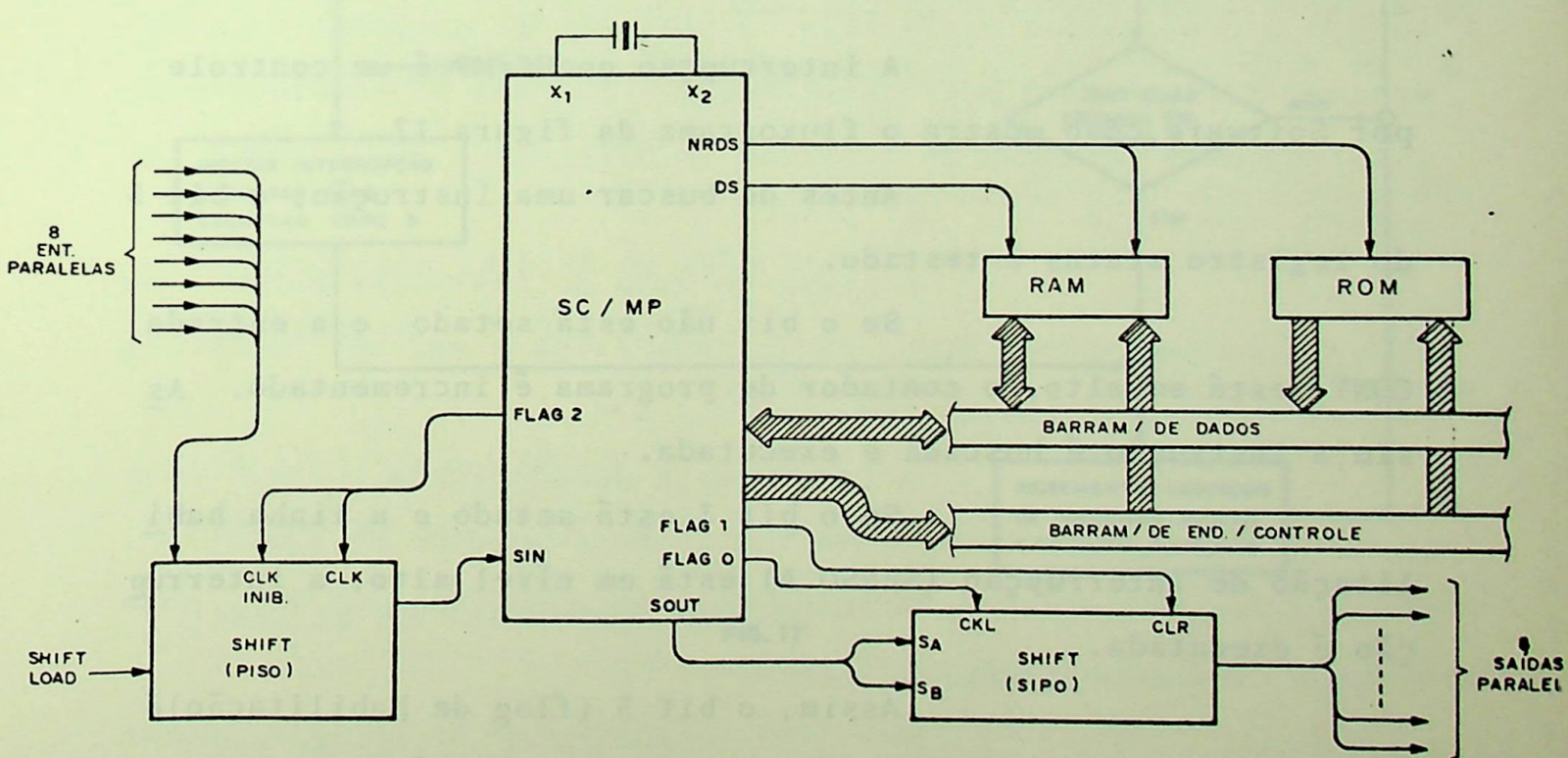


FIG. 16

FLAG E SENSO

Os bits 0,1, 2,4 e 5 do registro de status providenciam três saídas de flag e duas entradas de senso, arranjadas como seguem:

BIT O REGISTRO DE STATUS	DESCRIÇÃO
0	FLAG 0 (saida)
1	FLAG 1 (saida)
2	FLAG 2 (saida)
4	SENSO A(entrada)
5	SENSO B(entrada)

Cada uma destas funções podem ser monitoradas durante o programa, usadas para controle de Hardware, usadas como status no software ou numa combinação para controle de software /hardware.

INTERRUPÇÃO

A interrupçao no SC/MP é um controle por Software,como mostra o fluxograma da figura 17.

Antes de buscar uma instrução, o bit 3 do registro status é testado.

Se o bit não está setado e a entrada CONT está em alto, o contador de programa é incrementado. Assim,a instrução é buscada e executada.

Se o bit 3 está setado e a linha habilitação de interrupção (SENSO A) está em nível alto, a interrupção é executada.

Assim, o bit 3 (flag de habilitação)é é resetado e o conteúdo do contador de programa é trocado com o

conteúdo do registro POINTER 3, que contém o endereço da subrotina que serve ao comando de interrupção.

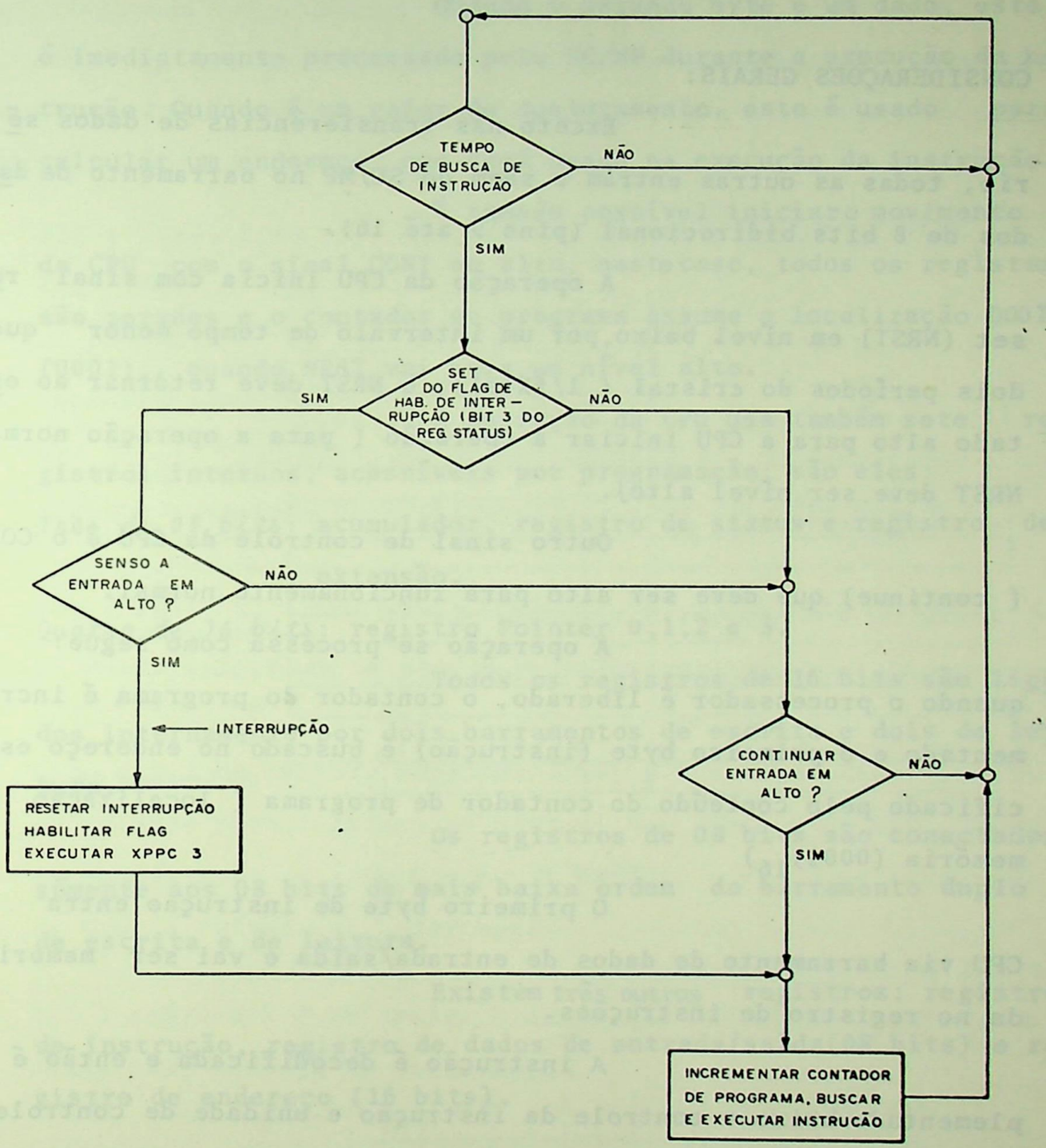


FIG. 17

MOVIMENTO DE DADOS E CONTROLE INTERNO

CONSIDERAÇOES GERAIS:

Exceto nas transferências de dados série, todas as outras entram e saem do SC/MP no barramento de dados de 8 bits bidirecional (pino 9 até 16).

A operação da CPU inicia com sinal reset (NRST) em nível baixo, por um intervalo de tempo menor que dois períodos do cristal (1/fXtal); o NRST deve retornar ao estado alto para a CPU iniciar a operação (para a operação normal NRST deve ser nível alto).

Outro sinal de controle da CPU é o CONT (continue) que deve ser alto para funcionamento normal.

A operação se processa como segue: quando o processador é liberado, o contador do programa é incrementado e o primeiro byte (instrução) é buscado no endereço especificado pelo conteúdo do contador de programa (localização de memória $(0001)_{16}$)

O primeiro byte de instruçao entra na CPU via barramento de dados de entrada/saída e vai ser memorizada no registro de instruções.

A instrução é decodificada e então é implementada baixo o controle da instrução e unidade de controle . Uma instrução simples especifica uma operação que o SC/MP pode executar sem outras referências à memória.

Uma instrução simples de um byte possui um "zero" na posição sete (bit mais significativo), enquanto um

"um" nesta posição indica instrução de dois bytes.

Uma instrução de dois byte possui além da informação operacional mais 8 bits de dados ou campo de deslocamento.

Quando o segundo byte é um dado, este é imediatamente processado pelo SC/MP durante a execução da instrução. Quando é um valor de deslocamento, este é usado para calcular um endereço que será usado na execução da instrução.

É também possível iniciaro movimento da CPU com o sinal CONT em alto; nestecaso, todos os registros são zerados e o contador de programa assume a localização (0001) $(0001)_{16}$, quando NRST vai para um nível alto.

A operação da CPU usa também sete registros internos, acessíveis por programação; são eles:

Três de 08 bits: acumulador, registro de status e registro de extensão.

Quatro de 16 bits: registro Pointer 0,1,2 e 3.

Todos os registros de 16 bits são ligados internamente por dois barramentos de escrita e dois de leitura.

Os registros de 08 bits são conectados sómente aos 08 bits de mais baixa ordem do barramento duplo de escrita e de leitura.

Existem três outros registros: registro de instrução, registro de dados de entrada/saída(08 bits) e registro de endereço (16 bits).

ANÁLISE DOS REGISTROS DO SC/MP

A figura 18 mostra o formato e extensão dos registros do SC/MP.

ACUMULADOR (AC)

O acumulador (AC) é o registro primário de trabalhos do SC/MP.

O acumulador é usado nas operações aritméticas e lógicas, guardando o resultado destas operações .

Transferências de dados, deslocamentos, rotações, também usam o acumulador .Das 46 instruções do SC/MP, 37 usam o acumulador.

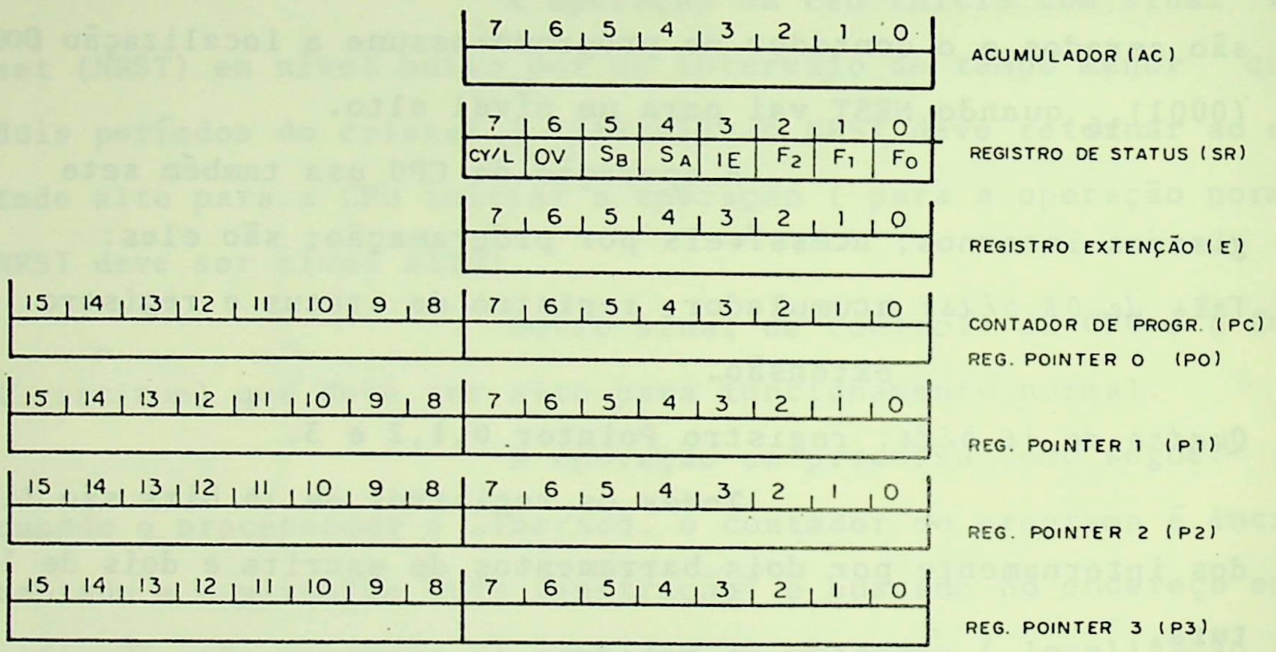


FIG.18

REGISTRO DE STATUS (SR)

7	6	5	4	3	2	1	0	POSIÇÃO DO BIT
CY/L	OV	SB	SA	IE	F2	F1	F0	FLAGS

O registro de status mantém armazena

mento para flags de status aritméticos, de controle e de software.

A função de cada bit do registro é:

BIT 0 (F0) : FLAG 0 - Usado para função de controle, para flag de status de software. A saída deste bit é disponível em um pino do SC/MP (pino 19).

BIT 1 (F1): FLAG 1 - O mesmo que o flag 0 disponível no pino 21.

BIT 2(F2): FLAG 2 - Idem ao anterior (pino 22)

BIT 3(IE): FLAG HABILITADOR DE INTERRUPÇÃO: O processador reconhece a entrada de interrupção se este flag for setado.

BIT 4 (SA): BIT A DE SENSO- Este bit é vinculado a um pino do circuito integrado, e pode ser usado como condição externa de senso.
Este bit é de leitura sómente; logo, a instrução CAS (copiar acumulador para registro de status) não afeta este bit. Quando habilitação de interrupção é setada, o bit A de senso serve como a entrada de interrupção.

BIT 5 (SB): BIT B DE SENSO: O mesmo que SA, exceto que não é usado como entrada de interrupção.

BIT 6 (OV): OVERFLOW(EXCESSO)- Este bit é setado se ocorre o Overflow na operação aritmética de soma (ASS,ADI ou ADE), ou uma instrução complementar e somar (CAI ou CAE). O Overflow

nao é afetado pelas instruções decimais de soma (DAD,DAI ou DAE).

BIT 7 (CY/L): CARRY/LINK - Este bit é setado se um carry (vai um) ocorre no bit mais significativo durante uma soma, uma instrução complementar e somar, ou uma instrução de soma decimal. Este bit também é incluido nas instruções *SHIFT RIGHT*(deslocamento à direita)*WITH LINK (SRL)* e *ROTATE RIGHT WITH LINK* (rotação à direita) *(RRL)*.

CY/L é uma entrada na posição do bit zero nas instruções de soma, complementar e somar e soma decimal.

REGISTRO EXTENSÃO: (E)

O registro de8 bits chamado *EXTENSÃO(E)* é usado com o acumulador para executar operações aritméticas , lógicas e de transferência de dados.

Se o espaçamento nas instruções indexados ou auto indexados referidas à memória se igualam a $(128)_{10}$ então o conteúdo do registro E é substituído pelo espaçamento para a dada instrução.

Outra função do registro E é o de entrada/saída série; a função entrada/saída série é uma operação simultânea (isto é, o bit de mais alta ordem é introduzido e o de mais baixa ordem é retirado).

CONTADOR DE PROGRAMA: (PC)

O contador de programa é um registro de 16 bits, coincidente com o registro Pointer 0.

No contador de programa, está o endereço da instrução que está sendo executada; ele é incrementado jus

tamente antes da instrução de busca.

Operações aritméticas afetam os primeiros 12 bits do contador de programa, os 04 últimos não são afetados.

Então, o conteúdo do PC pode movimentar até 4.096 bytes de endereço atuável.

REGISTRO POINTER: (PTR)

Este registro é de 16 bits, disponíveis para endereçamento de memória ou periféricos e uso com apontador de página, "STACK POINTER" ou registro de Index. Como mencionado antes, PO está ligado ao PC por Hardware.

FLUXO DE DADOS PELOS REGISTROS INTERNOS

As relações dos fluxos de dados pelos registros internos do SC/MP são mostrados na figura 19.

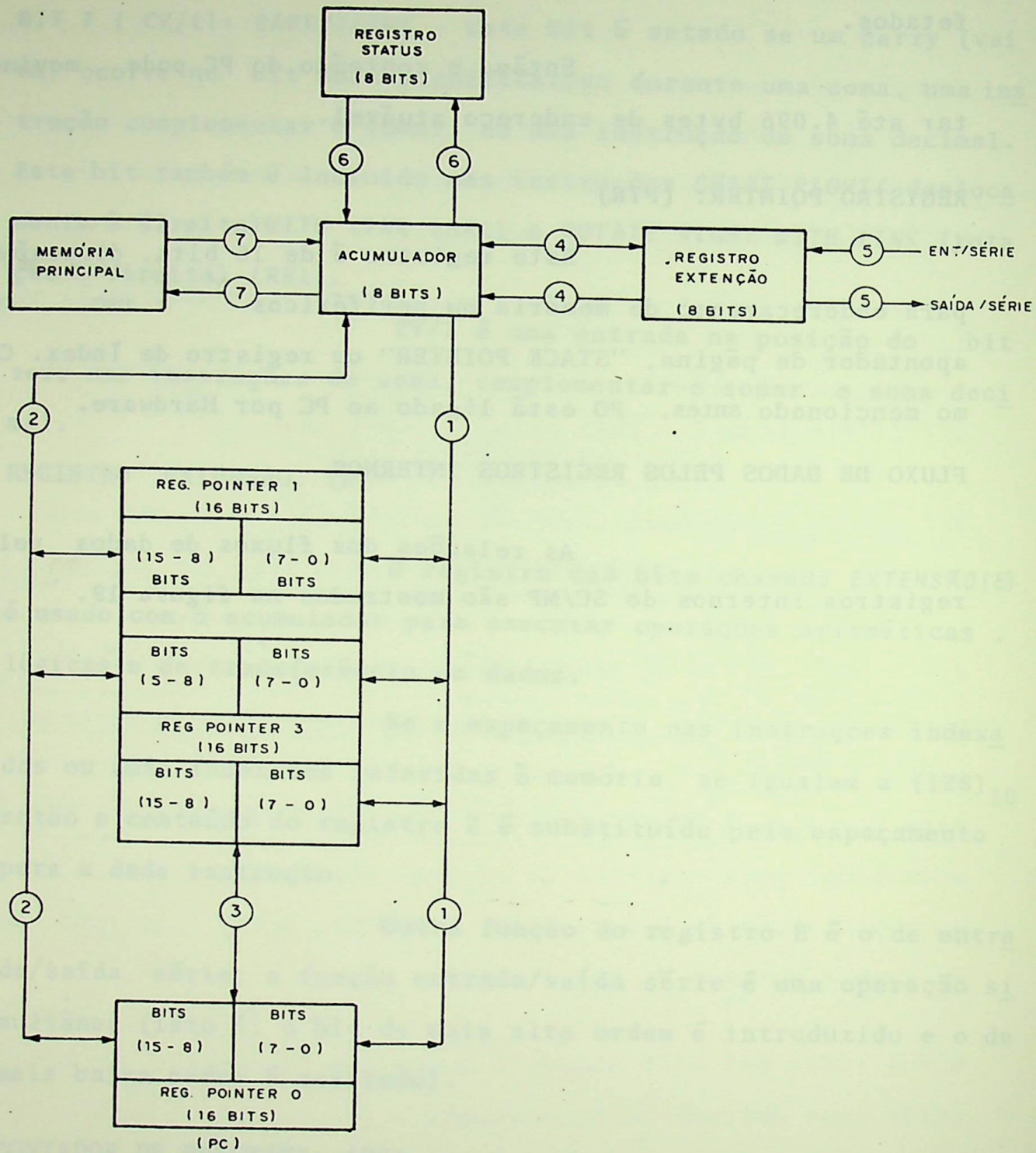

REGISTRO STATUS
(8 BITS)
6
6
MEMÓRIA PRINCIPAL
7
7
ACUMULADOR
(8 BITS)
4
4
REGISTRO EXTENÇÃO
(8 BITS)
5
ENT/SÉRIE
5
SAÍDA/SÉRIE
2
1
REG. POINTER 1
(16 BITS)
(15 - 8) BITS
(7 - 0) BITS
BITS (5 - 8)
BITS (7 - 0)
REG POINTER 3
(16 BITS)
BITS (15 - 8)
BITS (7 - 0)
2
3
1
BITS (15 - 8)
BITS (7 - 0)
REG. POINTER 0
(16 BITS)
(PC)

FIG. 19

Da figura 19, tiramos:

1. O byte de mais baixa ordem (bits 0-7) de qualquer registro pointer pode ser trocado com os 08 bits do acumulador.

2. O byte de mais alta ordem (bits 8-15) de qualquer registro Pointer pode ser trocado com os 08 bits do acumulador.

3. O conteúdo do registro Pointer 0 (bits 0-15), isto é, do contador de programa, pode ser trocado diretamente (sem auxílio do acumulador) com qualquer dos três registros Pointer.

4. O conteúdo do acumulador pode ser trocado com o conteúdo do registro extensão, ou o acumulador pode ser carregado pelo registro extensão.

5. Os dados podem ser introduzidos ou retirados de forma série do registro extensão (um bit de cada vez). Lembrar que a saída é memorizada a fim de manter a informação entre comandos de deslocamento.

6. O conteúdo do acumulador é copiado pelo registro status ou vice-versa. O conteúdo do registro original é perdido na operação. Quando o conteúdo do acumulador é copiado pelo registro status, os bits 3 e 4 deste último não são afetados.

7. Todo dado referente à memória entra ou deixa o processador via acumulador.

Escola Federal de Engenharia
Biblioteca
MAUÁ
Itajubá

ENDEREÇAMENTO NO SC/MP

CAPACIDADE GERAL

Um dos registros Pointer de 16 bits é usado em cada instrução de referência à memória.

O contador de programa (registro Pointer 0) é sempre usado para especificar o endereço das instruções do programa, enquanto os registros 1,2 e 3 são usados para outros requerimentos de endereço ou dados.

A figura 20 mostra o formato dos registros de endereçamento.

Cada registro Pointer pode identificar qualquer das 65536 posições de memória (de 0000_{16} até $ffff_{16}$).

Devido ao problema da pinagem no CI, o registro Pointer é subdividido. Os quatro bits mais significativos indicam a página e os 12 menos significativos, a localização na página. O endereço de página (04 bits) pode ser mudado via comando de programa.

Os endereços da localização (12 bits) pode ser alterado aritméticamente. Como mostra a parte c da figura 20, o campo de espaçamento de 08 bits na instrução executa uma seleção na página de memória.

Este espaçamento (08 bits) quando algébricamente combinado com o conteúdo de um dos registros POINTER produz uma variação de 256 posições (-128 até +127) na página selecionada.

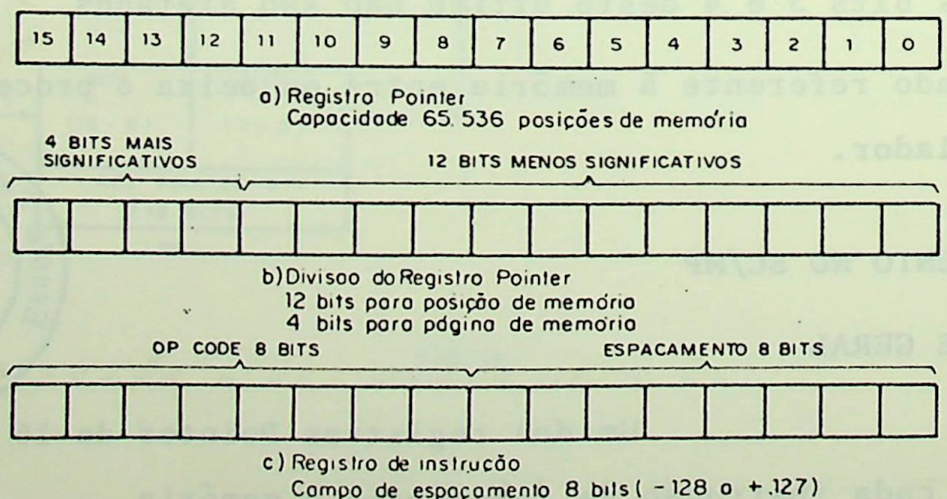


FIG. 20

ARQUITETURA DE ENDEREÇAMENTO

A memória é organizada como uma sequência de bytes de 08 bits; cada um é identificado por um endereço de 16 bits, que apresentam a posição sequencial de 0 até 65.536. Como mostra a figura 21, a memória é dividida em 16 páginas de 4096 bytes cada.

Assim, um endereço deve ter 04 bits indicando a página e 12 bits indicando a posição.

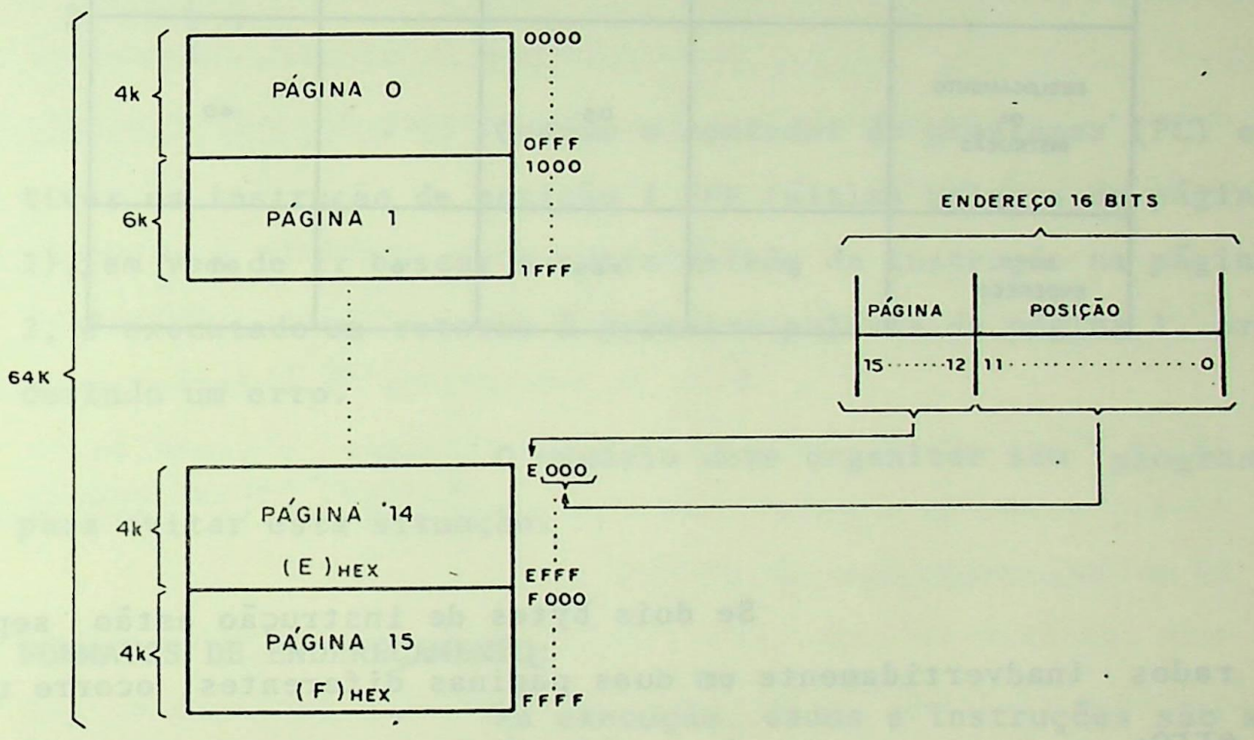


FIG. 21

Quando a aritmética é usada para o cálculo do endereço efetivo do operando, os cálculos são executados pela porção de mais baixa ordem do endereço (espaçamento), não existindo carry na posição de mais alta ordem (página); quando o deslocamento de endereço permanece na mesma página, nenhum carry é gerado.

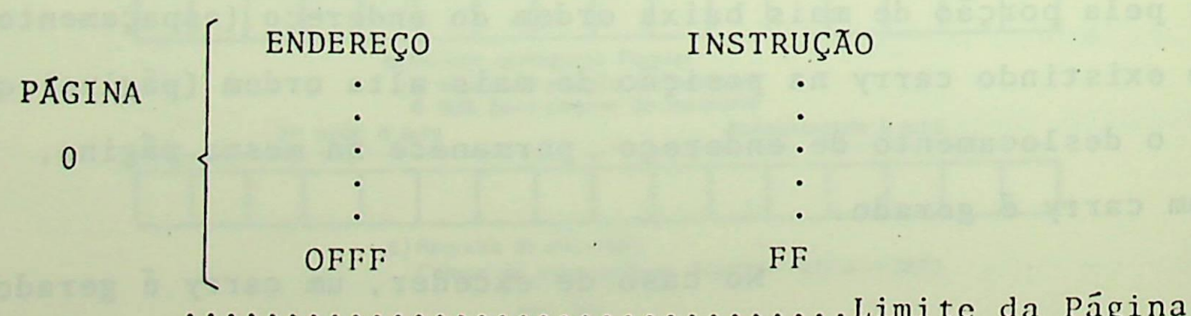
No caso de exceder, um carry é gerado ,

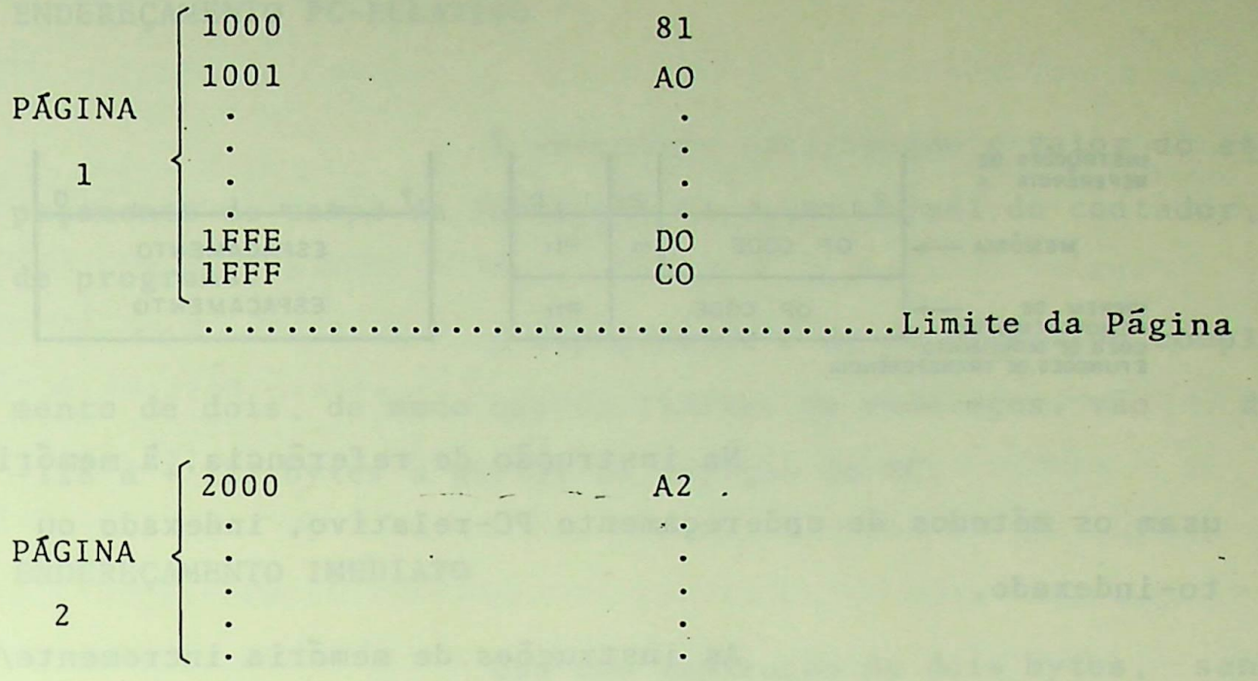
quando dois números são somados; porém, não é introduzido no campo de endereço da página.

ESPAÇAMENTO DE ENDEREÇO PERMANECE DENTRO DA PÁGINA			ESPAÇAMENTO DE ENDEREÇO EXCEDE A PÁGINA	
ENDEREÇO CORRENTE	ENDEREÇO DE PÁGINA 0	DESLOCAMENTO DENTRO DA PÁGINA FB4	ENDEREÇO DE PÁGINA 0	DESLOCAMENTO DENTRO DA PÁGINA FB4
DESLOCAMENTO DA INSTRUÇÃO		05		4D
NOVO ENDEREÇO	0	FB9	0	001

Se dois bytes de instrução estão separados inadvertidamente em duas páginas diferentes, ocorre um erro.

O esboço seguinte mostra os limites de páginas. Os primeiros defeitos designam a página, os próximos à localização.





Quando o contador de programas (PC) estiver na instrução de posição 1 FFF (última palavra de página 1), em vez de ir buscar a outra metade da instrução na página 2, é executado um retorno à primeira palavra da página 1, produzindo um erro.

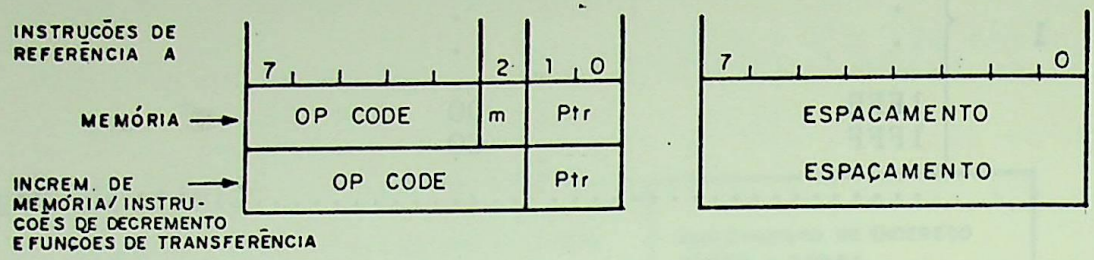
O usuário deve organizar seu programa para evitar esta situação.

FORMATOS DE ENDEREÇAMENTO:

Na execução, dados e instruções são armazenados em específicas localizações e carregadas no acumulador e outros registros.

Como a CPU, memórias e periféricos se ligam a um barramento comum. Qualquer instrução para endereçar a memória pode ser usada para endereçar os periféricos.

O formato das instruções referentes à memória, são:



Na instrução de referência, à memória usam os métodos de endereçamento PC-relativo, indexado ou auto-indexado.

As instruções de memória incremente/decremente e transferências usam os métodos de endereçamento PC relativo indexado.

O endereçamento imediato é um modo de endereçamento específico do grupo de instruções imediatas.

Para as instruções de referência, a memória PC-relativo, indexado e auto-indexado o conteúdo do registro extensão é substituído pelo espaçamento se o espaçamento da instrução iguala a $(128)_{10}$.

MODOS DE ENDEREÇAMENTO

TIPO DE ENDEREÇAMENTO	FORMATO DO OPERANDO		
	m	ptr	ESPAÇAMENTO
PC – RELATIVO	0	0	-128 a +127
INDEXADO	0	1, 2 ou 3	-128 a +127
IMEDIATO	1	0	-128 a +127
AUTO- INDEXADO	1	1, 2 ou 3	-128 a +127

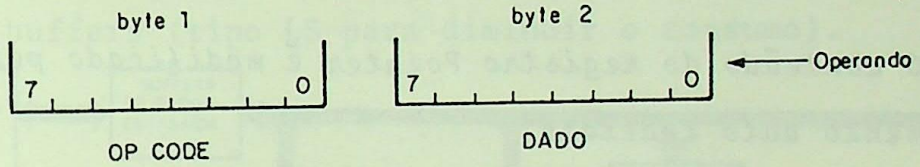
ENDEREÇAMENTO PC-RELATIVO

É executado adicionando o valor do espaçamento do campo da instrução ao valor normal do contador, de programa.

O espaçamento é de 08 bits tipo complemento de dois, de modo que os limites de endereços, vão de -128 a + 127 bytes a partir da posição do PC.

ENDEREÇAMENTO IMEDIATO

Usa uma instrução de dois bytes, sendo a segunda parte do byte de instrução o operando.



ENDEREÇAMENTO INDEXADO

Este modo de endereçamento habilita o programador endereçar qualquer localização de memória, com o uso do registro Pointer e do espaçamento. Quando o endereçamento indexado é especificado numa instrução, o espaçamento é adicionado ao conteúdo do registro Pointer especificado, para formar o endereço efetivo.

O conteúdo do registro Pointer não é modificado pelo endereçamento indexado. Este endereçamento é usado para entrar em tabelas ou subrotinas, controle de transferência para uma sessão da mesma página fora do limite da transferência do PC-relativo.

As regras dos limites das páginas devem,

ser acauteladas.

ENDEREÇAMENTO AUTO-INDEXADO:

É idêntico ao endereçamento indexado na capacidade de incrementar (espaçamento positivo) e decrementar (espaçamento negativo) o conteúdo do registro Pointer.Porém , se o espaçamento é maior ou igual a zero, o registro Pointer é usado como um endereço efetivo e depois é incrementado pelo espaçamento após este endereço ser buscado ou memorizado.

Se o espaçamento é menor que zero, o registro pointer é decrementado pelo espaçamento após o endereço efetivo ser operado.

NOTA: *O conteúdo do registro Pointer é modificado pelo endereçamento auto indexado.*

MODULO DE APLICAÇÃO

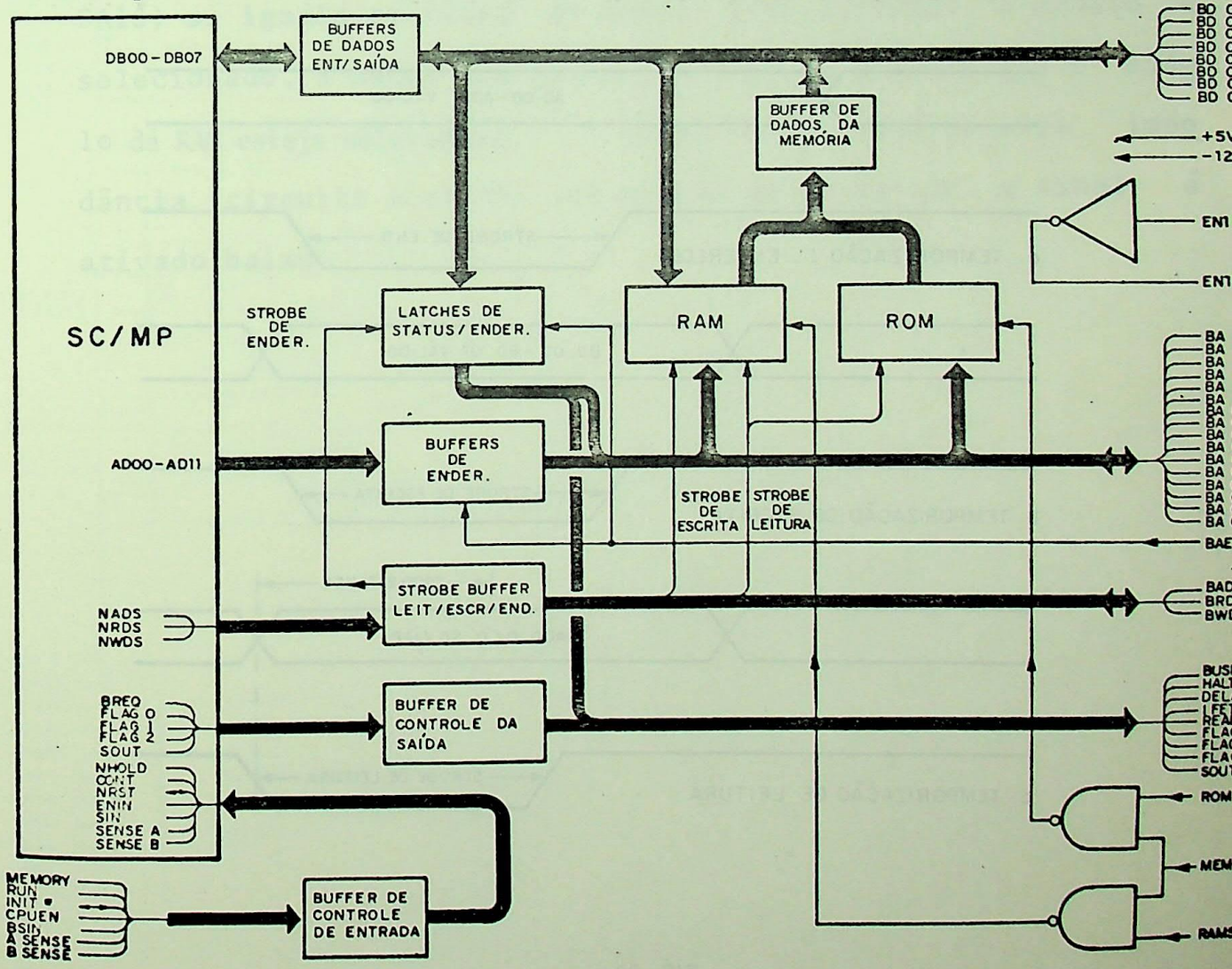
CPU - (Unidade Central de Processamento)

O diagrama em bloco funcional da CPU do módulo de aplicação do SC/MP está mostrado na figura 22. A CPU possui memórias RAM, ROM e todos os circuitos de buffer e latchs necessários à execução e armazenamento das instruções.

A memória RAM possui 256 bytes e a PROM 512 bytes (podendo ser expandido para até 2048 bytes).

Como mostra a figura 22, os quatro bits mais significativos do endereço (AD12-AD15) e os sinais de status de entrada/saída (flag-H, Flag-D, Flag-I e Flag-R) são memorizados em um latch.

As outras linhas de entrada/saída passam por buffers (tipo LS para diminuir o consumo).



Se o sistema de memória exceder a capacidade de um cartão, então três sinais de controle devem ser dados o usuário(MEMSEL , RAMSEL e ROMSEL). Se o módulo da RAM é selecionado,o sinal MEMSEL é baixo.

Para o módulo PROM/ROM, o sinal MEMSEL também é aplicado da mesma forma. Porém, sendo alto, é possível selecionar a RAM com RAMSEL ou a ROM com ROMSEL. A temporização da CPU é mostrada na figura 23.

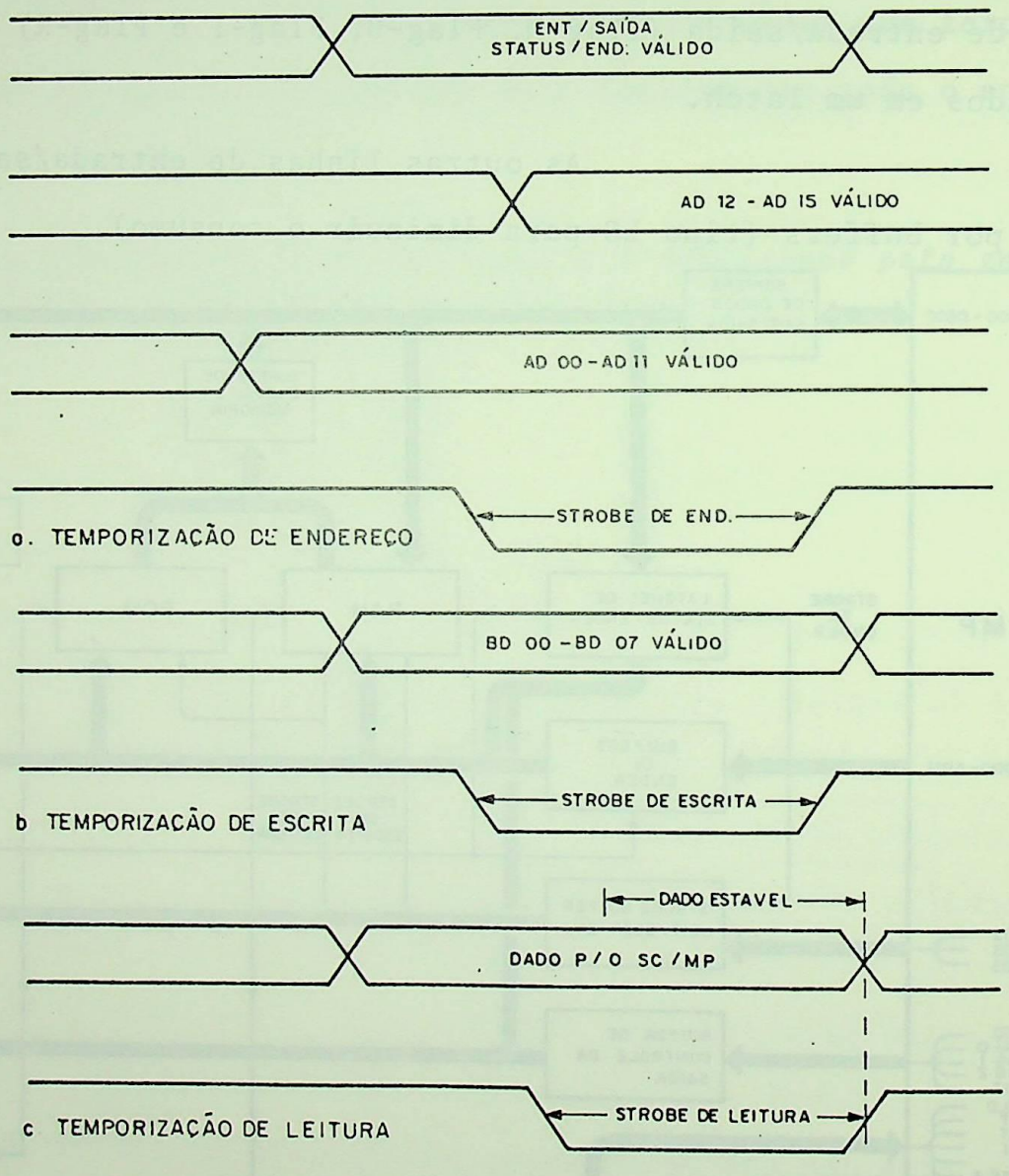


FIG. 23

RAM

A figura 24 mostra o diagrama em blocos do módulo da RAM.

Basicamente, o módulo contém 16 RAM's de 1K por 1bit, organizados em arranjos de 2k de 8 bits(2.048 bytes), buffers de entrada e saída,circuitos de controle e lógica de seleção do módulo.

Os sinais de strobe(BRDS) de leitura e (BWDS) devem permanecer inativos até serem selecionados pelo circuito de controle.

A lógica de preset (PS1 até PS5) permite a RAM assumir qualquer uma das 32 possíveis designações.

Quando o código de endereço (BA11 até BA15) se iguala ao código de preset (PS1 até PS5), o módulo é selecionado, a menos que NMD estiver baixo.A menos que o módulo da RAM esteja selecionado, o sinal MEMSEL está em alta impedância (circuito aberto); com a seleção do cartão, o sinal é ativado baixo.

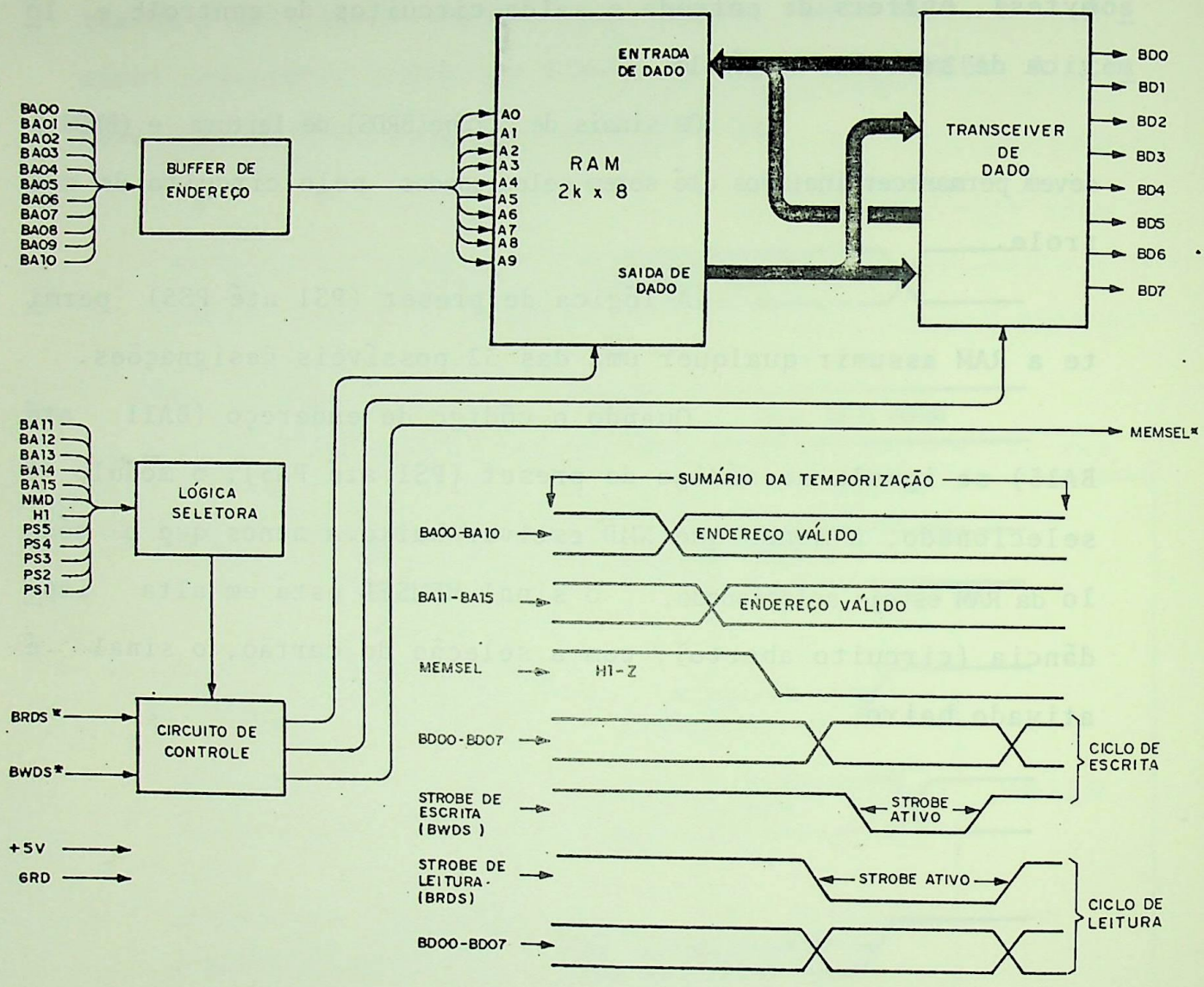

BA00
BA01
BA02
BA03
BA04
BA05
BA06
BA07
BA08
BA09
BA10
BUFFER DE ENDEREÇO
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
RAM
2k x 8
ENTRADA DE DADO
SAIDA DE DADO
TRANSCEIVER DE DADO
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7
MEMSEL*
BA11
BA12
BA13
BA14
BA15
NMD
H1
PS5
PS4
PS3
PS2
PS1
LÓGICA SELETORA
BRDS*
BWDS*
CIRCUITO DE CONTROLE
+5V
GRD
SUMÁRIO DA TEMPORIZAÇÃO
BA00-BA10
ENDERECO VÁLIDO
BA11-BA15
ENDEREÇO VÁLIDO
MEMSEL
HI-Z
BD00-BD07
CICLO DE ESCRITA
STROBE DE ESCRITA (BWDS)
STROBE ATIVO
STROBE DE LEITURA (BRDS)
STROBE ATIVO
CICLO DE LEITURA
BD00-BD07

FIG. 24

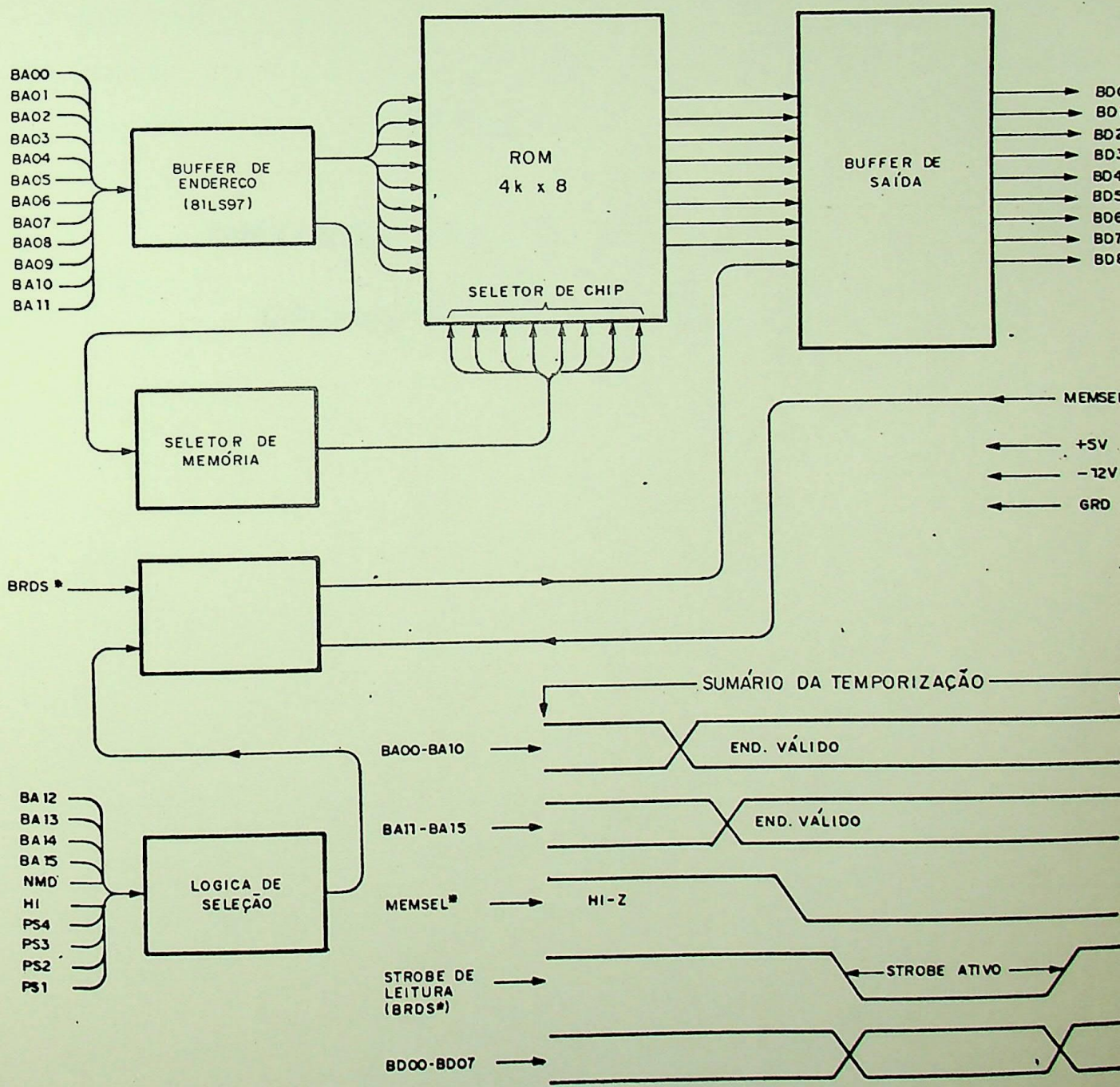
Escola Federal de Engenharia
Biblioteca
MAUÁ
Itajubá

PROM/ROM:

Exceto pelos diferentes controles e organização do tipo de memória, o módulo PROM/ROM é idêntico ao da RAM (fig.24).

O módulo PROM/ROM usa arranjo de 4k por 8 bits, 12 bits são necessários para endereços e quatro presets(PS1 até PS4) são necessários para selecionar o módulo.

Como a memória é PROM ou ROM, o strobe de escrita (NWDS) não é necessário.



CAPÍTULO IV

CONSTITUIÇÃO BÁSICA DO SC/MP LCDS

(LOW COST DEVELOPMENT SYSTEM)

CONTROLES, CONECTORES E INDICADORES:

TECLA INIT : Dá o início ao microprocessador SC/MP. Então, quando pressionada durante o modo de operação RUN, termina a execução do programa e faz o módulo entrar no modo DEBUG; quando pressionada durante o modo de operação DEBUG, termina os comandos em progresso e faz o módulo (LCDS) voltar ao modo DEBUG no ponto de entrada.

TECLA HALT: Funciona sómente durante o modo de operação RUN Quando pressionada no modo RUN, para o programa (após a instrução que está sendo executada ou completada) e faz o módulo , entrar no modo DEBUG.

TECLA MEM ADDR/ABORT : Funciona somente quando o módulo está no modo DEBUG e o painel de operação está habilitado. Quando está pressionada, quando as entradas numéricas estão em progresso corta o comando anterior e preenche o display com pequenos zeros (0000.00). Quando pressionada enquanto não existem entradas numéricas, coloca no display uma série de traços, indicando que o microprocessador está pronto para receber um endereço de memória.

TECLA RUN : Funciona sómente quando o módulo se encontra no modo DEBUG e o painel de operação está habilitado. Ao ser pressionada, termina o modo DEBUG e providencia a execução do programa que inicia no endereço armazenado no contador de programa .

DISPLAY : Seis caracteres de 7 segmentos associados com o exame e alteração dos conteúdos das localizações de memória e valores armazenados no contador de programa, registros e acumulador.

TECLA READ PC/0 : Funciona somente quando o módulo está no modo DEBUG e o painel é selecionado, ou quando é endereçado pelo programa do usuário, enquanto o modo de operação RUN é habilitado. Quando pressionado, enquanto o teclado estiver modo de comando, faz com que o conteúdo do PC seja colocado no display; quando pressionado, enquanto o teclado estiver no modo numérico, prove na saida valor hexadecimal 0.

TECLA READ P1/1 : O funcionamento é idêntico ao do ítem anterior, somente que agora, ou faz a leitura do valor memorizado no registro P1, ou acrescenta o valor hexadecimal 1 ao display.

TECLA READ P_2/2 : Ìdem ao anterior, sòmente que agora relacionado com o registro P_2 e o valor hexadecimal 2.

TECLA READ P_3/3 : Ìdem ao anterior, agora ligado com o registro P_3 e o valor hexadecimal 3.

TECLA READ AC/4 : Ìdem ao anterior, agora ligado com o registro acumulador AC e o valor hexadecimal 4.

TECLA READ EX/5: Ìdem, agora ligado com o registro EXTENSÃO(E) e o valor hexadecimal 5.

TECLA READ ST/6: Ìdem, agora ligado ao registro STATUS(s) e o valor hexadecimal 6.

TECLA READ MEM/7: Funciona quando o módulo está no modo DEBUG e o painel de operações é selecionado, ou quando endereçado pelo programa do usuário, enquanto o modo de operação RUN está habilitado. Quando pressionada, enquanto o teclado está no modo comando, coloca-se no display o valor memorizado no endereço referência, cada vez que volta a ser pressionada apresenta o conteúdo da próxima posição de memória a partir da referência. Quando atuada no modo numérico, acrescenta ao display o número hexa 7.

TECLA LOAD PC/8: Funciona somente quando o módulo está no modo DEBUG e o painel de operação é selecionado, ou quando endereçado pelo programa do usuário, enquanto o modo de operação RUN está habilitado. Quando pressionado, enquanto o teclado está no modo comando, habilita a alteração do valor memorizado no PC e quando pressionado, enquanto o teclado está no modo numérico, acrescenta o valor hexadecimal 08 ao display.

TECLA LOAD P_1/9 : Idem ao anterior, agora ligado ao registro P_1 e o valor hexa 9.

TECLA LOAD P_2/A : Idem ao anterior, agora ligado ao registro P_2 e o valor hexa A.

TECLA LOAD P_3/B : Idem ao anterior, agora ligado ao registro P_3 e o valor hexa B.

TECLA LOAD AC/C: Idem ao anterior, agora ligado ao registro Acumulador(AC) e o valor hexa C.

TECLA LOAD EX/D: Idem ao anterior, agora ligado ao registro EXTENSÃO(E) e o valor hexa D.

TECLA LOAD ST/E

Idem ao anterior, agora ligado ao registro STATUS(s) e o valor hexa E.

TECLA LOAD MEM/F

O funcionamento é idêntico ao dos comandos anteriores. Quando pressionada, permite a alteração do conteúdo da memória, guardado no endereço referência; quando é repressionada, vai permitindo a alteração sequencial da memória a partir do endereço referência. De outro modo, acrescenta ao display o valor hexa F.

CHAVE HALT INST : Associado com o modo de operação RUN do módulo. Quando é selecionada para DEBUG, o módulo termina o modo RUN e entra no modo DEBUG após a execução da instrução HALT.

Quando selecionada para PULSE, o modo de operação RUN não é afetado pela execução da instrução HALT.

CHAVE HALT MODE: Associada com a operação DEBUG do módulo. Quando é selecionada para PNL, habilita o comando operacional DEBUG entrar via teclado. Quando é selecionada para TTY, habilita o comando operacional DEBUG entrar via teclado do TTY.

CHAVE RUN MODE: Associada com o modo de operação RUN do módulo. Quando é selecionada para step, habilita a execução de cada instrução separada do programa. Quando é selecionada para CONT, habilita a execução normal do programa.

CONECTORES J1-J4: Conectores de interface para as placas de aplicação da família SC/MP para hardware e software.

CONECTOR J8 - Habilita interconexões para várias aplicações específicas.

RÉGUA TERMINAL TB1 : Habilita a interconexão com as fontes primárias de alimentaçao e TTY (teletipo)

MODOS DE OPERAÇAO

O módulo possui basicamente dois modos de operação: modo RUN e modo DEBUG. No modo RUN, o módulo executa a funçao de monitoração e o cartão da CPU do SC/MP opera como parte do sistema de aplicaçao do usuário, assim um programa pode ser executado nesta forma de operação. No modo DEBUG, o módulo faz a função de controle, habilitando a CPU do SC/MP a executar o "filmware" residente de DEBUG sem afetar o sftware, ou o hardware de aplicação. Neste modo é possível comunicar com o módulo via teclado de controle.

O modo DEBUG inclui o seguinte:

. Coloca no display o conteúdo do contador de programa, registros e acumulador em formato hexadecimal.

. Coloca no display o conteúdo de qualquer locação de memória, no formato hexadecimal.

. Altera o conteúdo do contador de programa, registros e acumulador.

. Altera o conteúdo de qualquer posição de memória.

. Inicia a execução do programa em qualquer posição de memória

. Interrompe a execução do programa em qualquer ponto sem alteração do sistema.

Quando o modo DEBUG é selecionado, ele entra via uma sub-rotina que é iniciada, quando a instrução cor

rente no modo RUN é completada.

A proposição desta sub-rotina é memorizar os conteúdos do contador de programa, registros e acumulador nas posições de memória que vão de 77F5 até 77FF para permitir um subsequente retorno ao modo RUN sem perda do status .

Durante o processamento desta sub- rotina, o conteúdo do contador de programa (PC) é incrementado de um, de modo a apontar a próxima instrução a ser buscada. Quando o modo RUN voltar a ser selecionado, os conteúdos do registro e acumulador não são alterados.

ENDEREÇO: 77 F5 (*byte de mais alta ordem do contador de programa*)

77 F6 (*byte de baixa ordem do PC*)

77 F7 (*registro* P_1 *byte de alta ordem*)

77 F8 (*registro* P_1 *byte de baixa ordem*)

77 F9 (*registro* P_2 *byte de alta ordem*)

77 FA (*registro* P_2 *byte de baixa ordem*)

77 FB (*registro* P_3 *byte de alta ordem*)

77 FC (*registro* P_3 *byte de baixa ordem*

77 FD (*acumulador*)

77 FE (*Registro extensão*)

77 FF (*registro etatus*)

NOTA: *O módulo reserva os endereços de 7000 até 7FFF para operações de DEBUG. Todos os restantes endereços (0000 até 6FFF e 8000 até FFFF) são disponíveis para uso do programador.*

OPERAÇÃO PELO TECLADO

SELEÇÃO DO MODO RUN/DEBUG:

O modo **RUN** é selecionado simplesmente pressionando a tecla **RUN** e o modo **DEBUG** é selecionado diretamente ao término do modo **RUN**. As teclas associadas com a operação do modo RUN são :

- *RUN*
- *HALT*
- *RUN MOVE*
- *HALT INST*

Todas as outras (exceto INIT) são associadas com o modo DEBUG.

Para selecionar o modo RUN, se procede:

1. Colocar a chave RUN MODE para CONTIN, para habilitar a execução do programa de aplicação do usuário, ou para STEP, habilitando a execução passo a passo do programa. Selecionando, STEP, uma instrução é executada a cada atuação na tecla **RUN** e o módulo retorna automaticamente ao modo DEBUG.

Selecionando CONTIN, o programa é executado completamente, quando se atua na tecla **RUN**.

2. Colocar a chave **HALT INST** para DEBUG ou PULSE para habilitar ou desabilitar respectivamente a introdução de uma parada no programa.

Quando a chave **HALT INST** está em DEBUG, o aparecimento de uma instrução **HALT** causa uma passagem do modo RUN para o modo DEBUG.

Quando a chave **HALT INST** está em PUL

SE, o modo RUN não é afetado por uma instrução HALT no programa.

3. Para iniciar o programa, basta atuar na TECLA READ e depois na tecla RUN, isto,quando se deseja iniciar o programa no valor contido no PC.

Quando se deseja iniciar o programa em outro valor do PC, pressionar a tecla LOAD PC, introduzir o valor desejado, atuando nas teclas numéricas e então atuar na TECLA RUN, iniciando o programa.

Após iniciado o programa, o display permanece apagado até o modo RUN estar terminado e voltar ao modo DEBUG. Para voltar o modo DEBUG, é necessário:

1. Atuar na TECLA INIT
2. Atuar na TECLA HALT
3. A chave RUN/STEP está em STEP e uma instrução foi executada.
4. A chave HALT INST está em DEBUG e aparece uma instrução HALT
5. A entrada DEBUG do módulo é externamente colocada em nível baixo.

COMANDOS DE DEBUG

Quando o módulo está no modo DEBUG, o teclado funciona sob o controle do filmware de DEBUG.

Os comandos sao indicados pelas legendas das teclas READ,PC,READ P_1, etc). Os números são indicados, por valores hexadecimais (0,1,2...). A operação do filmware de DEBUG é tal, que a saída de comando da tecla está sempre habili

Quando isto ocorre, o filmware de DEBUG habilita as teclas numéricas e o display apresenta (-----) ou (--).

Pressionando a tecla, o valor em hexadecimal entra no display da esquerda para a direita.

Após ter sido completado o registro, automaticamente é executado o comando e o filmware de DEBUG reabilita novo comando pelas teclas. Quando é necessário cancelar um comando já dado pelas teclas, basta atuar nas TECLAS MEM ADDR ABORT, o display irá apresentar uma sequência de pequenos zeros (oooo-oo), indicando que o comando foi cancelado e está a espera de outro.

Deste modo, sob o controle do filmware de DEBUG, é possível examinar ou modificar o conteúdo do contador de programas (PC), Registros (P_1, P_2, P_3, EX, ST) e Acumulador (AC).

É também possível examinar ou alterar o conteúdo de qualquer localização da memória via tecla MEM ADDR/ABORT, tecla READ MEM e tecla LOAD MEM.

CAPACIDADE DE INTERFACE DO MÓDULO:

O barramento de Interface do módulo SC/MP LCDS foi projetado de modo a permitir a CPU do SC/MP funcionar como parte do sistema de aplicação do usuário quando no modo RUN e como parte do módulo LCDS quando no modo DEBUG.

A maneira do módulo completar este procedimento está definida na reserva de endereços (7000 até 7FFF) para a operação DEBUG, permitindo os restantes endereços (de 65K)

serem manuseados pelo usuário. Com um sofware adequado, o modo RUN cobre os endereços: de 0000 até 6FFF e de 8000 até FFFF .

Os conectores J1 até J4 possuem 72 pinos fêmea para conexão dos cartões de expansão do módulo LCDS, ou para interface com circuitos externos, como será o caso do nosso programador de memórias.

BARRAMENTO DE INTERFACE:

Neste ponto revemos a descrição dos sinais de interface disponível no barramento, que é apresentado fisicamente pelos conectores J1 até J4.

PINO 1 (+5V): fonte primária DC para alimentação da CPU do SC/MP.

PINO 2 (GND): Referência (terra) para sinais e alimentação do CPU do SC/MP.

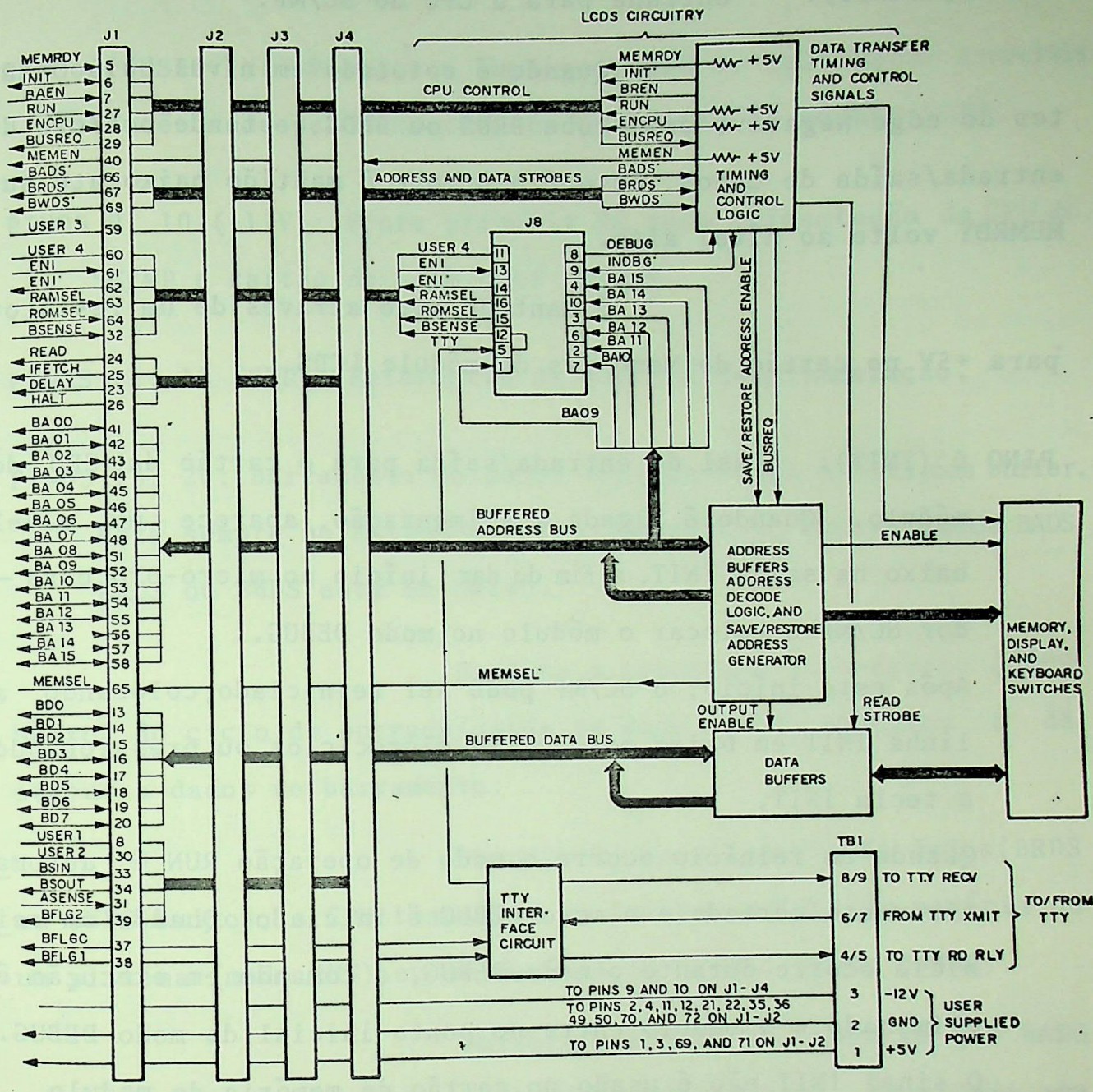

LCDS CIRCUITRY
J1
J2
J3
J4
MEMRDY 5
INIT 6
BAEN 7
RUN 27
ENCPU 28
BUSREQ 29
MEMEN 40
BADS' 66
BRDS' 67
BWDS' 68
USER 3 59
USER 4 60
ENI 61
ENI' 62
RAMSEL 63
ROMSEL 64
BSENSE 32
READ 24
IFETCH 25
DELAY 23
HALT 26
BA 00 41
BA 01 42
BA 02 43
BA 03 44
BA 04 45
BA 05 46
BA 06 47
BA 07 48
BA 08 51
BA 09 52
BA 10 53
BA 11 54
BA 12 55
BA 13 56
BA 14 57
BA 15 58
MEMSEL 65
BD0 13
BD1 14
BD2 15
BD3 16
BD4 17
BD5 18
BD6 19
BD7 20
USER1 8
USER2 30
BSIN 33
BSOUT 34
ASENSE 31
BFLG2 39
BFLG0 37
BFLG1 38
CPU CONTROL
ADDRESS AND DATA STROBES
MEMRDY
INIT'
BREN
RUN
ENCPU
BUSREQ
MEMEN
BADS'
BRDS'
BWDS'
+5V
+5V
+5V
+5V
TIMING AND CONTROL LOGIC
DATA TRANSFER TIMING AND CONTROL SIGNALS
J8
USER 4
ENI
ENI'
RAMSEL
ROMSEL
BSENSE
TTY
DEBUG
INDBG'
BA 15
BA 14
BA 13
BA 12
BA 11
BA10
BA09
SAVE/RESTORE ADDRESS ENABLE
BUSREQ
BUFFERED ADDRESS BUS
ADDRESS BUFFERS ADDRESS DECODE LOGIC, AND SAVE/RESTORE ADDRESS GENERATOR
ENABLE
MEMORY, DISPLAY, AND KEYBOARD SWITCHES
MEMSEL'
OUTPUT ENABLE
READ STROBE
BUFFERED DATA BUS
DATA BUFFERS
TB1
TTY INTER-FACE CIRCUIT
8/9 TO TTY RECV
6/7 FROM TTY XMIT
4/5 TO TTY RD RLY
TO/FROM TTY
TO PINS 9 AND 10 ON J1-J4
TO PINS 2, 4, 11, 12, 21, 22, 35, 36 49, 50, 70, AND 72 ON J1-J2
TO PINS 1, 3, 69, AND 71 ON J1-J2
3 -12V
2 GND
1 +5V
USER SUPPLIED POWER

FIG. 2

PINO 3 (+5V): Idem ao pino 1

PINO 4 (GND): Idem ao pino 2

PINO 5(MEMRDY): Entrada para a CPU do SC/MP.

Quando é colocado em nível baixo antes do edge negativo do strobe BRDS ou BWDS, estende o ciclo de entrada/saída de dados, isto é,o strobe é mantido baixo até que MEMRDY volte ao nível alto.

É mantido alto através de um resistor para +5V no cartão de Memórias do módulo LCDS.

PINO 6 (INIT): Sinal de entrada/saída para o cartão da CPU do módulo. Quando é ligada a alimentação, aparece um nível baixo na saída INIT, a fim de dar início ao micro-processador SC/MP e colocar o módulo no modo DEBUG.

Após este início, o SC/MP pode ser reiniciado,colocando a linha INIT em baixo,por quatro microciclos ou pressionando a tecla INIT.

Quando um reinício ocorre,o modo de operação RUN é automaticamente cortado e o modo DEBUG é iniciado. Quando um reinício ocorre durante o modo DEBUG, o comando em execução é eliminado e o módulo entra no ponto inicial do modo DEBUG.

O sinal INIT não é usado no cartão de memória do módulo.

PINO 7 (BAEN): Entrada de controle do CPU.

Quando alta, habilita a saída de endereço BA00 até BA15 durante o ciclo de entrada/saída de dado.

Quando baixa,coloca os bits de endereço em alta impedância (TRISTATE) para permitir a endereços externos serem coloca

dos em contato com o barramento de endereços. Mantida baixa durante DEBUG e rotina de rememorização, a fim de controlar o cartão do CPU do SC/MP. Não é usado no cartão de memórias.

PINO 8 (USER 1): Permite a implementação de aplicações especiais. Não é usado pelos cartões de aplicação do módulo LCDS.

PINOS 9, 10 (-12V): Fonte primária DC para alimentação da CPU do SC/MP e cartão de READ ONLY MEMORY.

PINOS 11, 12 (GND): Referência de sinal e de alimentação.

PINOS 13, 20: Barramento de dados entrada/saída, 8 bits, com buffer. Está sempre no estado de alta impedância, exceto quando BADS, BRDS ou BWDS está em baixo.

Durante o intervalo de endereço (DADS baixo) do ciclo de entrada/saída de dado, a CPU endereça e dá status a dados no barramento.

Durante a transferência de dados (BRDS ou BWDS baixo), a CPU transmite ou recebe um byte da memória de um sistema (modo RUN) ou da memória do módulo (modo DUBUG).

Durante o espaço de tempo em que BADS está no nível baixo (intervalo de endereçamento), os sinais nos pinos 13, 14, 15, 16, 17, 18, 19 e 20 apresentam as seguintes características:

- PINO 13 (BDO) BA12: quarto bit mais significativo de um endereço de 16 bits.

Escola Federal de Engenharia
Biblioteca
MAUÁ

- PINO 14 (BD1) BA13: terceiro bit mais significativo de um endereço de 16 bits.

- PINO 15 (BD2) BA14: segundo bit mais significativo de um endereço de 16 bits.

- PINO 16 (BD3) BA15: Bit mais significativo de um endereço de 16 bits.

- PINO 17 (BD4) READ: quando alto, o ciclo de entrada de dados inicia; quando baixo o ciclo de saída de dados inicia.

- PINO 18 (BD5) IFETCH: quando alto, o primeiro byte da instrução está sendo buscada e processada pelo módulo no modo DEBUG.

- PINO 19 (BD6) DELAY: quando alto, indica que o ciclo de atraso iniciou, isto é, o segundo byte da instrução Delay foi buscado.

- PINO 20 (DB7) HALT: quando alto, indica que a instrução HALT está sendo executada, faz o módulo terminar o modo RUN e entrar no modo DEBUG quando a chave HALT INST é colocada para DEBUG.

PINO 21, 22 (GND): Nível de referência para alimentação e sinais.

PINO 23 (DELAY): Saída de Delay do Flag de Status da CPU do SC/MP.

Armazenado nível alto no edge positivo do strobe BADS, indicando que o ciclo de delay iniciou, de outro modo, armazenado nível baixo no edge positivo do strobe BADS.

Não é usado nos cartões de memória.

PINO 24 (READ): Memorizada a saída do flag status Read da CPU do SC/MP.

Armazena nível alto ou baixo, no edge positivo do strobe ' BADS,para indicar se o ciclo de leituras ou escrita de dados está em execução.

Não é usado nos cartões de memória.

PINO 25 (IFETCH): Memoriza a saída do flag de status da instrução FETCH do CPU do SC/MP. Memoriza nível alto no edge positivo do strobe BADS,quando o primeiro byte da instrução está para ser buscado durante a execução do ciclo de entrada/saída de dados. De outro modo,memoriza nível baixo no edge positivo do strobe BADS. Não é usado nos cartões de memória.

PINO 26 (HALT) : Memoriza a saída de flag de status do HALT da CPU do SC/MP. Memoriza um nível alto no edge positivo do strobe BADS, para indicar que uma instrução HALT está sendo executada. De outro modo,memoriza nível baixo no edge positivo do strobe BADS.

Não é usado nos cartões de memória.

PINO 27 (RUN): Entrada de habilitação do modo RUN da CPU do SC/MP. Quando nível alto, habilita a execução normal do programa, quando baixo, a operação do SC/MP é suspensa (após ter sido completada a instrução corrente), sem perda dos Status internos. Colocada em nível alto por um resistor a +5V no módulo.

Não é usado nos cartões de memória.

PINO 28 (ENCPU): Entrada de habilitação de acesso ao barramento para a CPU. Quando em nível alto, habilita o SC/MP executar o ciclo de entrada/saída.

Quando em baixo, inibe a execução do ciclo de entrada/saída. É ligado ao +5V por um resistor.

Não é usado nos cartões de memória.

PINO 29 (BUSREQ): Saída de solicitação ao barramento da CPU do SC/MP. Vai para o nível alto quando o SC/MP solicita ao barramento o início do ciclo de entrada/saída e permanece alto até que o ciclo esteja terminado. Processado pelo módulo para controle da saída de endereço durante o DEBUG e Rotina de rememorização.

Não é usado nos cartões de memória.

PINO 30 (USER 2): Ajuda a implementação de funções especiais pelo usuário. Não é usado no cartão da CPU nem nos cartões de memória.

PINO 31 (ASENSE): Entrada Senso/solicitação de Interrupção para a CPU do SC/MP. Serve como solicitação de interrupção, quando o flag de habilitação interna de interrupção (IE) é setado e como entrada geral de senso quando o Flag IE é resetado.

Não é usado nos cartões de memória.

PINO 32 (BSENSE): Entrada de condição geral de senso para a CPU do SC/MP. Esta entrada é dedicada a recepção de teletipos.

PINO 33 (BSIN): Entrada Série com buffer para a CPU do SC/MP. Quando SIO (serial input/antput), instrução de entrada/saída, for executada, os dados nesta linha são deslocados para o registro Extensão de 8 bits.
Não é usado nos cartões de memória.

PINO 34 (BSOUT): Saída série com buffer da CPU do SC/MP. Quando a instrução SIO for executada, os dados nesta linha são rapidamente deslocados para fora do registro Extensão.
Não é usado nos cartões de memória.

PINO 35,36 (GND): Referência para fonte e sinais.

PINO 37 (BFLGO): Saída com buffer de Flag da CPU do SC/MP. Durante o modo DEBUG é usado pelo módulo para transmitir dado para o TTY. Não é usado nos cartões de memória.

PINO 38 (BFLG1): Saída com buffer de Flag da CPU do SC/MP . Durante o modo DEBUG é usado para controle do relé de leitura de fita do TTY. Não é usado nos cartões de memória.

PINO 39 (BFLG2): Saída com buffer de Flag da CPU do SC/MP.
Não é usado nos cartões de memória.

PINO 40 (MEMEN): Entrada de habilitação de endereço para os cartões de memória, quando em nível alto habilita o endereçamento e cartões de memória. Quando em nível baixo, inibe. É colocado em +5V através de um resistor. Não é usado no cartão da CPU do SC/MP.

PINO 41 até 48 (BA00 até BA07):

PINO 51 até 58 (BA08 até BA15):

Barramento de endereço de 16 bits com buffer, o qual é colocado no estado de alta impedância, exceto quando o ciclo de entrada/saída está acontecendo.

Usado pela CPU do SC/MP para se comunicar com o sistema de aplicação ou com os cartões de memória, exceto durante DEBUG ou Rotina de Rememorização.

Durante o tempo de DEBUG ou da Rotina, o módulo seta para nível baixo o sinal BAEN, desabilitando a saída de endereço da CPU e força o endereço a controlar o modo DEBUG.

O cartão da CPU possui internamente um comparador que, continuamente, compara o endereço do barramento com o endereço dado pelo cartão para habilitar ou desabilitar o cartão como for o caso.

PINO 49, 60 (GND): Referência para sinais e fonte de alimentação.

PINO 59, 60 (USER 3, USER 4): Tem uso em aplicações especiais.

PINO 61, 62 (EN1, EN1*): Sinal de entrada/saída da CPU do SC/MP, usada para controle do cartão de memória.

A entrada EN1 é invertida para gerar EN1*. Ambos os sinais são acessíveis no conector J8.

PINO 63 (RAMSEL): Entrada seletora da memória RAM do cartão da CPU do SC/MP. Quando em nível alto, habilita o cartão de memórias RAM se a entrada MENSEL do cartão da CPU também estiver em alto.

Quando um nível baixo desabilita o cartão de memórias.

PINO 64 (ROMSEL): Entrada que seleciona a memória ROM do cartão da CPU do SC/MP.

Quando alto,habilita o cartão de memórias ROM,se a entrada MEMSEL* também é alta,quando baixo,desabilita a memória ROM.

PINO 65 (MENSEL): Entrada seletora de memória para a CPU do SC/MP.

Se mantem baixa durante o ciclo de entrada saída de dados quando o módulo ou o cartão de memórias é endereçado para todos os outros tempos,está ligado a +5V por um resistor.

Então, habilita a memória da CPU do SC/MP,se o sistema de aplicação ou o cartão de memória não está sendo endereçado.

PINO 66 (BADS): Saída strobe de endereço com buffer, atirada no nível baixo, indica que endereços válidos e status estão presentes no barramento.

Usado pelo cartão de memória para habilitar a comparação do endereço presente no barramento,com endereços prefixados.

PINO 67 (BRDS): Saída strobe de leitura de dados, atirada no nível baixo, com buffer. Quando em nível baixo, processada pelo Módulo, cartão de memória e CPU, controla a saída de dados do barramento vindos de uma determinada posição de memória.

No edge negativo, os dados disponíveis no barramento são introduzidos no microprocessador SC/MP.

PINO 68 (BWDS): Saída strobe de escrita de dados, com buffer, atirada no nível baixo. Quando em nível baixo, indica que saída válida de dados está presente no barramento.

Processado pelo módulo e cartão da CPU,controla a escrita de dados nas localizações de memória especificadas pelo endereço.

PINO 70, 72 (GND): Referência para a alimentação e sinais do módulo.

DESCRIÇÃO FUNCIONAL

Conhecendo-se os diversos controles modos de operação e conexões do módulo LCDS, como visto nos parágrafos anteriores, neste ponto, faremos um apanhado geral do comportamento global do módulo. Isto requer um conhecimento prévio da operação da CPU do SC/MP, também já enfocada.

Funcionalmente o módulo LCDS consiste de programa residente (firmware), um teclado, um display e todos os circuitos lógicos e buffers necessários para a transferência de informações e controle entre o filmware residente (Modo DEBUG) e o programa do usuário (Modo RUN).

Quando o modo DEBUG é selecionado, ele entra via hardware forçado, isto é, no programa inserido em uma ROM denominado SAVE ROUTINE, que é executada quando o modo RUN termina. Durante a SAVE ROUTINE, o conteúdo do acumulador, registros status e $P_{1,2,3}$, são memorizados.

Então no último passo da SAVE ROUTINE o conteúdo do registro P_3 e do contador de programa é trocado, de modo que o programa residente (filmware) de DEBUG possa iniciar no endereço 7809.

No início do programa de DEBUG, o conteúdo dos registros do SC/MP são guardados na memória do módulo LCDS. Então, a chave HALT MODE deve ser testada para ver se os comandos estão vindo do teclado ou de um TTY.

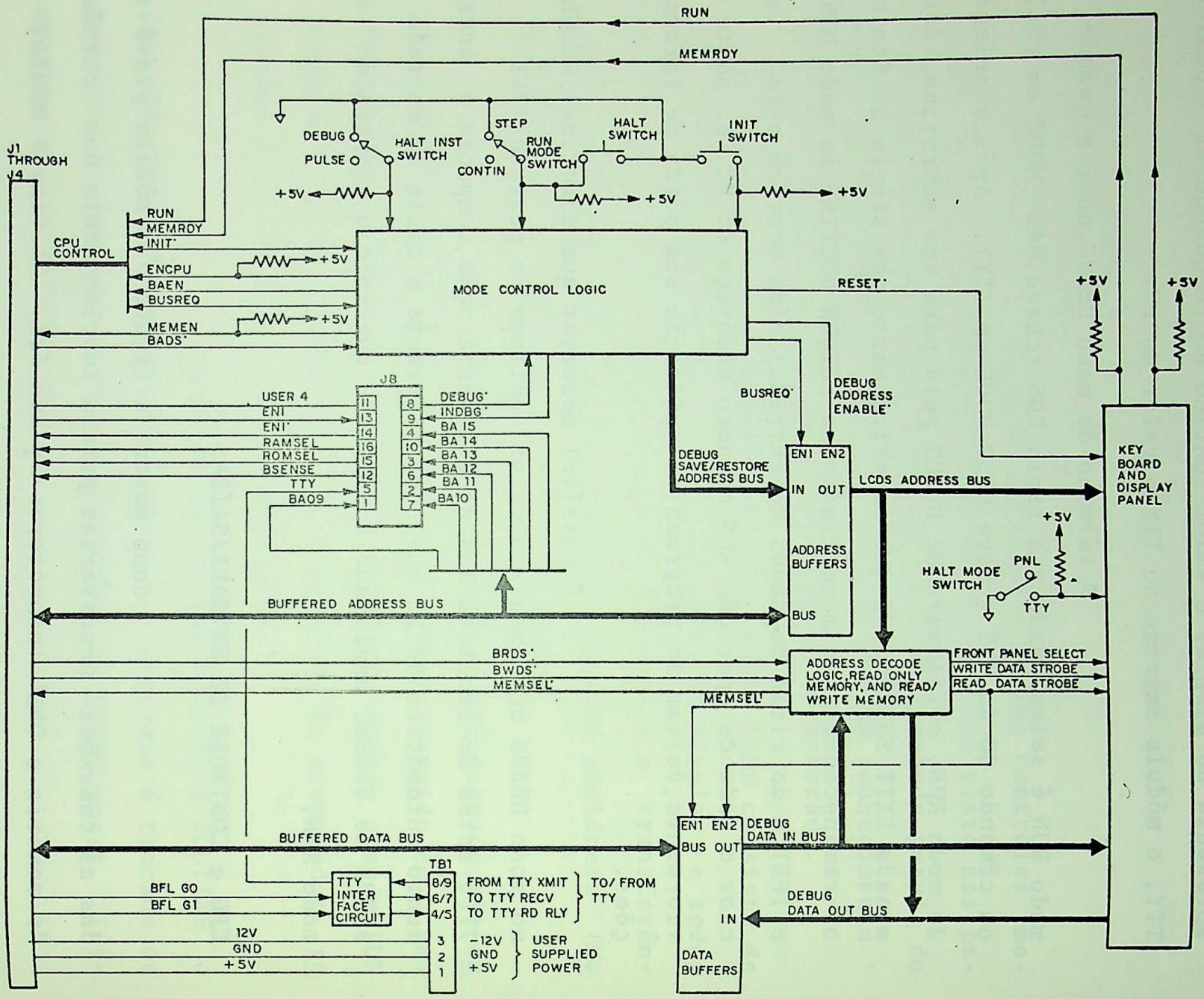
Na posição PNL a chave HALT MODE habilita os controles de DEBUG serem introduzidos pelo teclado e apresenta no display o valor contido no PC, quando na posição TTY, o módulo imprime no TTY o valor do PC.

O término do modo DEBUG ocorre quando o modo RUN é selecionado via tecla RUN (chave HALT MODE em PNL), ou comando GO do TTY (chave HALT MODE em TTY). Após a seleção do modo RUN, o filmware de DEBUG pula para uma subrotina denominada EXIT SUBROUTINE que memoriza parte dos sinais de Status os remanecentes são memorizados na RETURN ROUTINE no modo DEBUG e IFETCH do ciclo de dados de entrada/saída ocorrem para iniciar o modo de operação RUN no novo endereço do PC e, posteriormente, voltar ao programa de aplicação sem perda de informações.

É possível observar que a rotina RETURN no modo DEBUG decrementa o valor contínuo na posição 77F5 e 77F6 antes de recolocá-los no PC. Esta ação compensa o incremento automático do PC que ocorre durante o ciclo de entrada / saída de dados, analisado com IFETCH que inicia a operação do modo RUN.

CPU e CONTROLE DE ENTRADA/SAÍDA:

Como mostra a figura 3 o módulo prevê todas as conexões necessárias para o funcionamento dos cartões de memória e CPU (conectores J_1 até J_4). Os sinais MEMRDY , RUN e ENCPU são colocados no nível +5V para garantir o funcionamento contínuo funcionamento da CPU no modo DEBUG e controles do sistema de aplicação, quando no modo RUN.


RUN
MEMRDY
DEBUG
PULSE
HALT INST SWITCH
STEP
CONTIN
RUN MODE SWITCH
HALT SWITCH
INIT SWITCH
+5V
J1 THROUGH J4
CPU CONTROL
RUN
MEMRDY
INIT'
ENCPU
BAEN
BUSREQ
MEMEN
BADS'
MODE CONTROL LOGIC
RESET'
J8
USER 4
ENI
ENI'
RAMSEL
ROMSEL
BSENSE
TTY
BA09
DEBUG'
INDBG'
BA 15
BA 14
BA 13
BA 12
BA 11
BA10
BUSREQ'
DEBUG ADDRESS ENABLE'
DEBUG SAVE/RESTORE ADDRESS BUS
EN1 EN2
IN OUT
ADDRESS BUFFERS
BUS
LCDS ADDRESS BUS
KEY BOARD AND DISPLAY PANEL
HALT MODE SWITCH
PNL
TTY
BUFFERED ADDRESS BUS
BRDS'
BWDS'
MEMSEL'
ADDRESS DECODE LOGIC, READ ONLY MEMORY, AND READ/ WRITE MEMORY
FRONT PANEL SELECT
WRITE DATA STROBE
READ DATA STROBE
BUFFERED DATA BUS
EN1 EN2
BUS OUT
DEBUG DATA IN BUS
DEBUG DATA OUT BUS
IN
DATA BUFFERS
TB1
TTY INTER FACE CIRCUIT
BFL 60
BFL 61
8/9 FROM TTY XMIT
6/7 TO TTY RECV
4/5 TO TTY RD RLY
TO/ FROM TTY
12V
GND
+5V
3 -12V
2 GND
1 +5V
USER SUPPLIED POWER

De maneira similar,o sinal de entrada MEMEN (habilitação de endereço) também é garantido no nível alto (+5V), de modo que o circuito permanece habilitado,a menos que MEMEN é feito baixo por um comando externo.

Buffers TRI STATE são colocados para interconectar buffers de endereço com barramento de dados, permitindo operação controlada por programas de aplicação. Então a saída da DEBUG Address Select do Mode Control Lógic (Modo lógico de Controle) está em alto, exceto durante as rotinas de DEBUG, SAVE e RESTORE ROUTINES, de modo que os buffers de endereço,continuamente,alimentam com a saída de endereço do cartão da CPU do SC/MP, barramento de endereços do módulo. Se a saída de endereço da CPU indica valores entre 7000 e 7FFF, circuito Address Decode Lógic (Lógica de decodificação de endereços) alimenta com nível baixo a saída MENSEL e habilita o dispositivo de memória apropriado (de 7000 até 7FFF), teclado e Display (7000 até 71FFF).

A saída strobe BRDS ou BWDS faz com que o dado seja lido ou escrito no display ou posição de memória. O strobe BRDS controla o gatilhamento de dados para escrita/leitura de/ para o barramento de dados. Quando BRDS é baixo enquanto o módulo é endereçado, os buffers de dados recebem dados do DEBUG Data Out Bus vindos do barramento de dados e a CPU aceita o dado no edge negativo do strobe BRDS. A todo o tempo, os buffers de dado são alimentados pelo barramento de dados pelo DEBUG Data In Bus, de modo que é possível escrever num endereço do módulo com o strobe BWDS.

HARDWARE FORÇADO (DEBUG SAVE ROUTINE)

DEBUG Save Routine é uma Subrotina contro

lada por hardware, que é executada seguindo o início da operação do módulo ou seleção do modo DEBUG.

A função da subrotina Save Routine, em conjunto com o filmware de DEBUG, é armazenar os Status internos do SC/MP nas localizações de memória entre 77F5 até 77FF, antes do filmware saltar para o teclado ou rotina do TTY.

A execução da DEBUG Save Routine ocorre em qualquer das seguintes condições:

1. Pressionar a tecla INIT;
2. Pressionar a tecla HALT, enquanto o modo de operação RUN está habilitado (inicia a Save Routine após a instrução em progresso ser executada).
3. A chave RUN/STEP é colocada na posição STEP e uma instrução for executada após o modo RUN ser selecionado.
4. A chave HALT INST é colocada na posição DEBUG e a instrução HALT for executada enquanto o modo de operação RUN está habilitado.
5. A entrada DEBUG do módulo é externamente feita baixo, enquanto o modo de operação RUN está habilitado.

O controle lógico habilita a execução da DEBUG Save Routine mantendo as saídas BAEN e Seleção de Endereço em nível baixo, enquanto processa as saídas BUSREQ do SC/MP para forçar uma série endereços para fora do barramento de endereços.

Enquanto BAEN está em nível baixo, as saídas de endereço da CPU são mantidas em alta impedância, mas a operação do resto do circuito não é afetada.

Como mostra a figura 4, fazendo as saídas BAEN e seleção de endereços serem nível baixo no edge positivo do pulso de INIT, ou no edge positivo do strobe BADS associado com o primeiro ciclo de IFETCH dados de entrada/saída, faz a seleção do modo DEBUG.

Enquanto as saídas de endereço da CPU é mantida em alta impedância, pelo nível baixo do sinal BAEN, endereços da DEBUG Save Routine providenciam as necessárias funções para a execução das instruções do SC/MP.

Por exemplo, o primeiro endereço sempre sai durante o ciclo IFETCH entrada/saída, de modo que ísto faz o dado CB ser lido pelo SC/MP e tratado como uma instrução. Este código de operação (opcode) especifica uma instrução de memorizar (STORE ST), a qual requer dois ciclos adicionais de entrada/saída de dados.

Normalmente o segundo ciclo de entrada / saída de dados poderá habilitar o byte de deslocamento da instrução ser lido pelo SC/MP, para computação ou endereço efetivo, o terceiro ciclo poderá habilitar o conteúdo do acumulador ser memorizado no tal endereço efetivo. Com a saída de endereço da CPU desabilitado, o endereço efetivo associado com a instrução de memorizar é especificado pelo terceiro endereço do DEBUG, fazendo o conteúdo do acumulador ser guardado na posição de memória 77FD do módulo. Os próximos 16 endereços da DEBUG Save Routine são processados de modo idêntico. Primeiro, carregado o início do programa DEBUG (7808) no registro P_3. Segundo, guardar o conteúdo de P_3 na posição 77FB e 77FC. Terceiro guardar o conteúdo do registro status na posição 77FF.

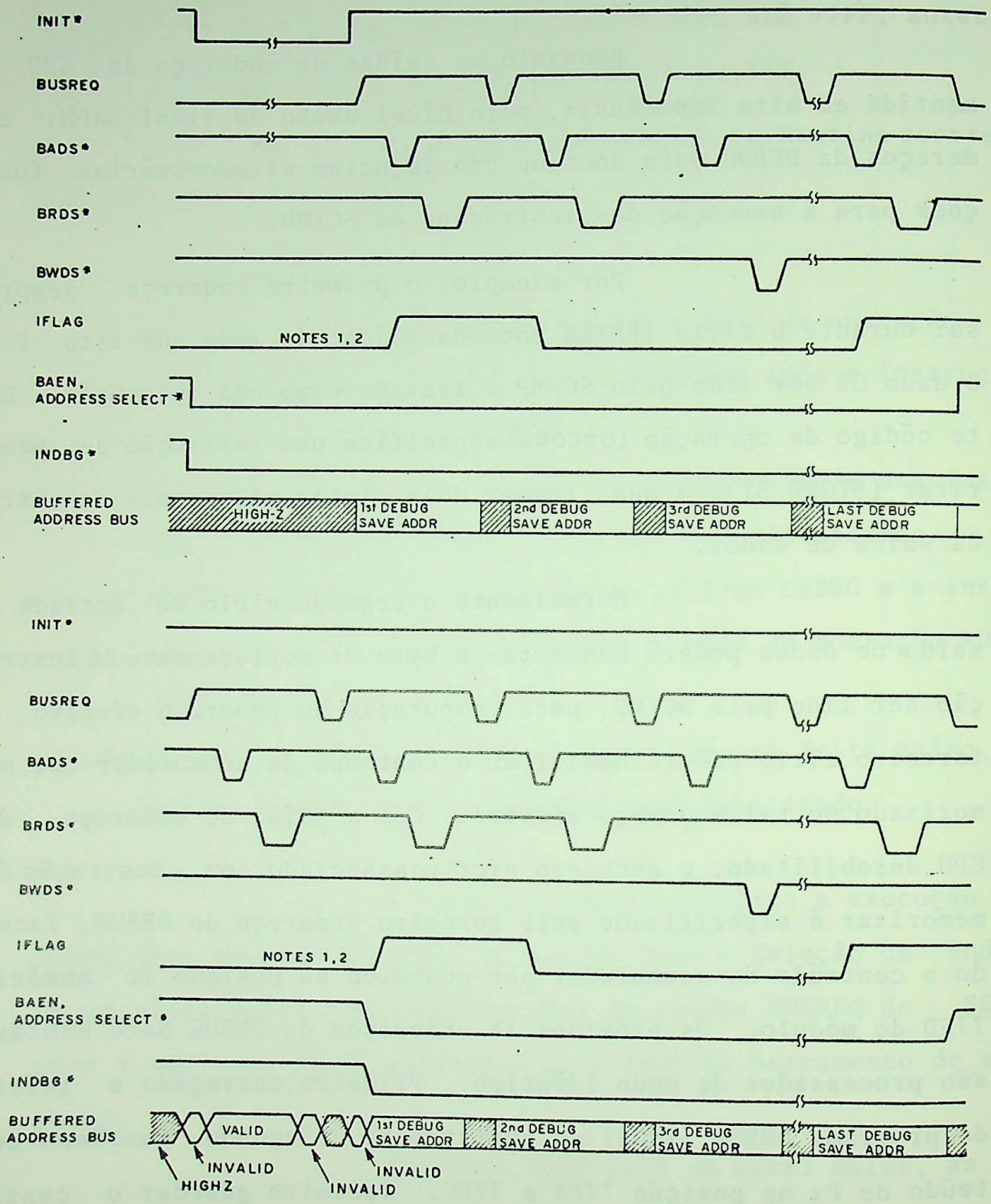

INIT*
BUSREQ
BADS*
BRDS*
BWDS*
IFLAG
NOTES 1, 2
BAEN, ADDRESS SELECT*
INDBG*
BUFFERED ADDRESS BUS
HIGH-Z
1st DEBUG SAVE ADDR
2nd DEBUG SAVE ADDR
3rd DEBUG SAVE ADDR
LAST DEBUG SAVE ADDR
INIT*
BUSREQ
BADS*
BRDS*
BWDS*
IFLAG
NOTES 1, 2
BAEN, ADDRESS SELECT*
INDBG*
BUFFERED ADDRESS BUS
VALID
1st DEBUG SAVE ADDR
2nd DEBUG SAVE ADDR
3rd DEBUG SAVE ADDR
LAST DEBUG SAVE ADDR
INVALID HIGHZ
INVALID
INVALID

FIG. 4

O último DEBUG Save Adress refere-se a instrução XPPC P_3 que troca o conteúdo do PC com o de P_3.

Após a saída do último endereço do DEBUG Save Routine, as saídas BAEN e Seleção de Endereço do controle lógico são colocados no nível alto no edge negativo do sinal BUSREQ para reabilitar as saídas de endereço da CPU do SC/MP. No próximo edge positivo do sinal BUSREQ, a CPU do SC/MP então mostra o endereço ,7809 que vai iniciar a execução do filmware de DEBUG.

A primeira rotina do programa filmware de DEBUG guarda o conteúdo dos registros PC, P_1, P_2 e EX nas posições já designadas; faz a leitura da posição da chave HALT MODE e efetua um salto para a rotina apropriada (teclado ou TTY).

HARDWARE FORÇADO (DEBUG RETURN ROUTINE)

A DEBUG Return Routine é uma subrotina controlada por hardware, executada pelo controle lógico em resposta a instrução HALT que acontece no final da DEBUG Filmware Exit Routine.

Como mostra a figura 5, a execução desta rotina é similar a anterior, isto é, as saídas BAEN e seleção de endereço do controle lógico são mantidas no nível baixo, enquanto a saída BUSREQ da CPU do SC/MP está sendo processada, forçando uma série endereços para fora do barramento de endereços. Um sumário geral das funções da DEBUG Return Routine fica:

1. O valor decrementado do contador de programa que é temporariamente guardado no registro P_3, retorna ao PC, de modo que é, automaticamente incrementado para o próximo valor de início do modo RUN, durante a execução da próxima DEBUG Return Routine.

2. O conteúdo do acumulador é copiado no Registro Status.

3. O Registro P3 e o acumulador são carregados na memória do módulo LCDS para completa reestruturação dos Status internos do SC/MP.

Após a saída do último endereço da DEBUG Return Routine, as saídas BAEN e Seleção de endereço do controle Lógico vão para nível alto no edge positivo do sinal BUSREQ, para reabilitar a saída de endereços da CPU do SC/MP. Então, a saída da CPU do SC/MP restaura o PC no edge positivo do próximo sinal BUSREQ e executa o ciclo IFETCH de entrada/saída de dados para ler o primeiro opcode do modo RUN na locação de memória referenciada.

A execução do programa do usuário continua até que o modo RUN termina como descrito na DEBUG Save Routine.

LÓGICA DE CONTROLE

A lógica de controle providencia o controle das funções de armazenamento dos status internos do SC/MP, - quando o modo DEBUG é selecionado e torna a memorizar quando o modo RUN for selecionado.

A seleção de cada modo faz a saída HALT do habilitador de modo DEBUG/RUN ir para nível alto. Deste modo habilita o SYNC LATCH ser resetado, quando o bit de dado DI5 vai para alto durante o BADS do ciclo IFETCH entrada/saída, que ocorre após a instrução em processo ser executada.

Quando SYNC LATCH é resetado, um edge negativo de Q seta DEBUG Save/Return Routine Enable Latch e rese

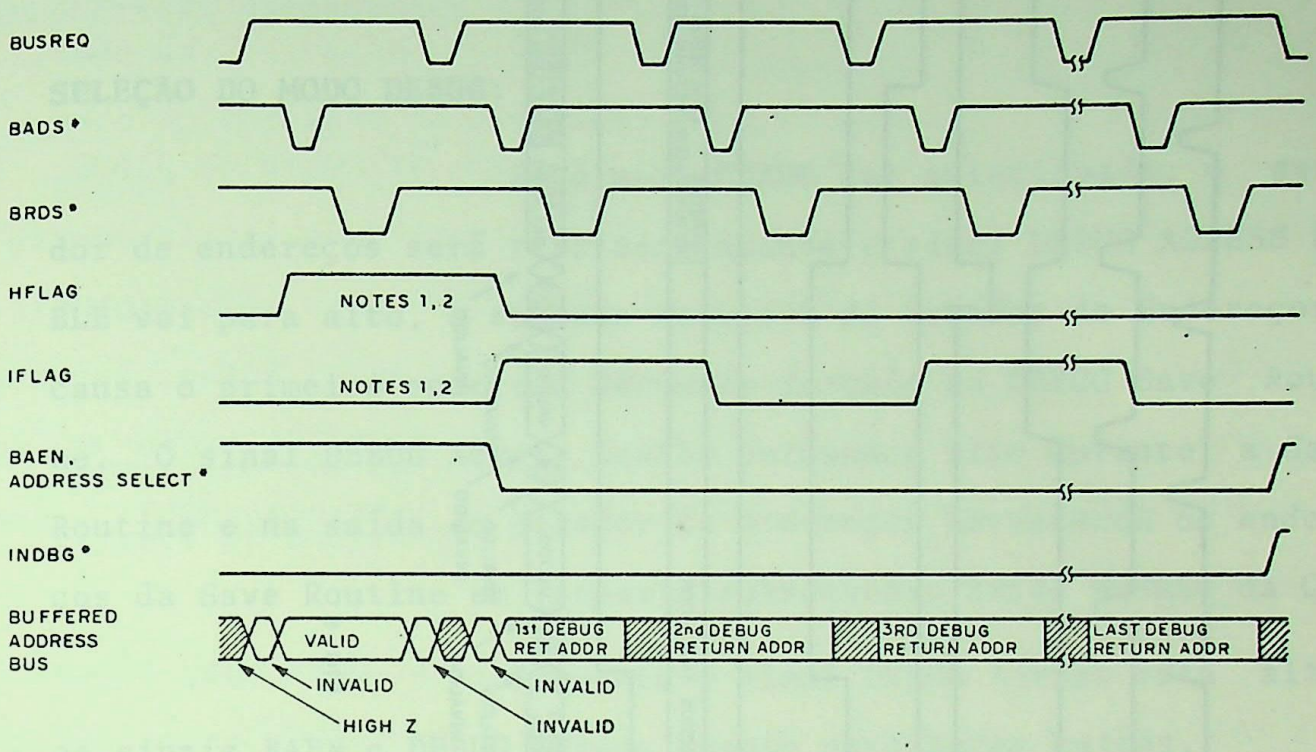

BUSREQ
BADS*
BRDS*
HFLAG
NOTES 1, 2
IFLAG
NOTES 1, 2
BAEN,
ADDRESS SELECT*
INDBG*
BUFFERED
ADDRESS
BUS
VALID
1st DEBUG
RET ADDR
2nd DEBUG
RETURN ADDR
3RD DEBUG
RETURN ADDR
LAST DEBUG
RETURN ADDR
INVALID
HIGH Z
INVALID
INVALID

FIG. 5

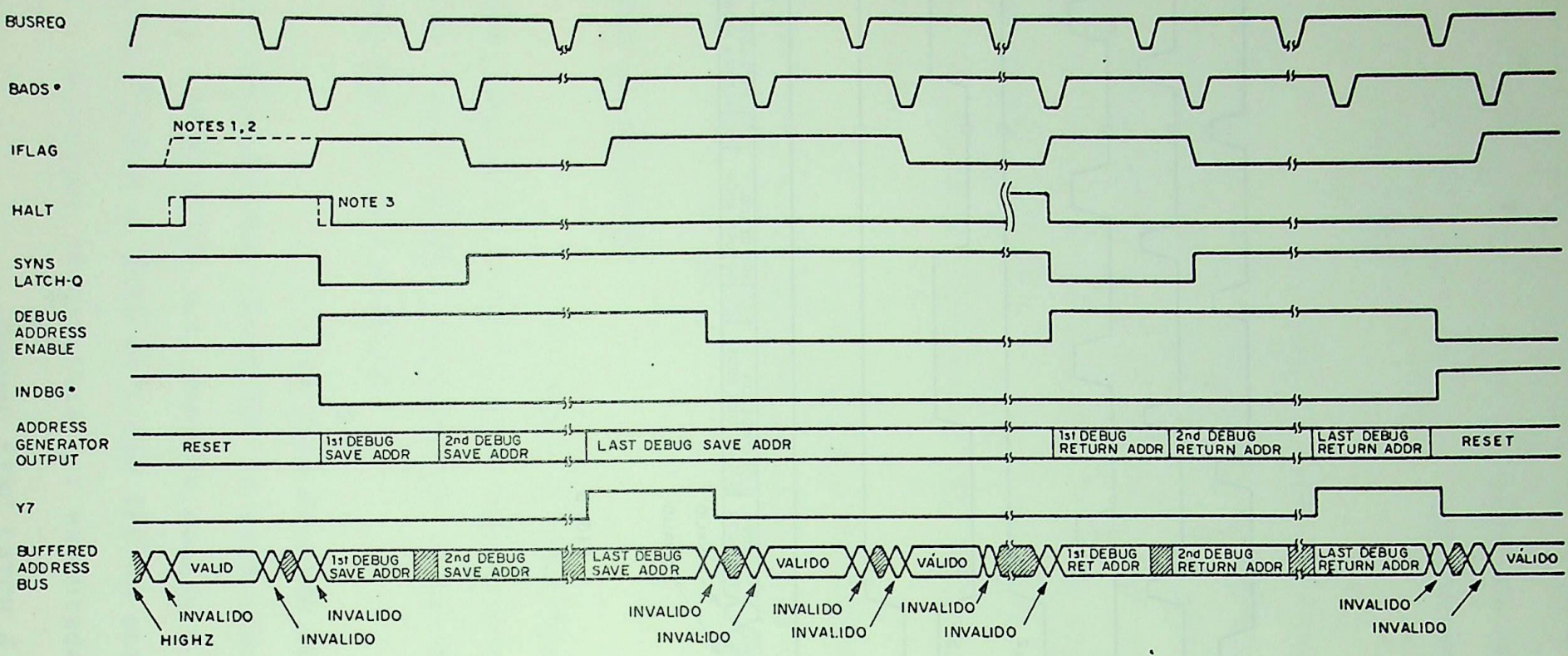

BUSREQ
BADS*
NOTES 1, 2
IFLAG
HALT
NOTE 3
SYNS LATCH-Q
DEBUG ADDRESS ENABLE
INDBG*
ADDRESS GENERATOR OUTPUT
RESET
1st DEBUG SAVE ADDR
2nd DEBUG SAVE ADDR
LAST DEBUG SAVE ADDR
1st DEBUG RETURN ADDR
2nd DEBUG RETURN ADDR
LAST DEBUG RETURN ADDR
RESET
Y7
BUFFERED ADDRESS BUS
VALID
1st DEBUG SAVE ADDR
2nd DEBUG SAVE ADDR
LAST DEBUG SAVE ADDR
VALIDO
VÁLIDO
1st DEBUG RET ADDR
2nd DEBUG RETURN ADDR
LAST DEBUG RETURN ADDR
VÁLIDO
HIGHZ
INVALIDO
INVALIDO
INVALIDO
INVALIDO
INVALIDO
INVALIDO
INVALIDO
INVALIDO
INVALIDO
INVALIDO
INVALIDO

FIG. 6

ta o DEBUG status Latch habilitando o DEBUG Save ou Return Routine. A saída HALT do RUN/DEBUG Mode Enable lógic é então resetada pelo resultado da saída dos latches de DEBUG e SYNC LATCH, é setado no edge positivo do próximo strobe BADS para permitir posteriores operações dos latches de DEBUG controlados pelo Gerador de endereços.

SELEÇÃO DO MODO DEBUG:

Se o modo DEBUG for selecionado, o gerador de endereços será resetado, quando o sinal DEBUG ADRESS ENABLE vai para alto, a entrada de clock do Gerador de Endereços ' causa o primeiro endereço Hardware-forçado da DEBUG Save Routine. O sinal DEBUG Adress Enable permanece alto durante a Save Routine e na saída do Gerador de endereços permanecem os endereços da Save Routine em resposta subsequente saída BUSREQ da CPU.

Enquanto o sinal DEBUG Adress está alto, os sinais BAEN e DEBUG Adress Enable permanecem baixos.

A saída de endereços da CPU é mantida em alta-impedância durante o Hardware, fôrçado da DEBUG Save Routine e a entrada BUSREQ para os Buffers de Endereços habilitam os endereços da Save Routine serem colocados nos Buffers e barramento do módulo.

Quando os endereços da Save Routine saem, o controle Y7 vai para alto. Deste modo,habilita os latchs da DEBUG Save/Return Routine serem resetados no edge negativo do sinal BUSREQ. Após os latches da DEBUG Save/Return Routine serem resetados, o sinal DEBUB Adress Enable baixo inibe os clocks do Gerador de Endereços, o sinal BAEN alto reabilita a saída de en

dereços da CPU e o sinal alto DEBUG Adress Enable habilita os endereços da CPU serem colocados no barramento.

SELEÇÃO DO MODO RUN:

Após o latch de status do DEBUG ser resetado no início da DEBUG Save Routine, ele permanece resetado até ser selecionado o modo RUN.

Enquanto o programa filmware de DEBUG é executado, o sinal baixo STATUS INDBG é aplicado continuamente na lógica de habilitação de modo RUN/DEBUG para inibir o circuito, exceto em resposta a entrada DI7 (HFLAG). Então, durante a execução do filmware de DEBUG, a lógica de habilitação do modo RUN/DEBUG somente detecta a execução da instrução HALT que ocorre após o modo RUN ser selecionado.

Quando DI7 (HFLAG) vai para alto durante a execução da instrução HALT do filmware de DEBUG, as saídas da lógica de habilitação do modo RUN/DEBUG apresentam um sinal HALT alto e seta o SYNC e Latches de DEBUG Save/Return Routine.

As saídas seguintes dos latches de DEBUG Save/Return Routine então resetam o sinal HALT e habilitam a execução do Hardware forçado do DEBUG Return Routine como descrito previamente.

Isto é o sinal alto do DEBUG Adress Enable habilita o Gerador de Endereços, o sinal baixo BAEN desabilita as saídas de endereços da CPU e o sinal baixo DEBUG Adress Enable habilita os Buffers de Endereços alimentar com endereços da Rotina Return o barramento do módulo, enquanto BUSREQ é baixo . Quando sai o último endereço da Return Routine, a saída Y7 do Ge

rador de Endereços vai para alto e os latches da DEBUG Save/Return Routine são resetados no edge negativo do sinal BUSREQ. Ao mesmo tempo, o Gerador de Endereços provê um pulso de CLEAR que reseta o contador de Endereços interno e seta o Latch de Status de DEBUG.

Então o Gerador de Endereços volta ao estado de reset, o modo RUN inicia na rememorização do endereço do contador de programas no edge positivo do próximo sinal BUSREQ.

INÍCIO:

Se o sinal INIT é setado para baixo, isto serve como uma entrada assíncrona de Reset para o Latch SYNC e Gerador de Endereços.

Então, quando o sinal INIT volta para alto, a DEBUG Save Routine é executada.

LÓGICA DE DECODIFICAÇÃO DE ENDEREÇO:

Esta lógica continuamente monitora os bits de endereços de A12 até A15 para fazer a seleção da Memória - (MENSEL) quando a memória do módulo, teclado ou Display e endereçado (endereço 7000 até 7FFF). Enquanto o sinal seleção de Memória é baixo, o Decodificador de Endereço de Memória é habilitado e os bits de endereço de A9 até A11 são decodificados para fazer a habilitação do circuito apropriado.

READ ONLY MEMORY:

A memória ROM do módulo consiste de 512 por 8 bits, palavras que funcionam sobre controle nas entradas de habilitação da ROM.

Então quando a ROM é endereçada, o sinal apropriado vai para baixo, de modo que as saídas do ROM apresentam o endereço especificado pelo bits de endereço A0 até A8.

READ/WRITE MEMORY: de dois integrados

A memória RAM do módulo consiste de 256 por 4 bits palavras. Quando o sinal de habilitação da RAM é baixo e o bit de endereço A8 é alto, a memória é habilitada e lê ou escreve os dados que ocorrem sob controle da entrada WRITE STROBE.

Em um ciclo de entrada/saída de dados a leitura é feita com WRITE STROBE em alto e a escrita em baixo. A faixa de endereços da RAM é de X'7700 até X'77FF, as localizações usadas pelo filmware de DEBUG são X'7774 até X' 777F e X'77D8 até X'77FF.

BUFFERS DE DADOS

Os buffers de dados são bidirecionais que funcionam sob controle das entradas MENSEL e READ STROBE.

Quando ambas MEMSEL e READ STROBE são baixas, os buffers de dados são direcionados para alimentar dados do DEBUG Data Out BUS para Buffered Data Bus.

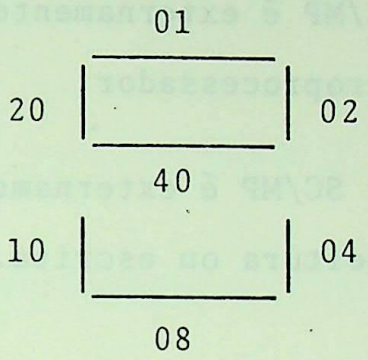
A todo tempo, os buffers de dados continuamente permitem passagem de dados do Buffered Data Bus para o DEBUG Data In Bus.

DISPLAY

O display consiste de seis indicadores hexadecimal que podem ser endereçados individualmente ou em paralelo. A tabela seguinte apresenta os dígitos e os respectivos endereços:

DIGITO	ENDEREÇO
1º (mais a esquerda)	7020
2º	7010
3º	7008
4º	7004
5º	7002
6º (mais a direita)	7001

SEGMENTOS



VALOR HEXADECIMAL:

0 - 3F	5 - 6D	A - 77	F - 71
1 - 06	6 - 7D	B - 7C	
2 - 5B	7 - 07	C - 39	
3 - 4F	8 - 7F	D - 5E	
4 - 66	9 - 67	E - 79	

O endereçamento do display ocorre quando as entradas PANEL ENABLE e WRITE STROBE são baixos (durante a instrução de memorizar ao endereço X'7000 até X'71FF), fazendo os bits de endereço A0 até A5 ser colocado nos latches de Digit Select, e os bits de dados DI 0 até DI 6 nos latches Digit Display.

Latches de endereço e de dado são continuamente aplicados no display, até que nova informação é envidada aos latches ou os latches são resetados via sinal Panel clear.

Ocorre o reset via sinal Panel clear sob as condições:

1. O sinal INIT é feito baixo para dar início ao microprocessador, causando uma entrada NRST baixa para ser aplicada ao display.

2. A entrada RUN da CPU do SC/MP é externamente feita baixa suspendendo a operação do microprocessador.

3. A entrada MEMROY da CPU do SC/MP é externamente feito baixa' para estender o ciclo de leitura ou escrita.

TECLADO:

Consiste de 18 teclas que são habilitadas pelo programa de controle via entrada de endereço de A0 até A4.

Então quando qualquer destes bits é alto, uma chave no correspondente ramo produz uma saída a ser aplicada nos Buffers do teclado. Estes buffers permanecem desabilitados até que ocorra um nível baixo nas entradas Panel Enable e Read Strobe, durante o ciclo de leitura de dados de entrada/saída, carregado do endereço 7000 até 71FF.

Quando os sinais Panel Enable e Read strobe são baixos, os buffers investem o dado na entrada IN e colocam este dado no DEBUG Data Out Bus.

CHAVE HALT MODE:

Além das entradas recebidas das teclas os buffers

recebem informação da chave HALT MODE.

Esta entrada é periodicamente testada pelo firmware de DEBUG.

CAPÍTULO V

DESENVOLVIMENTO DO CIRCUITO INTERFACE

PARA PROGRAMAÇÃO

DESENVOLVIMENTO DO CIRCUITO DE INTERFACE PARA PROGRAMAÇÃO:

Já estudadas as características do Microprocessador a ser usado e das memórias a serem programadas, passamos a elaboração do circuito de INTERFACE para programação. Posicionado nos conhecimentos adquiridos sobre o microprocessador SC/MP, decidimos optar pela capacidade de operação em Série deste microprocessador para o intercâmbio de informações com o circuito de programação. Este procedimento, mais tarde, nos apresentou vantagens adicionais sobre o circuito projetado que não levamos em conta no início do trabalho.

O problema então se resumia em desenvolver um circuito para apresentar os dados a serem programados na memória dentro dos restritos requisitos de temporização, bem como o controle dos sinais de comando de programação.

Este procedimento deveria ser efetuado pelo microprocessador através de uma programação adequada. Deste ponto tiramos que os dados a serem programados já deveriam estar armazenados no SC/MP.

O diagrama em blocos da Fig. 1 ilustra a nossa possível solução dó problema. Devido as características elétricas dos sinais no barramento do SC/MP, a opção para a construção do circuito de INTERFACE foi sobre a tecnologia TTL.

Todos os sinais no barramento e as diversas portas de entrada e saída do SC/MP possuem os níveis TTL de tensão, isto é, +5V para o nível lógico um e zero volt (ou seja 0,2V) para o nível lógico zero.

Como no caso da Memória semicondutora EPROM-2708, ainda existe a necessidade de outros níveis de tensão, como no

caso do 26 V para o pulso de programação (PGM) e de 12V para habilitar a programação do chip, houve a necessidade de serem adicionados dois circuitos auxiliares para prover estes comandos (como pode ser consultado no capítulo referente ao estudo da memória 2708)

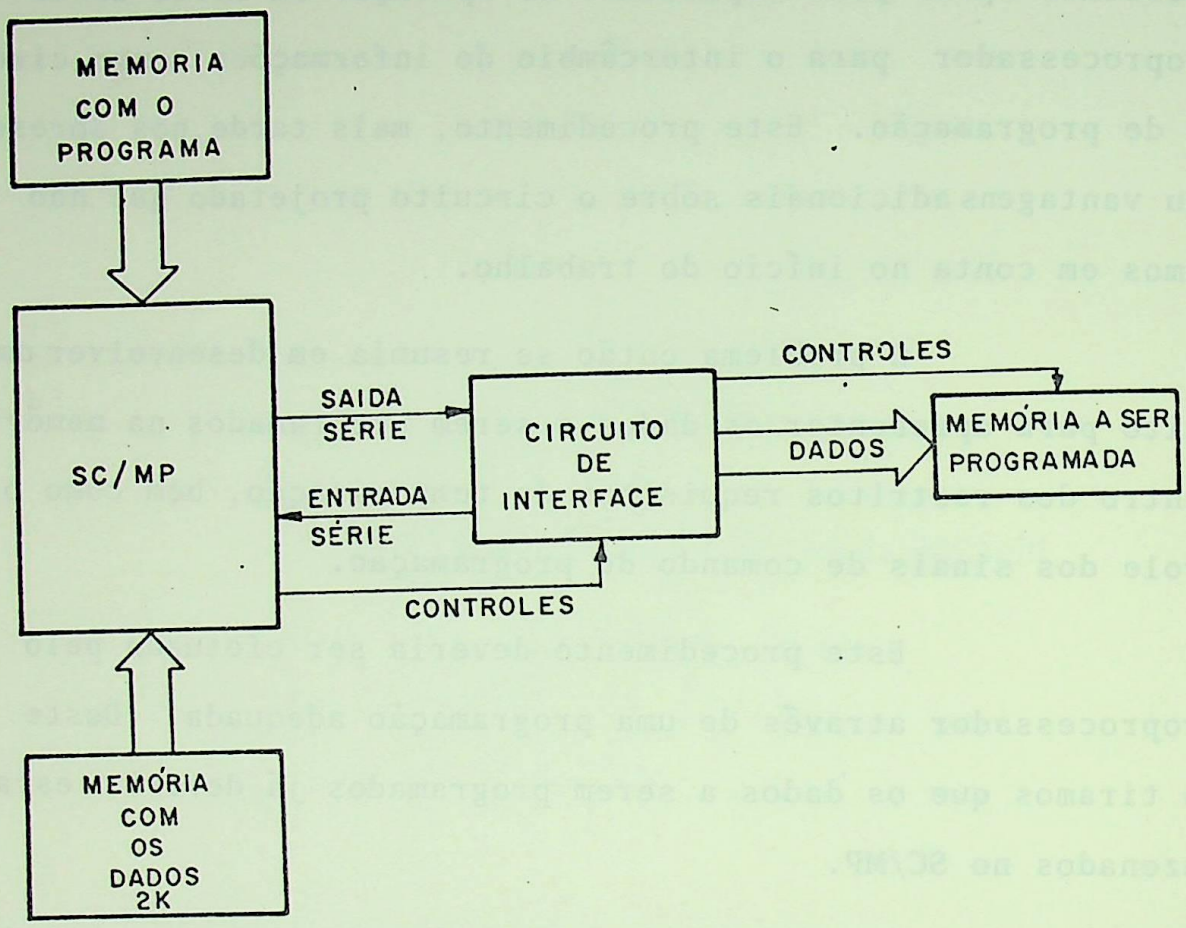


FIG.1

SINAIS DE COMANDO DO µP SC/MP.

Este programador de memórias, como já foi mencionado, usa as capacidades de comunicação série do microprocessador SC/MP. Deste modo fizemos uso dos dois registros de troca série, a saída SOUT e a entrada SIN.

Estas duas opções estão ligadas a um registro de 8 bits, que é o registro EXTENSÃO (extension). Em adição, o registro SOUT possui um LATCH para garantir o sinal de saída.

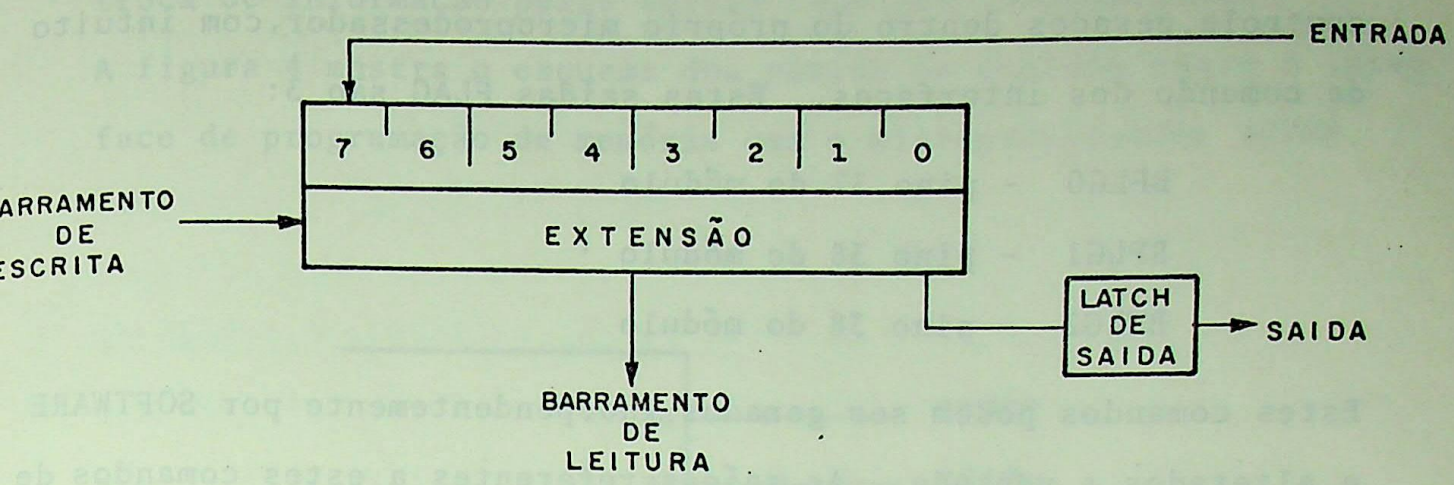


FIG. 2

Como pode ser observado pela Fig. 2, o primeiro bit a deixar o registro EXTENSÃO ao comando da instrução SIO (serial imput/output) do microprocessador é o bit 0 (zero) e o último é o bit 7 (sete); bem como o primeiro a entrar no registro EXTENSÃO é o bit 0 (zero) e o último, o bit 7.

A entrada é também comandada pela instrução SIO (código de operação 19).

Esta observação é fundamental para a elaboração do circuito de recepção dos dados a serem gravados na memória EPROM.

Estas duas saídas são disponíveis nos pinos:

Saída série BSOUT - pino 34

Entrada série BSIN - pino 33

do módulo SC/MP.

A fim de evitar erros na sincronização das transferências de dados, optamos por usar o microprocessador na geração de todos os sinais de comando.

Para isto, usamos de outra capacidade do microprocessador SC/MP,

que são as saídas de FLAG. As saídas de FLAG são sinais de controle, gerados dentro do próprio microprocessador, com intuito de comando dos interfaces. Estas saídas FLAG são 3:

BFLG0 - pino 37 do módulo

BFLG1 - pino 38 do módulo

BFLG2 - pino 38 do módulo

Estes comandos podem ser gerados independentemente por SOFTWARE e alterados a vontade. As saídas referentes a estes comandos de FLAG possuem BUFFERS, isto é, possuem um eforçador a fim de poder alimentar cargas relativamente significativas.

As saídas de FLAG estão associadas ao registro denominado STATUS, já mencionado anteriormente. A maneira como existe este relacionamento está mostrada na figura 3.

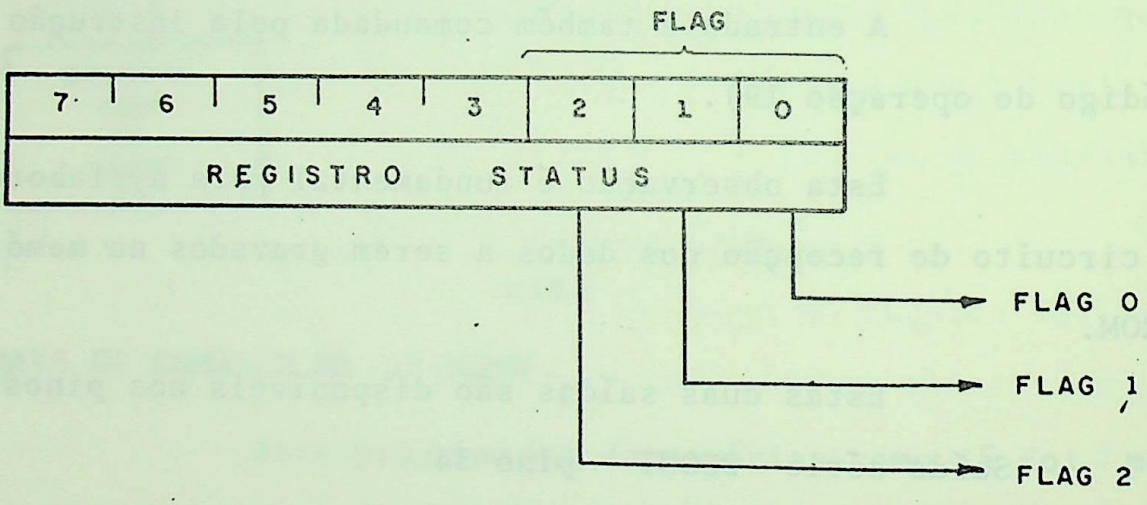


FIG-3

A natureza das demais posições do registro já foi tratada num estudo anteriormente e não vem ao caso para nós no momento.

Disponíveis estes comandos e a entrada e saída sé

rie, passamos agora a distribuição das obrigações de comando e troca de informação pelas várias conexões disponíveis.
A figura 4 mostra o esquema dos pontos de contato entre o interface de programação de memória com o microprocessador SC/MP.

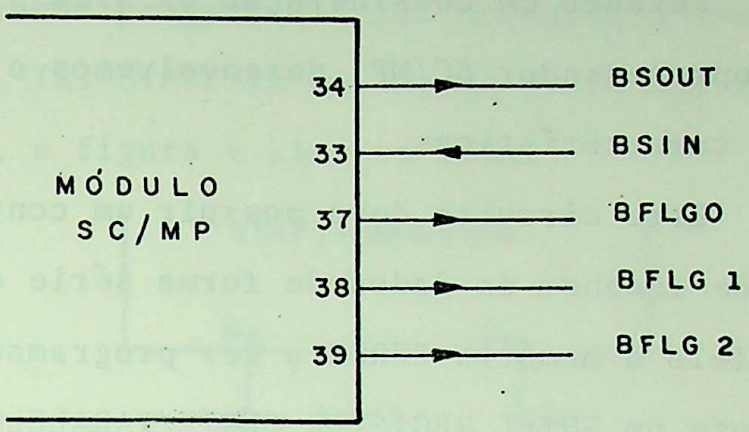


FIG.4

Com referência às saídas de FLAG, foram distribuídas da seguinte maneira:

BFLG0 - ficou responsável pelo clock base do sistema, isto é, a temporização que gera os endereços e desloca os dados de modo série para os shift registers de conversão série-paralelo e paralelo - série.

BFLG1 - ficou responsável pelo sinal de zeramento dos contadores de endereço.

BFLG2 - ficou responsável pelo pulso de programação da memória, associado com uma chave que seleciona a programação ou verificação do programa gravado.
Este sinal também atua no comando LOAD (carga) do shift register de carregamento paralelo e saída série.

Desta maneira distribuídos os diversos controles podemos passar a elaboração do circuito de INTERFACE propriamente dito.

O CIRCUITO DE INTERFACE

Levando em consideração os sinais e controles usados do microprocessador SC/MP, desenvolvemos o circuito com suas diversas características.

Este circuito deve possuir um conversor série/paralelo, a fim de receber os dados de forma série e apresentá-los na forma paralelo a memória EPROM a ser programada. Este circuito é basicamente um SHIFT REGISTER SÉRIE/PARALELO, que é mostrado esquematicamente na figura 5.

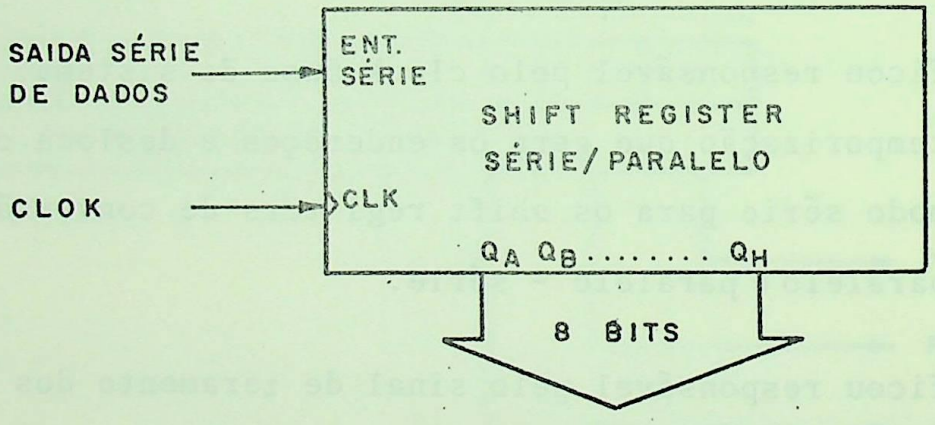


FIG.5

Deste modo os dados são deslocados em série, vindo do Registro Extensão do SC/MP, e vão sendo introduzidos neste SHIFT REGISTER a cada pulso de clock.

No oitavo pulso do clock está garantido que o

dado completo (8 bits) que anteriormente estava armazenado na memória RAM do SC/MP, se encontra agora disponível nas saídas deste shift.

Devido ao fato das memórias EPROM receberem os dados para programação pelas mesmas linhas de saída de dados, e no nosso caso vai existir o interesse em fazer também a leitura de memória recém programada, foi necessário isolar através de dispositivo TRI-STATE as saídas do shift com as entradas da memória EPROM, a figura 6 ilustra isto.

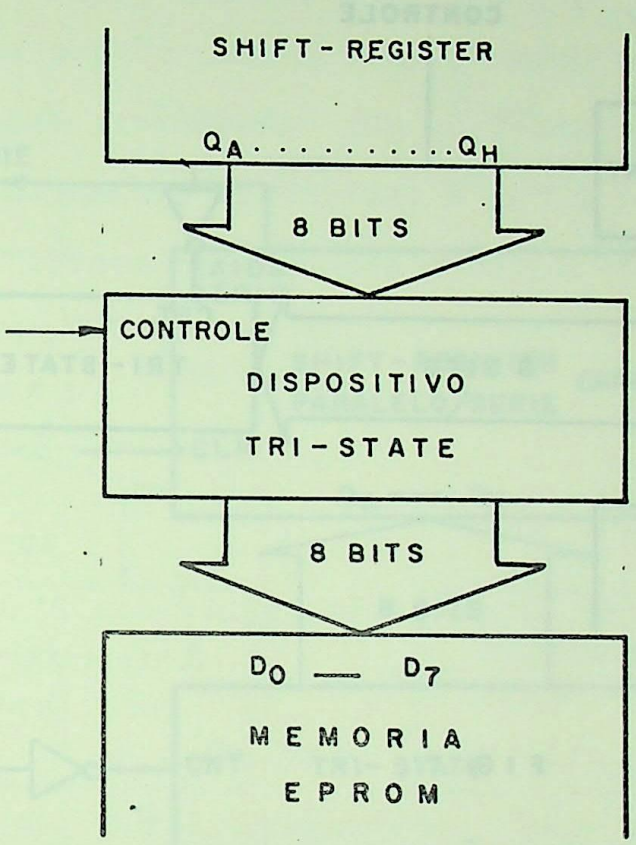


FIG. 6

Como já mencionamos, iremos fazer também a leitura da memória recém programada. Para isto é necessário captar os dados na saída da mesma, o que foi feito usando também os componentes TRIS-STATE.

Estes dispositivos foram arranjos de tal modo que, por um determinado controle, é habilitada a escrita na memória e desabilitada a leitura. Na troca de comando acontece o

inverso. A figura 7 ilustra o texto.

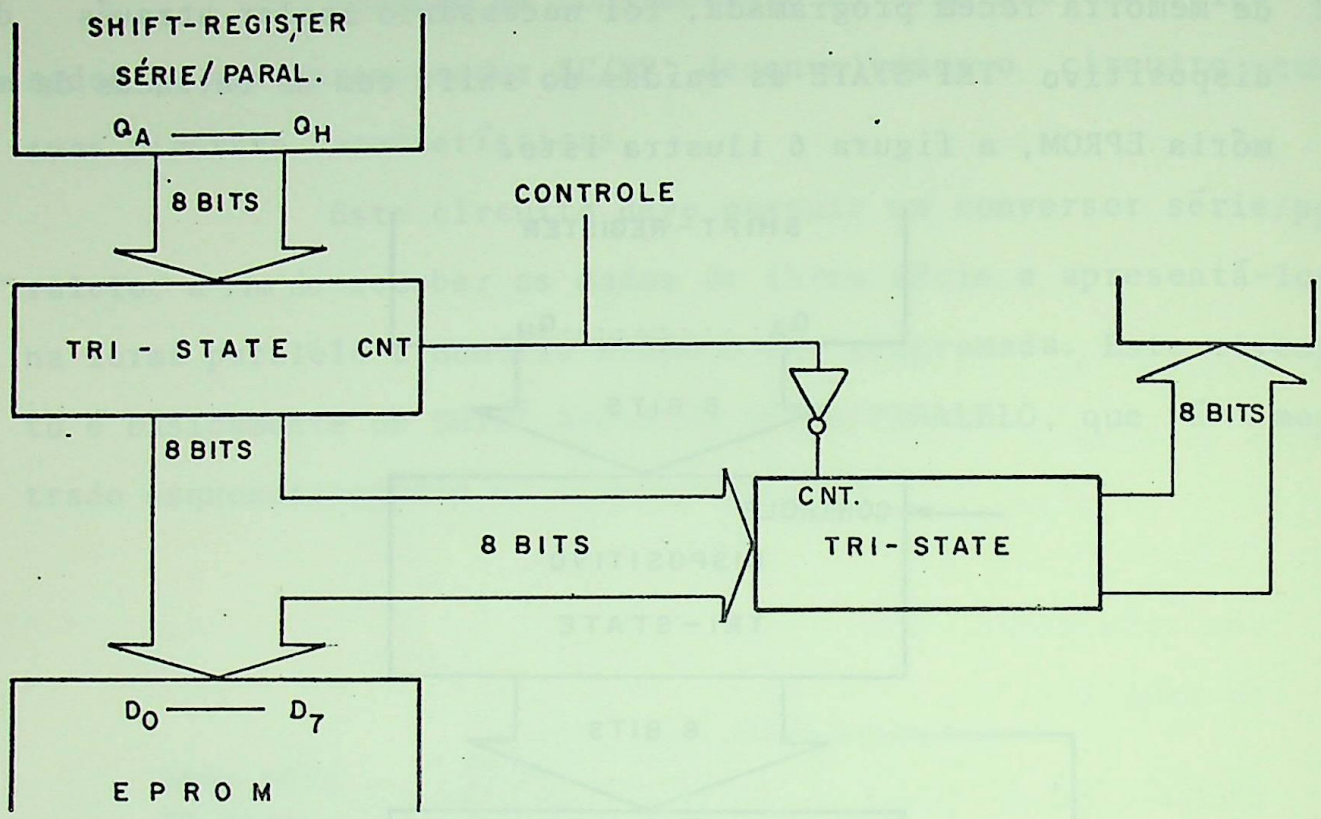


FIG. 7

Até o presente momento está desenvolvida toda a parte de tratamento dos dados vindos do microprocessador SC/MP. Porém, se desejarmos também fazer a leitura da memória, o intuito é testar a programação, isto é, detectar possíveis erros.

Deste modo, os dados gravados na memória devem ser devolvidos ao SC/MP e comparados com a matriz armazenada em suas memórias RAM. Por isso foi usado o mesmo registro Extensão

só que agora no Interface devemos fazer o procedimento inverso do que até agora apresentamos, isto é, captar os dados da memória EPROM de forma paralela, e entregá-los ao SC/MP de forma série.

Para isto foi necessário um conversor PARALELO-SÉRIE, que é também um SHIFT-REGISTER, porém, agora, de carregamento paralelo, deslocamento série. A figura 8 mostra isto de forma esquemática.

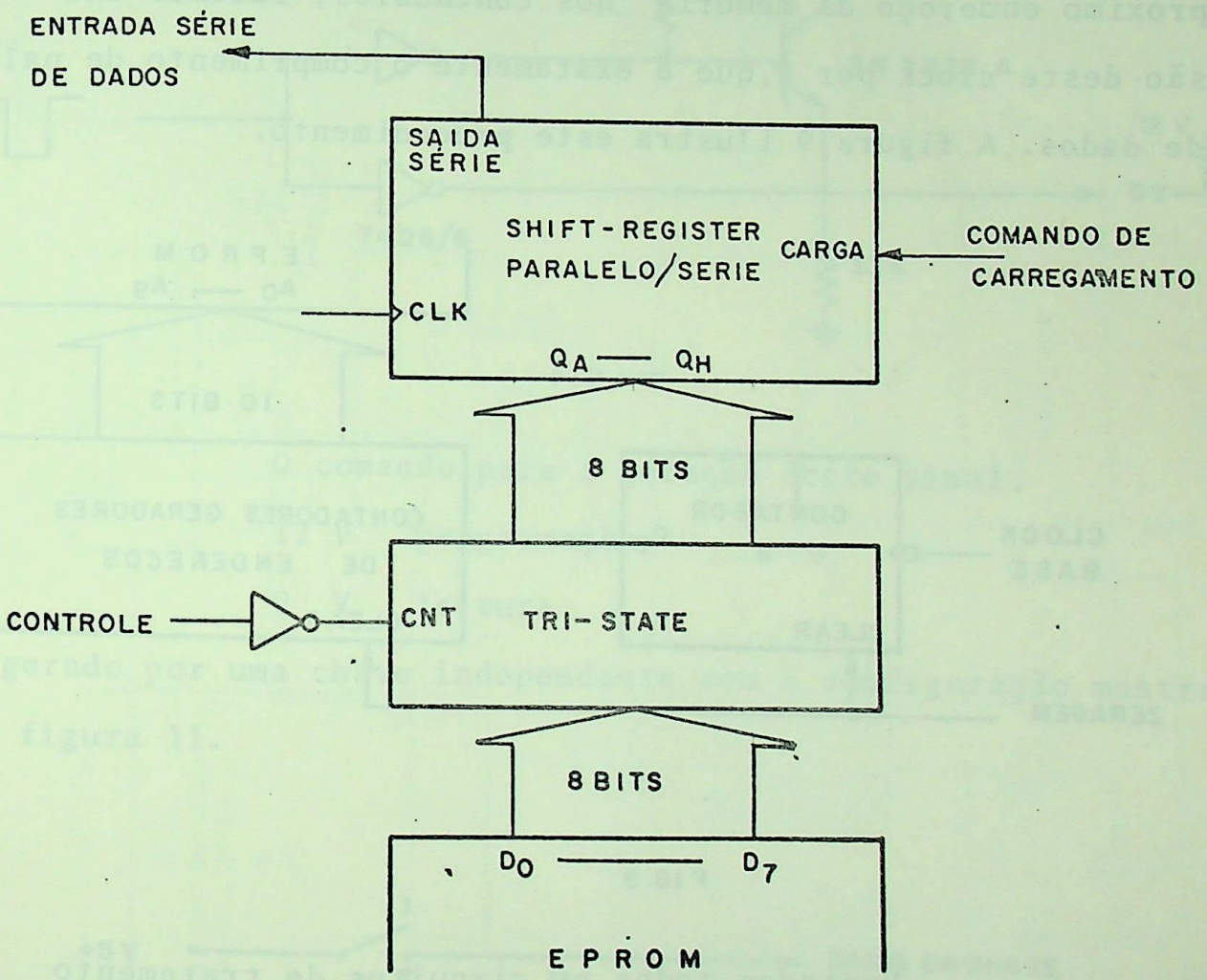


FIG. 8

Pronta a parte da apresentação dos dados a memória e retiragem dos dados da mesma, passamos a parte de geração dos endereços.

A geração de endereços da memória EPROM é naturalmente feita através de contadores binários acoplados em cascata. A única coisa que apareceem adição a este processo de geração de endereço é a forma de sincronismo usada.

A fim de conseguir perfeito sincronismo entre o deslocamento série dos dados e a geração de endereços, usamos o mesmo clock para as duas funções,porém, só que antes de gerar o próximo endereço de memória nos contadores, fazemos uma divisão deste clock por 8,que é exatamente o comprimento de palavra de dados. A figura 9 ilustra este procedimento.

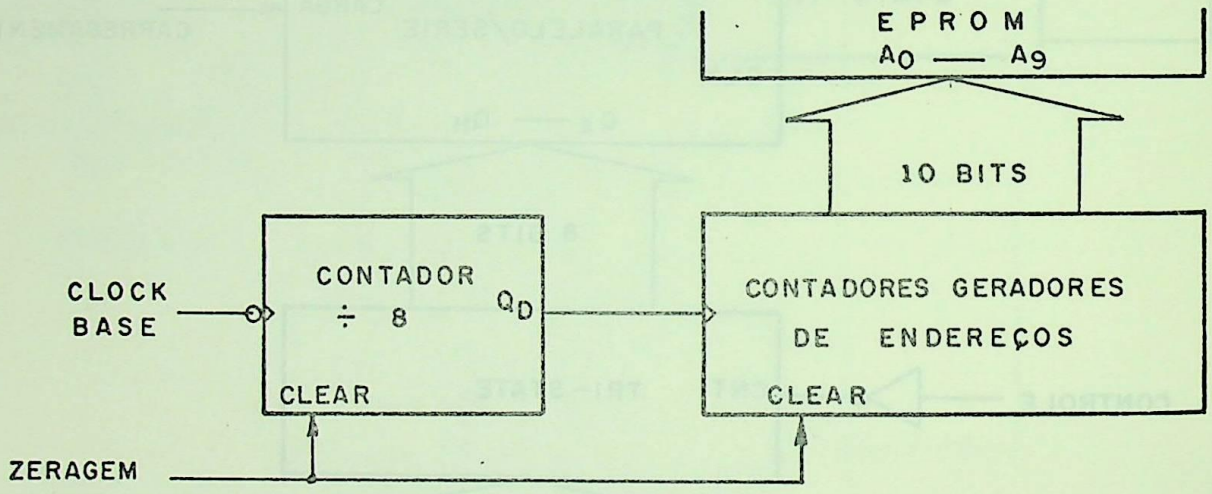


FIG.9

Montados todos os circuitos de tratamento dos dados e geração de endereços, passamos aos circuitos de controle de geração do pulso de programação da memória EPROM.

Como já descrito para a memória EPROM 2708, esta necessita de um pulso de controle que indica o modo de programação ou de leitura. Para isto foi necessário o desenvolvimento

de um circuito especial, visto que os níveis lógicos deste controle não coincidiam com os níveis lógicos TTL normal. Isto é, seria 0 volts para nível lógico zero e 12 volts para o nível lógico 1.

Assim através de um sinal com níveis TTL normal, ativamos estes níveis de 0 e 12 volts através do circuito mostrado na figura 10. Este circuito foi simplificado em relação ao dado pelo manual, porém, a performance é a mesma.

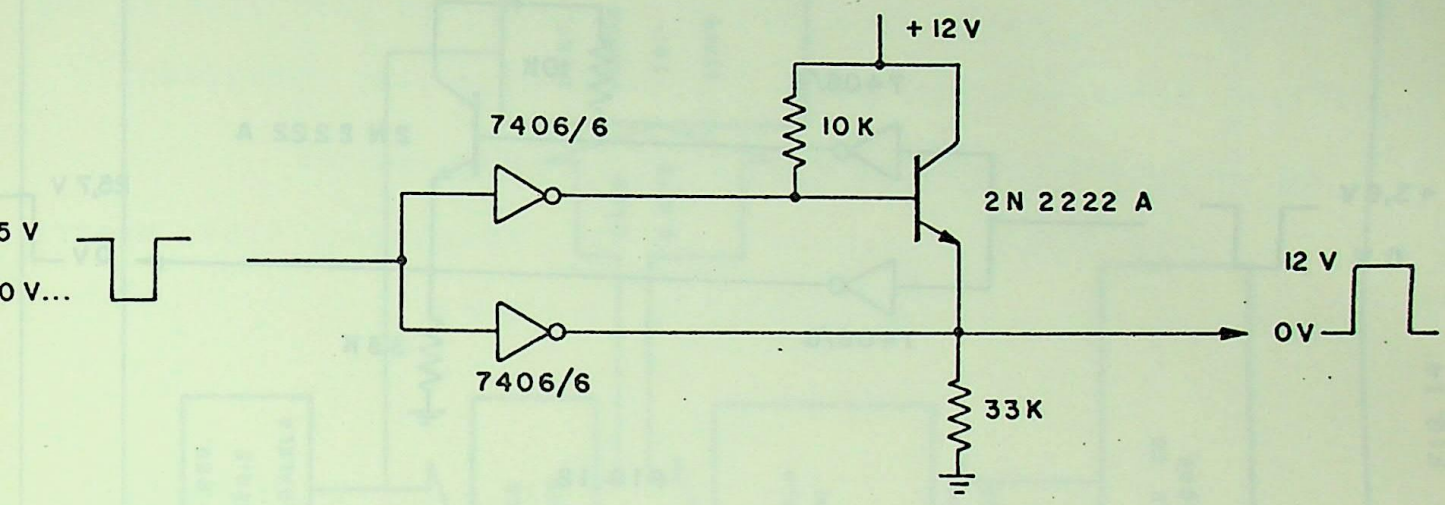


FIG.-10

O comando para a geração deste sinal,

12 V programação

0 V leitura

é gerado por uma chave independente com a configuração mostrada na figura 11.

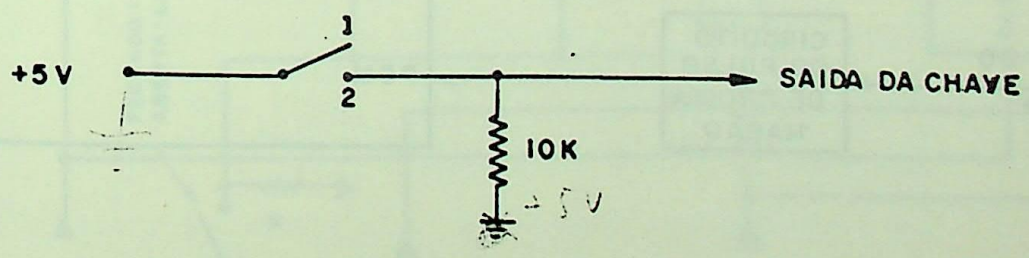


FIG.11

Na posição 1 temos o modo de programação e na posição 2, o de leitura.

O pulso de programação deve também ser gerado por um circuito semelhante, somente que, agora, o sinal de comando do circuito vem do próprio microprocessador SC/MP, com a devida temporização necessária a programação da memória específica.

O circuito do pulso fica (figura 12)

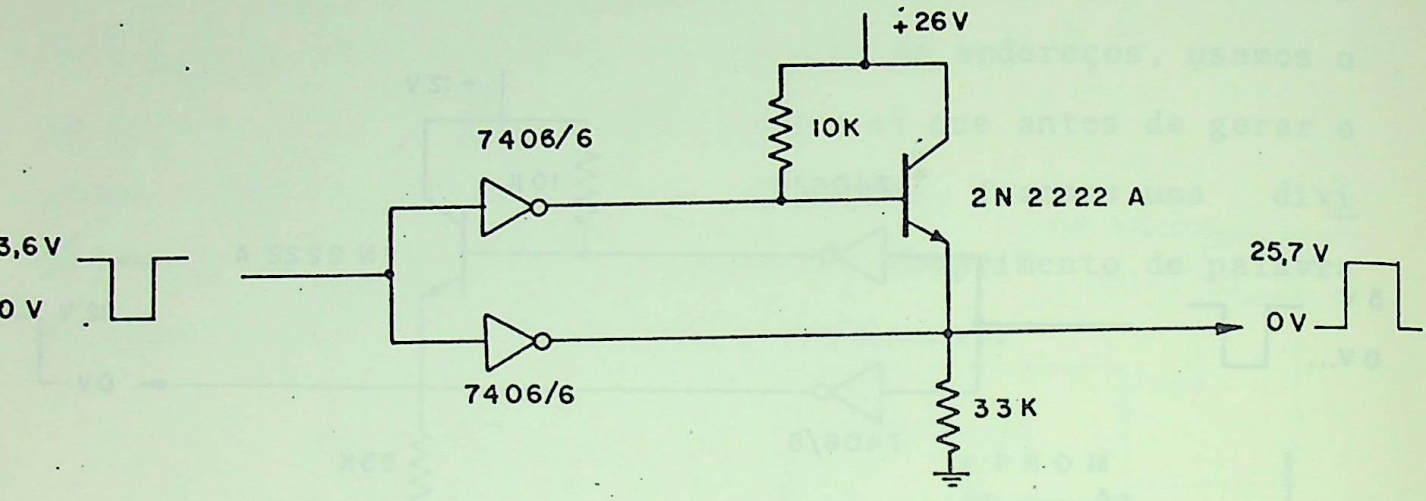


FIG. 12

A ligação destes circuitos com a memória está mostrada na figura 13.

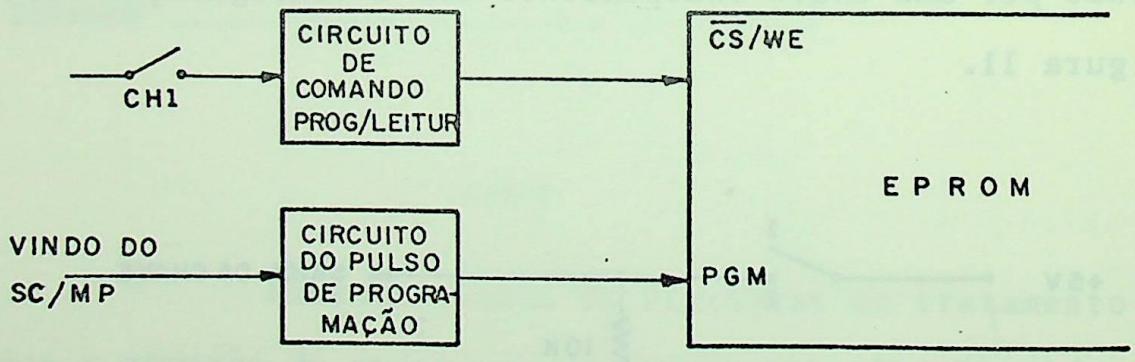


FIG. 13

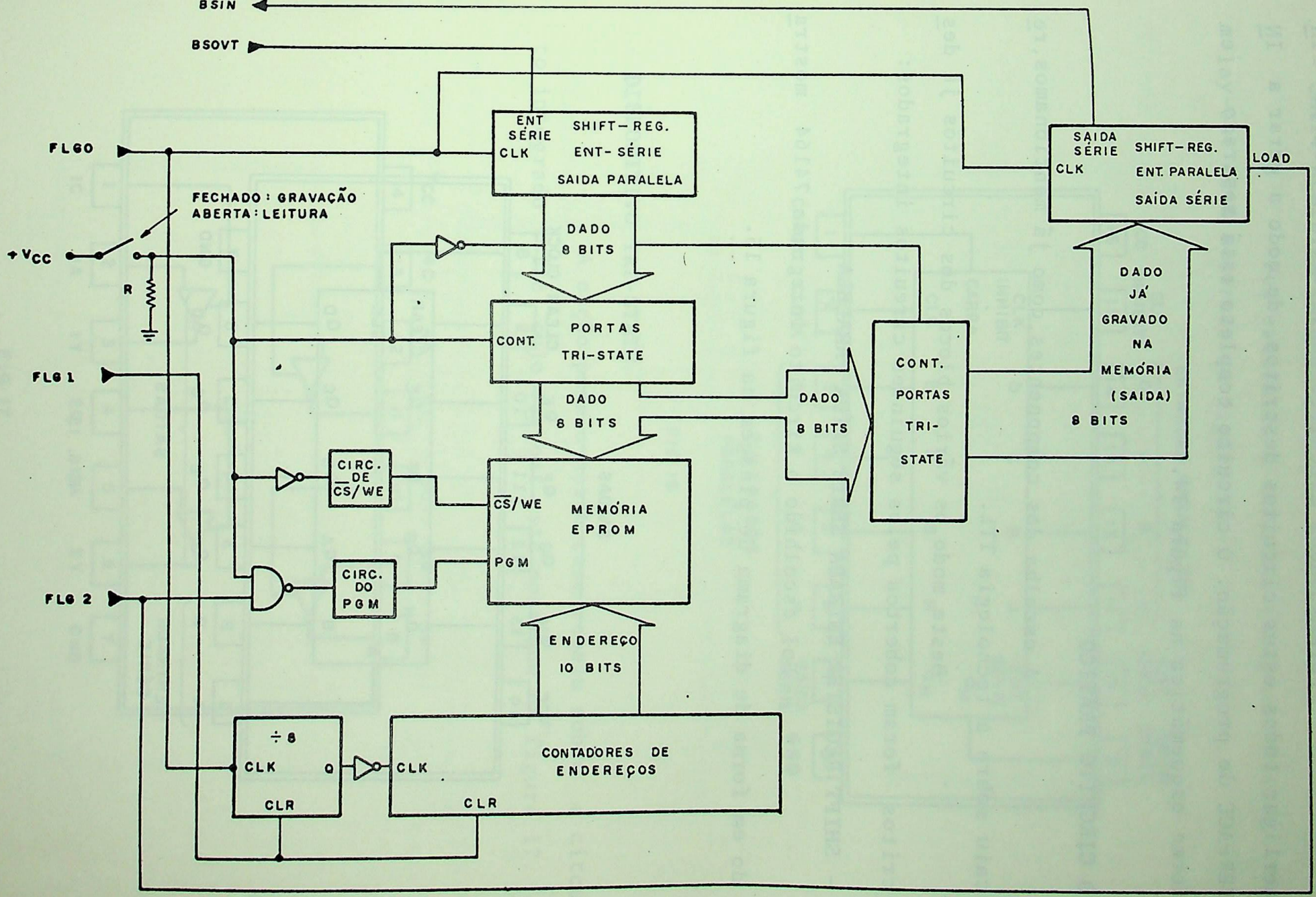

BSIN
BSOVT
FLG0
FECHADO : GRAVAÇÃO
ABERTA : LEITURA
+Vcc
R
FLG 1
FLG 2
ENT SÉRIE
CLK
SHIFT – REG.
ENT – SÉRIE
SAIDA PARALELA
DADO
8 BITS
CONT.
PORTAS
TRI-STATE
CIRC. DE CS/WE
CIRC. DO PGM
CS/WE
PGM
MEMÓRIA
E PROM
ENDEREÇO
10 BITS
÷ 8
CLK
Q
CLR
CONTADORES DE
ENDEREÇOS
DADO
8 BITS
CONT.
PORTAS
TRI-
STATE
SAIDA
SÉRIE
CLK
SHIFT – REG.
ENT. PARALELA
SAÍDA SÉRIE
LOAD
DADO
JÁ
GRAVADO
NA
MEMÓRIA
(SAIDA)
8 BITS

FIG. 14

Foram usados circuitos auxiliares, a fim de interligar todos estes circuitos descritos, de modo a gerar a INTERFACE de programação. O circuito completo está mostrado em forma esquemática na figura 14.

O CIRCUITO PRÁTICO

A escolha dos componentes, como já mencionamos, recaiu sobre a tecnologia TTL.

Deste modo, os vários blocos dos circuitos já descritos foram cobertos pelos seguintes circuitos integrados:

- SHIFT REGISTER ENTRADA SÉRIE SAÍDA PARALELA

Foi escolhido o circuito integrado 74164 mostrado em forma de diagrama de pinagem na figura 15.

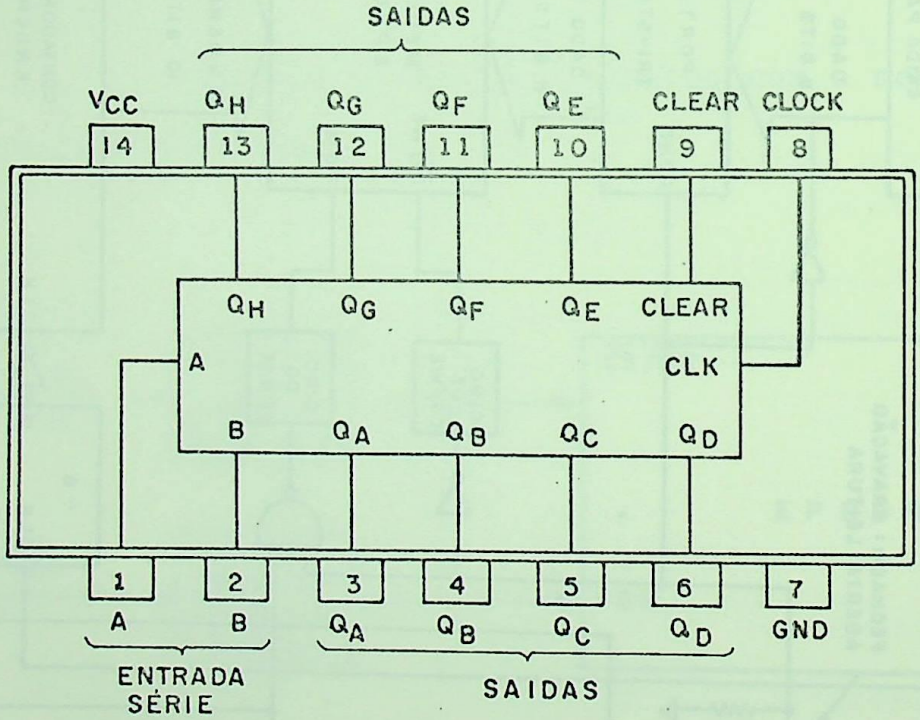


FIG. 15

- SHIFT REGISTER ENTRADA PARALELA SAÍDA SÉRIE

Foi escolhido o circuito integrado 74165, cujo lay-out está mostrado na figura 16.

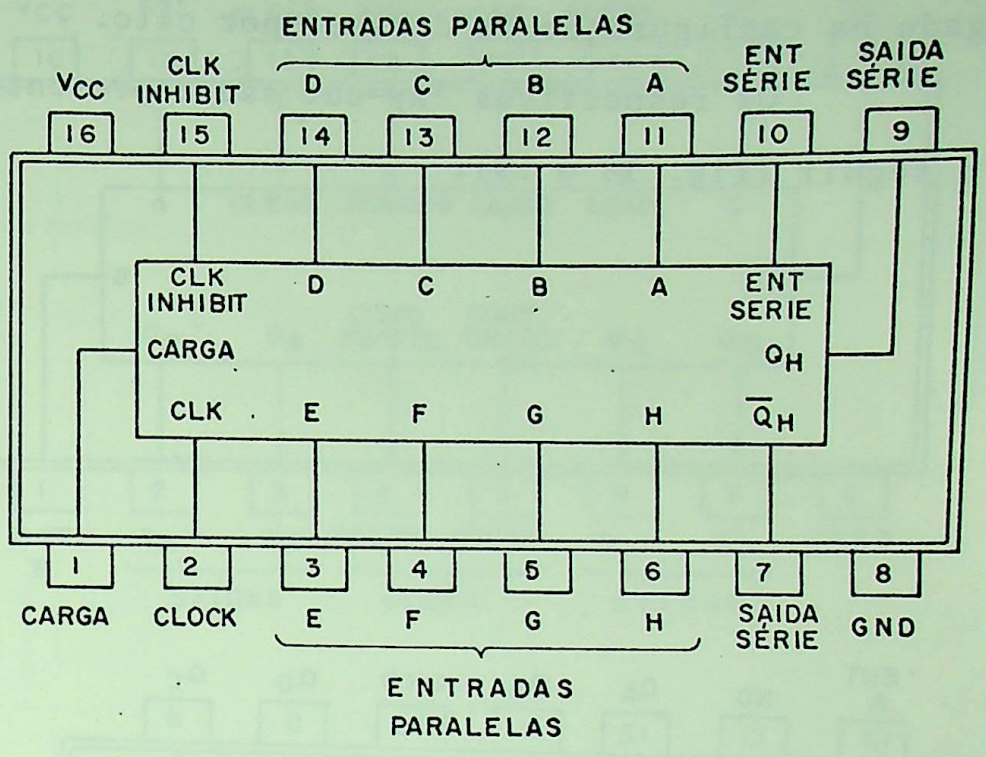


FIG. 16

- DISPOSITIVO TRI-STATE

A opção para o nosso caso recaiu sobre o circuito integrado 74126, cujo lay-out está mostrado na figura 17.

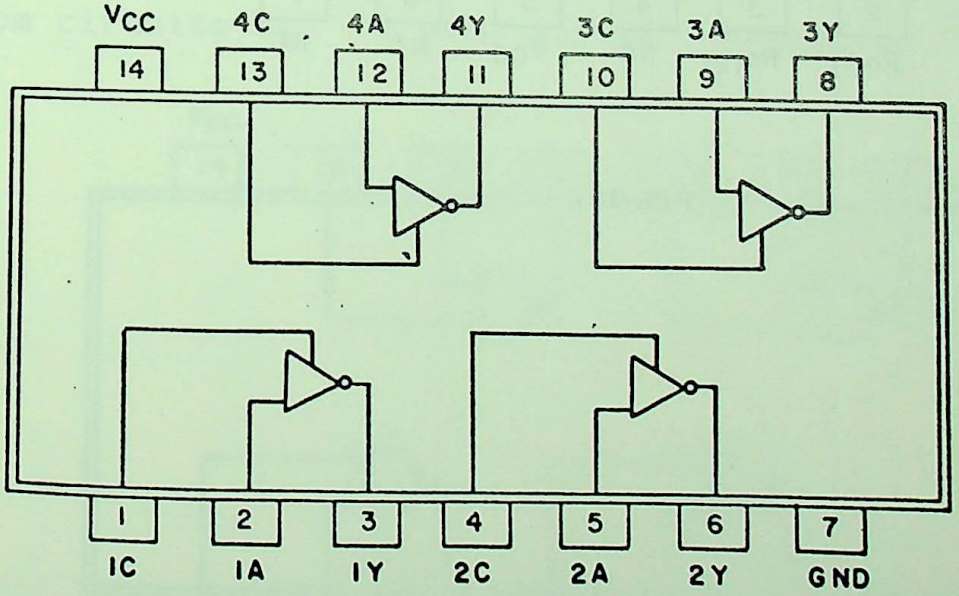


FIG. 17

- GERADORES DE ENDEREÇO

Para os geradores foram usados os contadores binários 7493 (fig.18) e 74193 (fig.19), sendo que o contador 7493 foi ligado na configuração de divisor por oito.

Os respectivos lay-out são representados nas figuras a seguir (fig. 18 e 19).

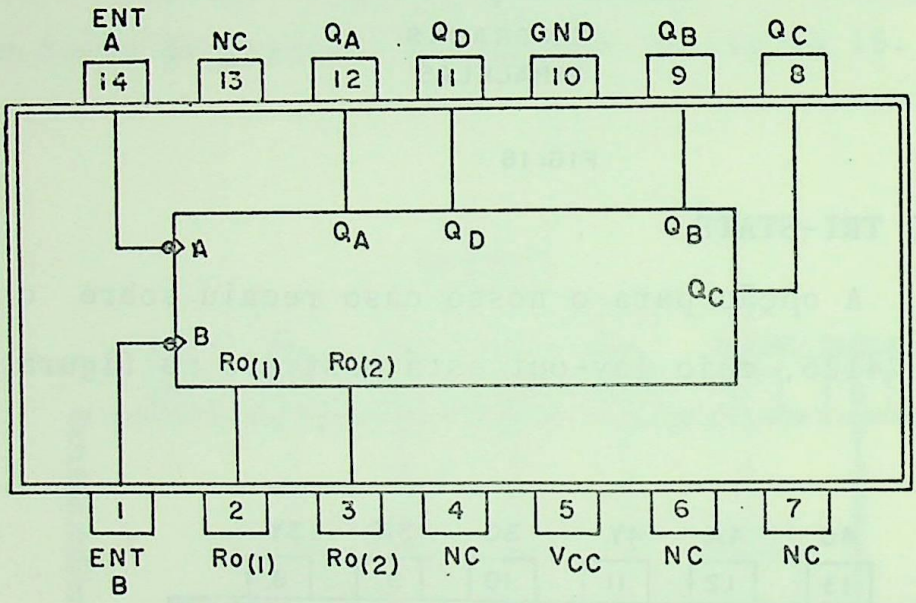


FIG. 18

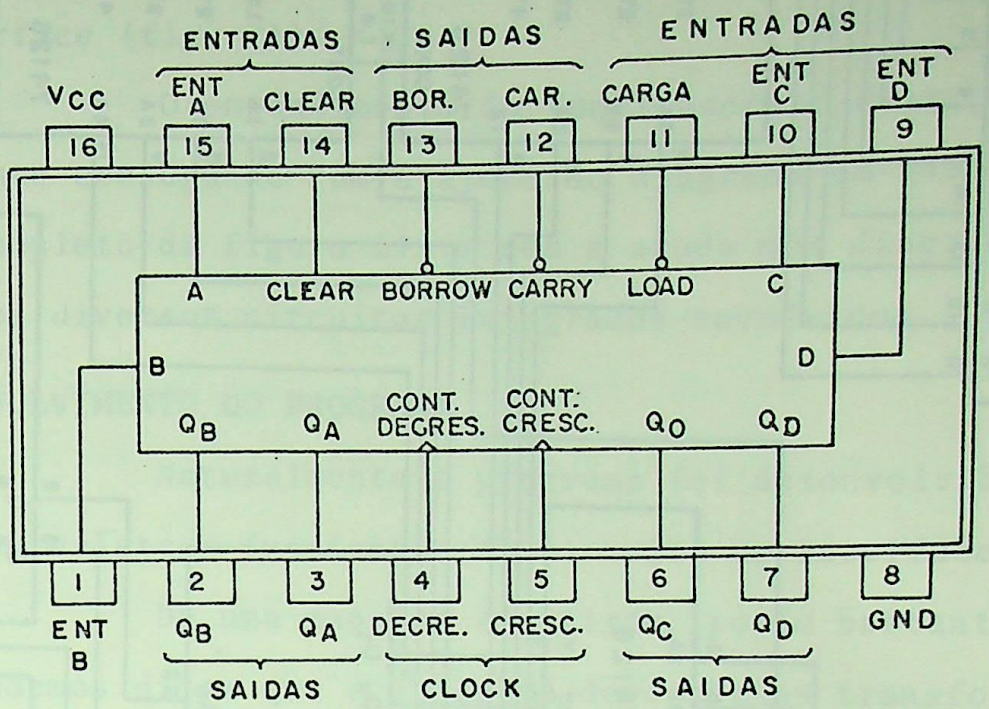


FIG. 19

- CIRCUITO PARA LÓGICA AUXILIAR

A fim de ajustar devidamente os vários sinais envolvidos no sistema, foi necessário uma lógica auxiliar, executada com circuito NÃO - E pelo circuito integrado 7400 (figura 20).

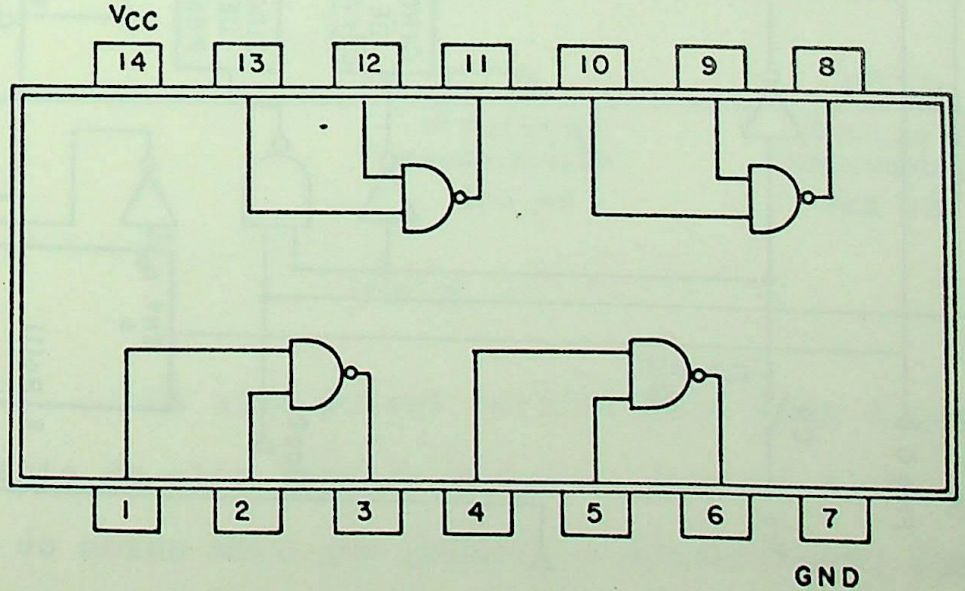


FIG. 20

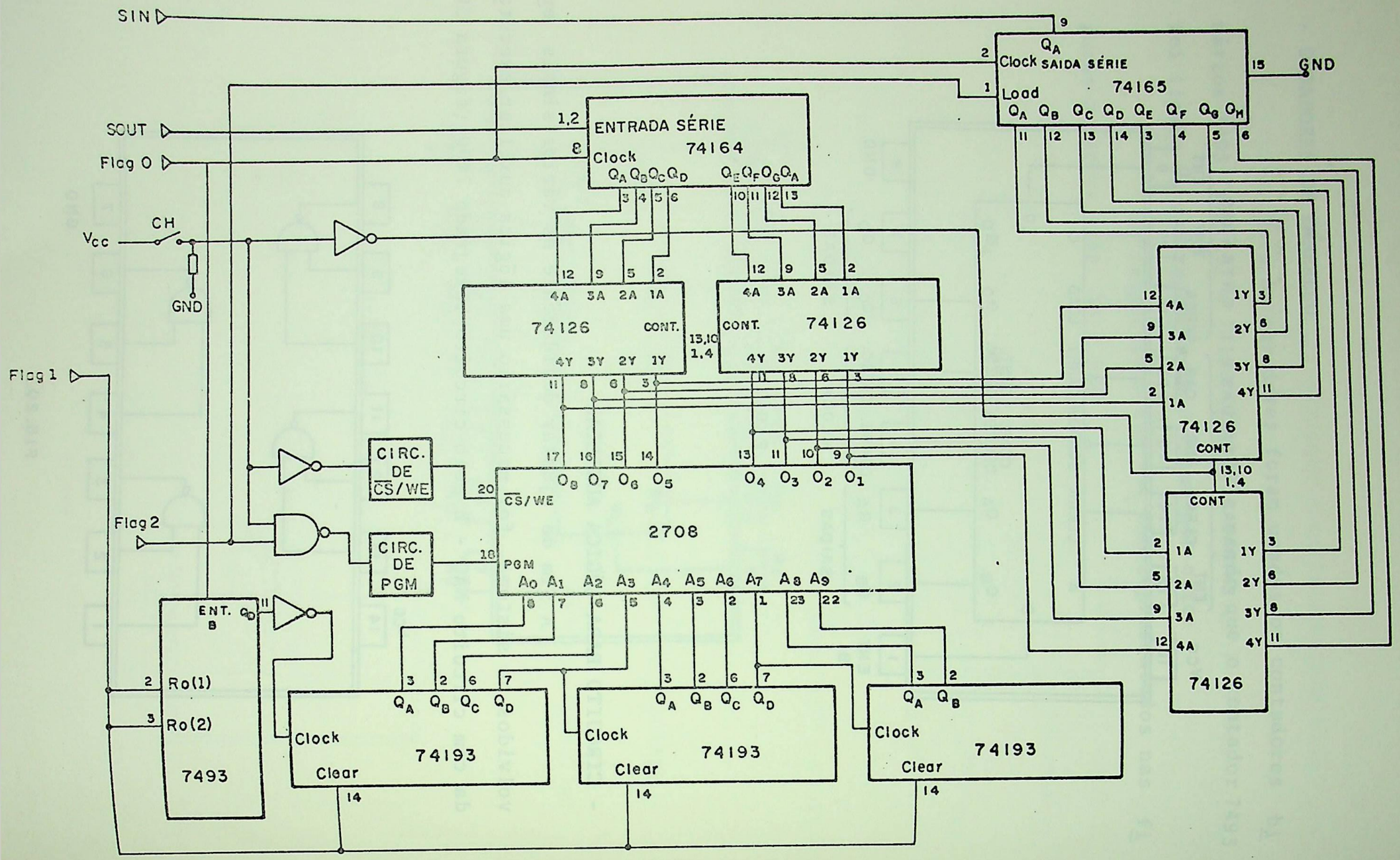

SIN
SOUT
Flag 0
Flag 1
Flag 2
Vcc
CH
GND
74165
Clock SAIDA SÉRIE
Load
Q_A Q_B Q_C Q_D Q_E Q_F Q_G Q_H
ENTRADA SÉRIE
74164
Clock
$Q_A Q_B Q_C Q_D$ $Q_E Q_F Q_G Q_A$
74126
4A 3A 2A 1A
CONT.
4Y 3Y 2Y 1Y
13,10
1,4
CIRC. DE CS/WE
CIRC. DE PGM
2708
$\overline{CS}$/WE
PGM
A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9
O_8 O_7 O_6 O_5 O_4 O_3 O_2 O_1
ENT. B
Ro(1)
Ro(2)
7493
74193
Clock
Clear

FIG. 21

Apresentados todos os circuitos envolvidos no interface, passamos agora a apresentação do circuito completo do interface (fig. 21).

O entendimento do funcionamento deste circuito é facilmente conseguido com a ajuda do diagrama em blocos do circuito completo da figura 14 e com a ajuda dos diagramas de pinagem dos diversos circuitos integrados envolvidos.

- DESENVOLVIMENTO DO PROGRAMA

Naturalmente o programa foi desenvolvido com base nas característcas funcionais do circuito de interface.

De uma maneira simplista, porém bastante elucidativa, podemos dizer que o programa deve apenas transformar dados paralelos em dados série e vice-versa e gerar os sinais de comando e temporização.

Deste modo, fazemos principalmente um levantamento da temporização necessária do circuito de interface. O resultado está mostrado na figura 22 para a fase de programação.

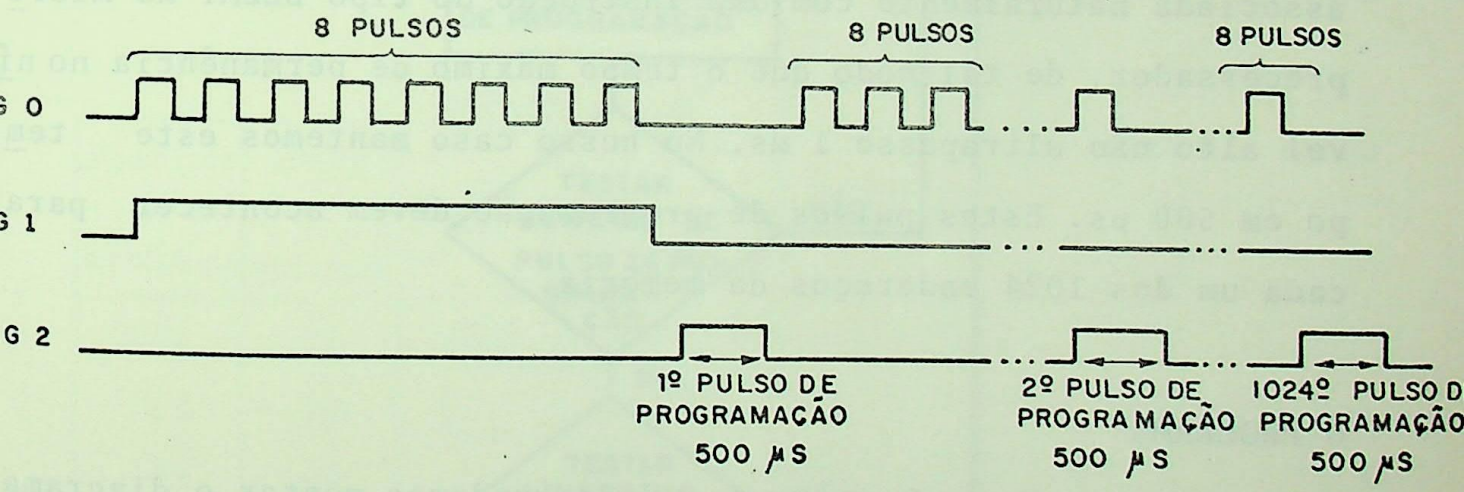


FIG. 22

Os oito pulsos gerados pelo flag 0 comandam o deslocamento do oito bits de dados da palavra a ser escrita na memória, do mesmo modo que promove o adiantamento dos contadores geradores de endereços.

O programa então deve promover o deslocamento de um bit da palavra de dado para a saída SOUT e logo após, deve aparecer um pulso na linha de comando Flag 0, para garantir que o shift-register de recepção de dado possa armazenar a palavra a ser processada.

Após os oito pulsos, toda a palavra (8 bits) estará armazenada no shift-register.

A linha de comando flag 1 promove o zeramento dos contadores geradores de endereço, como se faz necessário escrever na memória a partir do endereço zero, nos primeiros oito pulsosdo clock esta linha deve permanecer no nível alto, impedindo o adiantamento dos contadores de endereço.

Esta linha deve cair para o nível baixo, depois da linha de clock ter caído para o nível baixo, pois é o último dos oito pulsos que adianta os contadores de endereços.

A linha de comando Flag 2, como já mencionada, é responsável pela temporização do pulso de programação e deve ser associada naturalmente com uma instrução do tipo DELAY no microprecessador, de tal modo que o tempo máximo de permanência no nível alto não ultrapasse 1 ms. No nosso caso mantemos este tempo em 500 µs. Estes pulsos de programação devem acontecer para cada um dos 1024 endereços da memória.

O PROGRAMA

Disponíveis estes dados podemos montar o diagrama em blocos do programa (fig. 23).

Agora, de uma forma mais detalhada, temos o fluxograma da fig. 24.

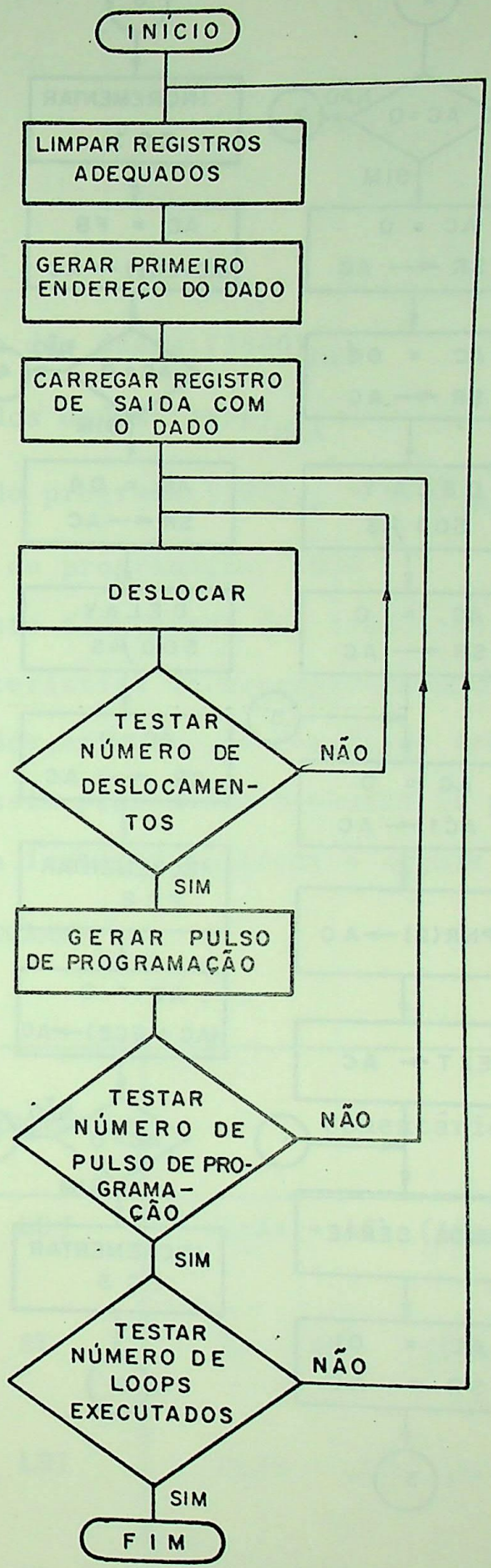

INÍCIO
LIMPAR REGISTROS ADEQUADOS
GERAR PRIMEIRO ENDEREÇO DO DADO
CARREGAR REGISTRO DE SAIDA COM O DADO
DESLOCAR
TESTAR NÚMERO DE DESLOCAMENTOS
NÃO
SIM
GERAR PULSO DE PROGRAMAÇÃO
TESTAR NÚMERO DE PULSO DE PROGRAMAÇÃO
NÃO
SIM
TESTAR NÚMERO DE LOOPS EXECUTADOS
NÃO
SIM
FIM

FIG. 23

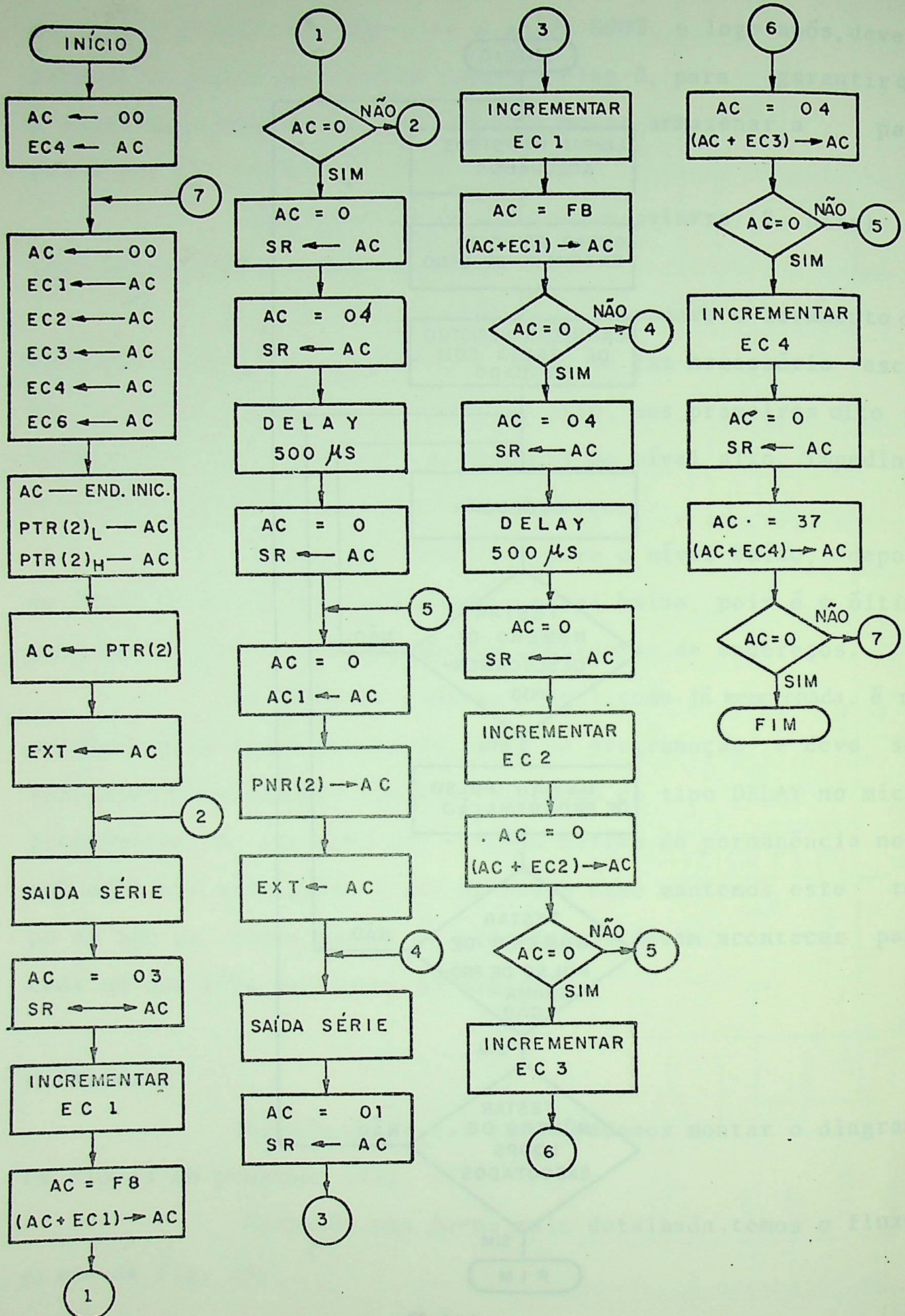

INÍCIO
AC ← 00
EC4 ← AC
7
AC ← 00
EC1 ← AC
EC2 ← AC
EC3 ← AC
EC4 ← AC
EC6 ← AC
AC — END. INIC.
PTR(2)L — AC
PTR(2)H — AC
AC ← PTR(2)
EXT ← AC
2
SAIDA SÉRIE
AC = 03
SR ↔ AC
INCREMENTAR EC 1
AC = F8
(AC + EC1) → AC
1
1
AC = 0
NÃO
2
SIM
AC = 0
SR ← AC
AC = 04
SR ← AC
DELAY 500 μS
AC = 0
SR ← AC
5
AC = 0
AC1 ← AC
PNR(2) → AC
EXT ← AC
4
SAÍDA SÉRIE
AC = 01
SR ← AC
3
3
INCREMENTAR EC 1
AC = FB
(AC+EC1) → AC
AC = 0
NÃO
4
SIM
AC = 04
SR ← AC
DELAY 500 μS
AC = 0
SR ← AC
INCREMENTAR EC 2
. AC = 0
(AC + EC2) → AC
AC = 0
NÃO
5
SIM
INCREMENTAR EC 3
6
6
AC = 04
(AC + EC3) → AC
AC = 0
NÃO
5
SIM
INCREMENTAR EC 4
AC = 0
SR ← AC
AC = 37
(AC+EC4) → AC
AC = 0
NÃO
7
SIM
FIM

FIG. 24

Neste fluxograma temos os seguintes números endereços: EC 1 → FA

EC 2 → FB

EC 3 → FC

EC 4 → FD

EC 5 → FE

EC 6 → FF

Primeiro endereço dos dados: $(3800)_{HEX}$

Último endereço dos dados: $(38FF)_{HEX}$

Número de loops do programa: $(201)_{10} = (C9)_{HEX}$

Número de pulsos de programação: 1024

Este fluxograma foi traduzido para a linguagem de máquina característica do microprocessador SC/MP, com ajuda das folhas de dados, em anexo, no fim deste trabalho.

Assim procedendo a partir de posição de memória $(0001)_{HEX}$ temos a listagem mostrada a seguir.

- LISTAGEM DO PROGRAMA

Posição de memória	Código de operação	minemônico	Comentário
0001	C4	LDI	Dado → AC (AC=0)
02	00		
03	C8	ST	AC → EC4 (EC4 = FD)
04	F9		
05	C4	LDI	Dado → AC (AC = 0)
06	00		
07	C8	ST	AC → EC2 (EC2 = FB)
08	F3		

09	C8	ST	AC → EC3 (EC3 = FC)
0A	F2		
0B	C8	ST	AC → EC1 (EC1 = FA)
0C	EE		
0D	C8	ST	AC → EC5 (EC5 = FE)
0E	F0		
0F	C8	ST	AC → EC6 (EC6 = FF)
10	EF		
11	32	XPAL	AC ↔ $PTR(2)_L$
12	C4	LDI	38 → AC
13	38		
14	36	XPAH	AC ↔ PTR(2)H (PTR(2) = 3800)
15	C6	LD	PTR(2) → AC (PTR(2) = PTR(2) + 1)
16	01		
17	01	XAE	AC ↔ Extensão (EXT=PTR(2)=3800)
18	19	SIO	Saída Série (1º Bit)
19	C4	LDI	03 → AC
1A	03		
1B	07	CAS	AC → SR SR=03 (Flg0=1, Flg1=1, Flg2=0)
1C	C4	LDI	02 → AC
1D	02		
1E	07	CAS	AC → SR SR=02 (Flg0=0, Flg1=1, Flg2=0)
1F	A8	ILD	(EC1 + 1) → EC1, AC
20	DA		
21	C4	LDI	F8 → AC (F8=complemento de dois de 8)
22	F8		
23	F0	ADD	(AC + EC1) → AC

24	D6		
25	9C	JNZ	Se AC ≠ 0 pular para endereço 18
26	F1		
27	C4	LDI	00 → AC
28	00		
29	07	CAS	AC → SR
2A	C4	LDI	04 → AC
2B	04		
2C	07	CAS	AC → SR (Pulso de programação)
2D	8F	DLY	Delay (500 μs)
2E	01		
2F	C4	LDI	00 → AC
30	oo		
31	07	CAS	AC → SR (fim do pulso de programa)
32	C4	LDI	00→ AC
33	00		
34	C8	ST	AC→ EC1 (EC1 = FA = 0)
35	C5		
36	C6	LD	PTR (2) → AC (PTR(2)+1) PTR(2)
37	01		
38	01	XAE	AC ↔ EXT EXT = PTR(2) (3801....3BF
39	19	SIO	sáída série (1 bit)
3A	C4	LDI	01 → AC
3B	01		
3C	07	CAS	AC → SR (SR = 01)
3D	C4	LDI	00 → AC
3E	00		
3F	07	CAS	AC → SR (SR = ∞)
40	A8	ILD	(EC1+1)→ EC1 (EC1 = FA)

41	B9		
42	C4	LDI	F8 → AC (F8 = compl. de dois de 8)
43	F8		
44	FO	ADD	(AC+EC1) → AC
45	B5		
46	9C	JNZ	SE AC ≠ 0 pular para end. 39
47	F1		
48	C4	LDI	04 → AC
49	04		
4A	07	CAS	AC → SR (SR = 04) (Pulso de Program)
4B	8F	DLY	DELAY (500 μs)
4C	01		
4D	C4	LDI	AC→ 00
4E	00		
4F	07	CAS	AC → SR (Fim do pulso de program.)
50	A8	ILD	(EC2+1) → EC2 (EC2 = FB)
51	AA		
52	C4	LDI	01 → AC (limite de endereço)
53	00		
54	E0	XOR	(AC⊕ EC2) → AC
55	A6		
56	9C	JNZ	Se AC ≠ 0 pular para End. 32
57	DA		
58	A8	ILD	(EC3+1)→ EC3
59	A3		
5A	C4	LDI	00 →AC
5B	00		
5C	07	CAS	AC → SR
5D	C4	LDI	04 → AC (limite de endereço)

5E	04		
5F	EO	XOR	(AC ⊕ EC3) → AC
60	9C		
61	9C	JNZ	Se AC ≠ 0 pular para end. 32
62	CF		
63	A8	ILD	(EC4+1) → EC4
64	99		
65	C4	LDI	00 → AC
66	00		
67	07	CAS	AC → SR
68	C4	LDI	37 → AC (37 = compl. de 2 de C9)
69	37		$(C9)_{nex} = (201)_{10}$
6A	FO	ADD	(AC+EC4) → EC4
6B	92		
6C	9C	JNZ	se AC ≠ 0 pular para end. 05
6D	97		
6E	00		
6F	08		

Levando em conta o mesmo procedimento da fase de programação, a temporização da fase de leitura esta mostrada na figura 25.

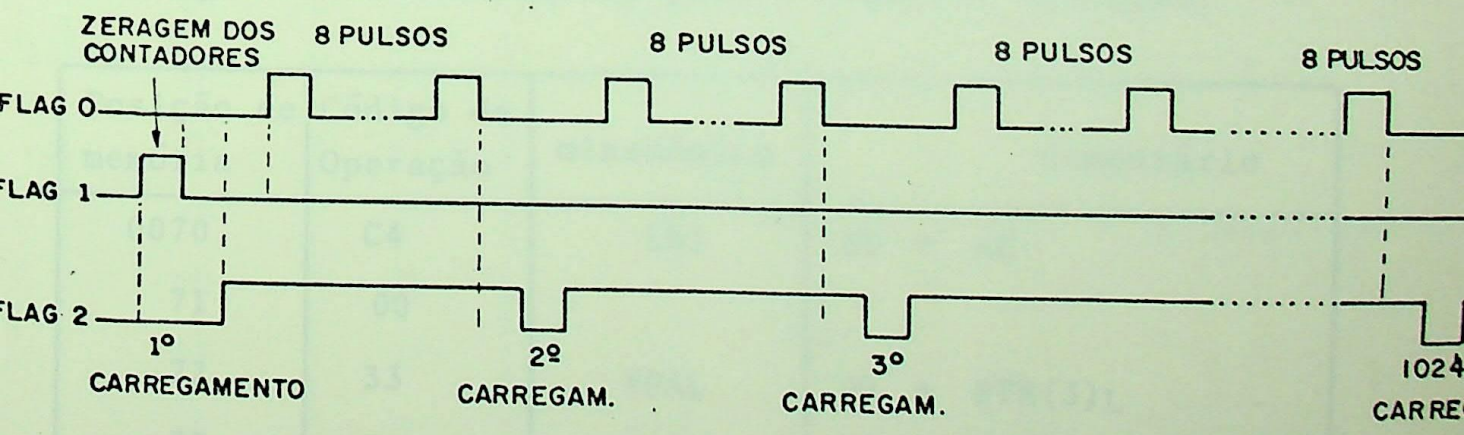


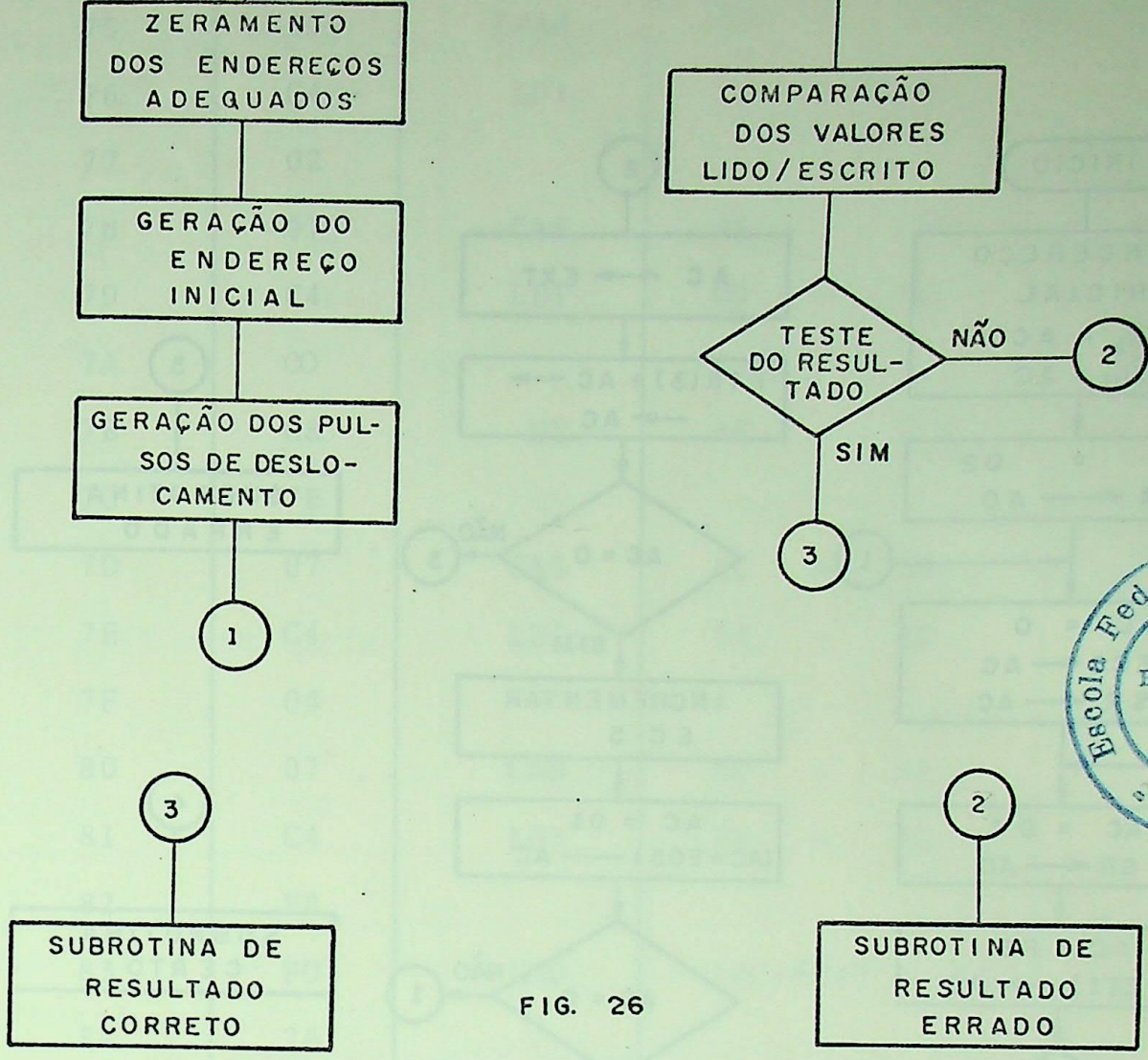
FIG. 25

Observando a carta de tempo da figura 25, vemos que o programa deve prover basicamente o pulso de carregamento do Shift-register, armazenando assim, o dado contido na memória; após este carregamento, deve apresentar os pulsos necessários para deslocar o dado para o interior do microprocessador.

O Flag 1 apresenta apenas um nível lógico positivo em todo o processo, pulso este necessário ao zeramento dos contadores de endereço.

O objetivo da fase de leitura é promover a auto-análise dos dados escritos na memória. A leitura da memória - EPROM é simultâneamente comparada com o dado original armazenado na memória RAM do microprocessador SC/MP e caso exista alguma diferença entre estes dois valores, o programa automaticamente apresenta um sinal de erro, transcrito pela letra E no display do microprocessador. Em caso de não haver discrepâncias entre o valor provado e o original, o display, ao fim do ciclo de todas as 1024 posições de memória, apresentará um sinal de gravação correta, transcrito pela letra C no display.

Deste modo, o diagrama em blocos deste programa está mostrado de modo resumido na figura 26



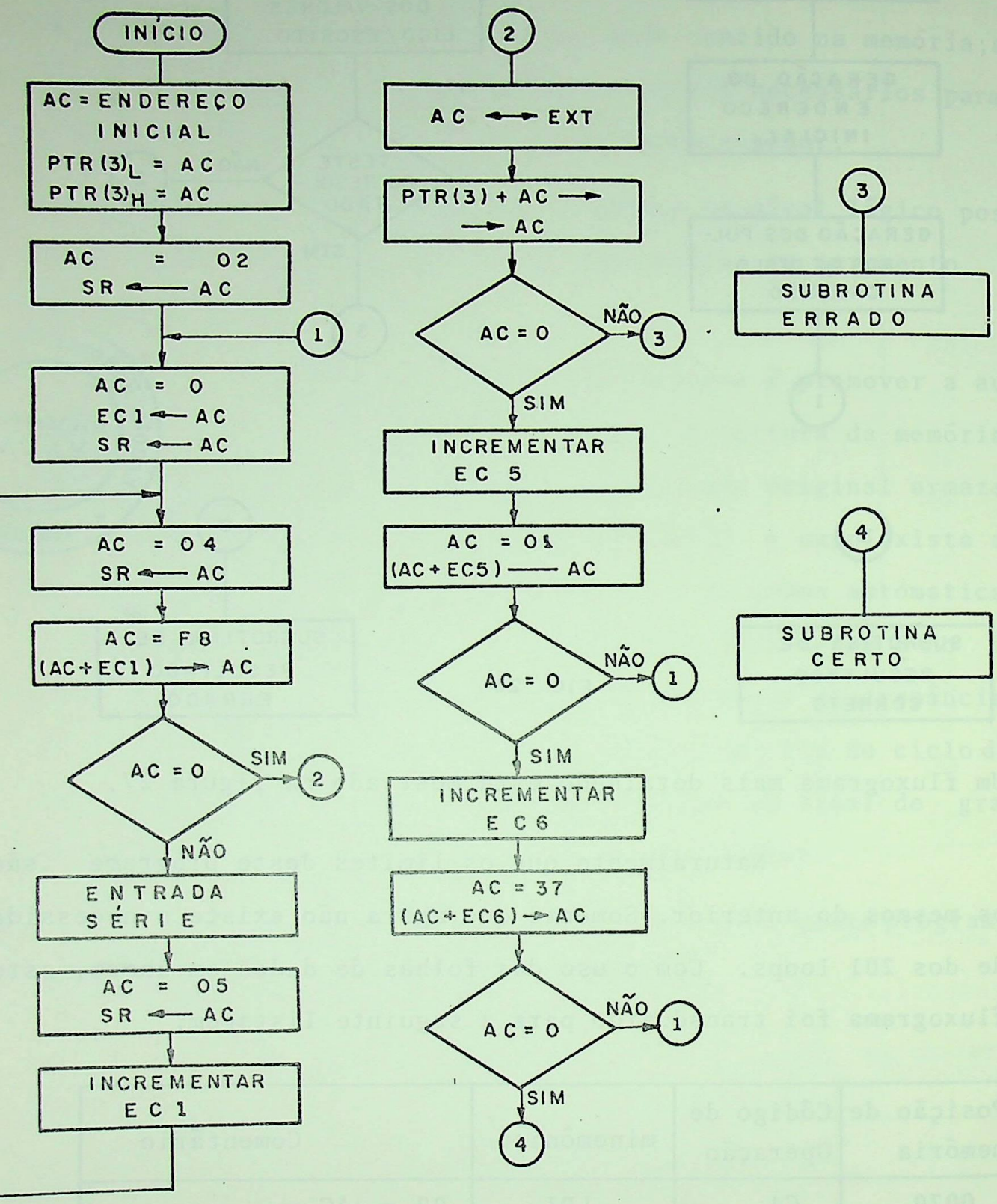
Escola Federal de Itajubá
Biblioteca MAUA

FIG. 26

Um fluxograma mais detalhado está mostrado na figura 27.

Naturalmente que os limites deste programa são os mesmos do anterior. Somente que, agora não existe a necessidade dos 201 loops. Com o uso das folhas de dados em anexo, este fluxograma foi transduzido para a seguinte listagem.

Posição de memória	Código de Operação	minemônico	Comentário
0070	C4	LDI	00 → AC
71	00		
72	33	XPAL	00 → $PTR(3)_L$
73	C4	LDI	38 → AC
74	38		


INÍCIO
AC = ENDEREÇO INICIAL
PTR(3)L = AC
PTR(3)H = AC
AC = 02
SR ← AC
1
AC = 0
EC1 ← AC
SR ← AC
AC = 04
SR ← AC
AC = F8
(AC + EC1) → AC
AC = 0
SIM
2
NÃO
ENTRADA SÉRIE
AC = 05
SR ← AC
INCREMENTAR EC 1
2
AC ↔ EXT
PTR(3) + AC → AC
AC = 0
NÃO
3
SIM
INCREMENTAR EC 5
AC = 08
(AC + EC5) → AC
AC = 0
NÃO
1
SIM
INCREMENTAR EC 6
AC = 37
(AC + EC6) → AC
AC = 0
NÃO
1
SIM
4
3
SUBROTINA ERRADO
4
SUBROTINA CERTO

FIG·27

	75	37	XPAH	38 → PTR(3)H
	76	C4	LDI	02 → AC
	77	02		
	78	07	CAS	AC → SR
	79	C4	LDI	00 → AC
	7A	00		
	7B	C8	ST	AC → EC1
	7C	7E		
	7D	07	CAS	AC → SR
	7E	C4	LDI	04 → AC
	7F	04		
	80	07	CAS	AC → SR
	81	C4	LDI	F8 → AC
	82	F8		
	83	FO	ADD	(AC+EC1) → AC
	84	76		
	85	98	JZ	Se AC = 0 saltar para 8F
	86	08		
	87	19	SIO	Entrada série de um BIT de dado
	88	C4	LDI	05 → AC
	89	05		
	8A	07	CAS	AC → SR
	8B	A8	ILD	(EC1+1) → EC1
	8C	6E		
	8D	90	JMP	Saltar para o endereço 7E
	8E	EF		
	8F	01	XAE	AC ↔ EXT.
*	90	E7	XOR	(AC ⊕ PTR(3)) → AC (PTR(3)+1)→PTR1
*	91	01		

* 92	9C	JNZ	Se AC ≠ 0 saltar para subrotina ERRADO
* 93	2C		
94	A8	ILD	(EC5+1) → EC5
95	69		
96	C4	LDI	FF → AC
97	FF		
98	FO	ADD	(EC5+AC) → AC
99	65		
9A	9C	JNZ	Se AC ≠ 0 saltar para END. 79
9B	DD		
9C	A8	ILD	EC6+1 → EC6.
9D	62		
9E	C4	LDI	FB → AC
9F	FB		
A0	FO	ADD	(EC6+AC) → AC
A1	5E		
A2	9C	JNZ	Se AC≠0 saltar para END. 79
A3	D5		
A4	90	JMP	Saltar para subrotina CERTO
A5	OA		
A6	00		

Passamos agora a listagem das duas subrotinas de CERTO e ERRADO; que devido a sua simplicidade não exige a elaboração de um fluxograma.

Posição da memória	Código de Operação	Minemônico	Comentário
			SUBROTINA CERTO (C)
00B0	C4	LDI	20 → AC
B1	20		
B2	31	XPAL	20 → $PTR(1)_L$
B3	C4	LDI	70 → AC
B4	70		
B5	35	XPAH	70 → $PTR(1)_H$
B6	C4	LDI	39 → AC (Letra C)
B7	39		
B8	C9	ST	AC → PTR(1)
B9	00		
BA	90	JMP	Saltar para endereço B0
BB	F4		
BC	00		

Posição da memória	Código de Operação	Minemônico	Comentário
			SUBROTINA ERRADO (E)
00C0	C4	LDI	20 → AC
C1	20		
C2	31	XPAL	AC → $PTR(1)_L$
C3	C4	LDI	70 → AC
C4	70		
C5	35	XPAH	AC → $PTR(1)_H$
C6	C4	LDI	39 → AC (letra E)
C7	79		
C8	C9	ST	AC → PTR (1)
C9	00		
CA	90	JMP	Saltar para endereço C0
CB	F4		
CC	00		

FONTE

Para as características especiais de funcionamento e programação de memória EPROM 2708, foi necessário desenvolver uma fonte com características especiais, isto é, a apresentação simultânea das tensões +26 V, +12 V, +5 V, 0 V, -5 V e -12 V.

Com o auxílio de diodos Zener e uma configuração adequada (mostrada na figura 28) foi possível conseguirmos tal gama de tensões com apenas um transformador.

PREPARAÇÃO DA RAM DO SC/MP PARA ARMAZENAMENTO DOS DADOS A SEREM GRAVADOS

O armazenamento dos 1024 bytes a serem gravados na memória PROM serão feitos na expansão de memória de módulo SC/MP LCDS que possui uma capacidade de 2048 bytes. Este módulo de memória necessita de um código de entrada, a fim de dar abertura ao processo da leitura e escrita na memória. Este código deve ser programado pela memória através de "jumps" no soquete já preparado para isto. No caso específico do nosso trabalho usamos o código:

Binário -	0011	1XXX	XXXX	XXXX
Hexa -	3	8(ou mais)	X	X

Este código foi conseguido com a execução das ligações mostradas na figura 29 pino 8, ligado ao 10, 11 e 12, e pino 16, ligado ao 13 e 14.

Esta ligação representa em diagrama de blocos a informação apresentada na figura 30.

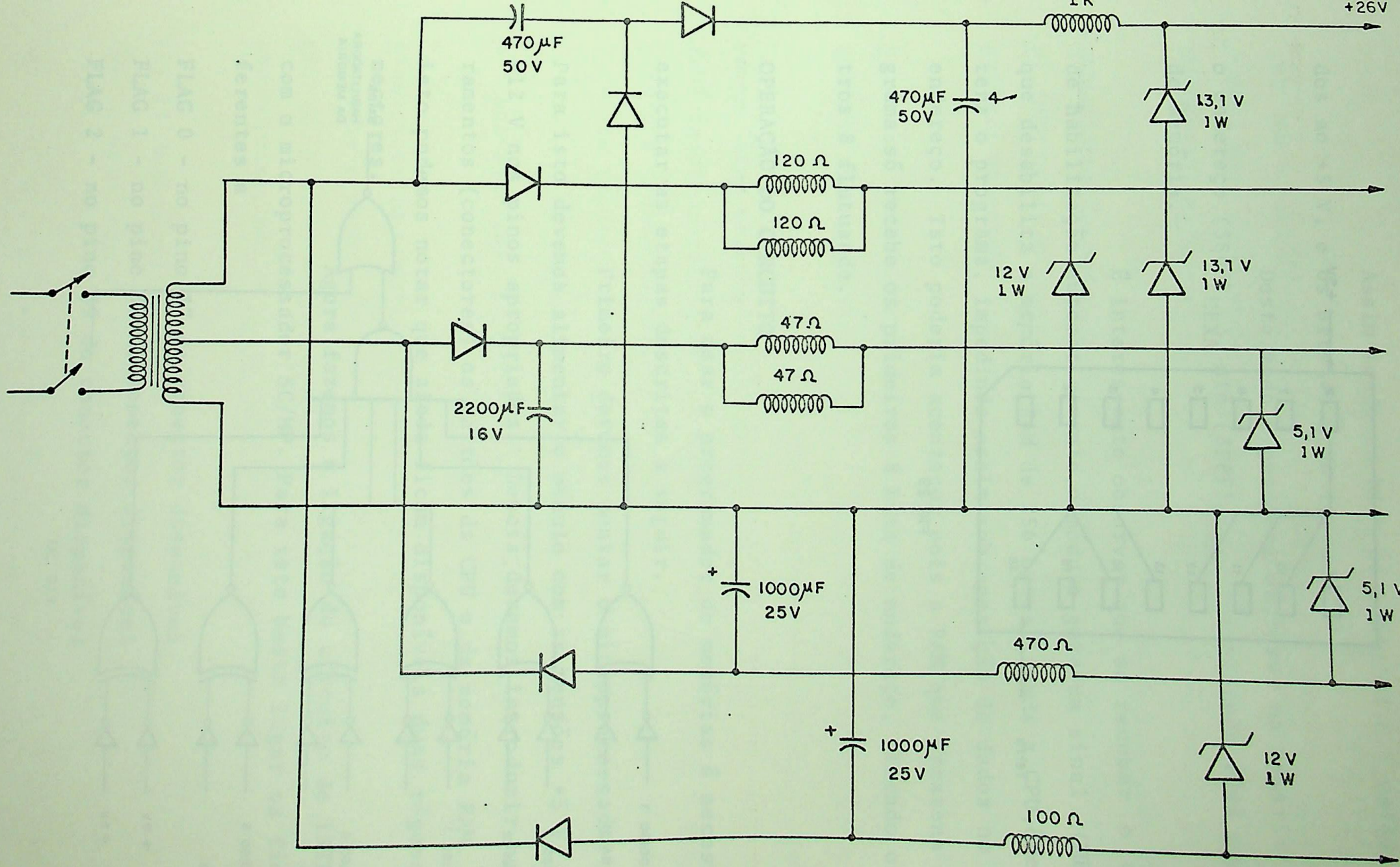

470µF
50V
1K
+26V
470µF
50V
13,1 V
1W
120 Ω
120 Ω
12 V
1W
13,1 V
1W
47 Ω
47 Ω
2200µF
16V
5,1 V
1W
1000µF
25V
5,1 V
1W
470 Ω
1000µF
25V
12 V
1W
100 Ω

FIG. 28

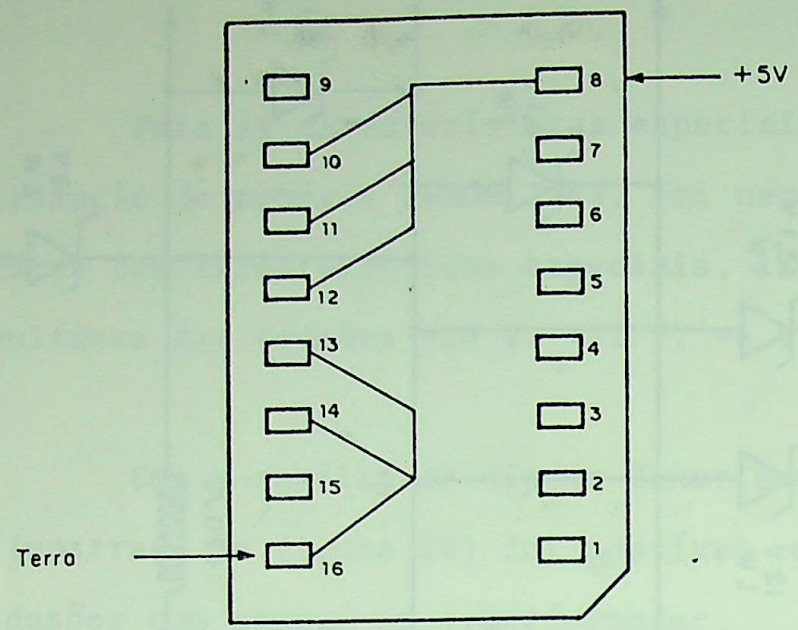

9
10
11
12
13
14
15
16
8
7
6
5
4
3
2
1
+5V
Terra

FIG. 29

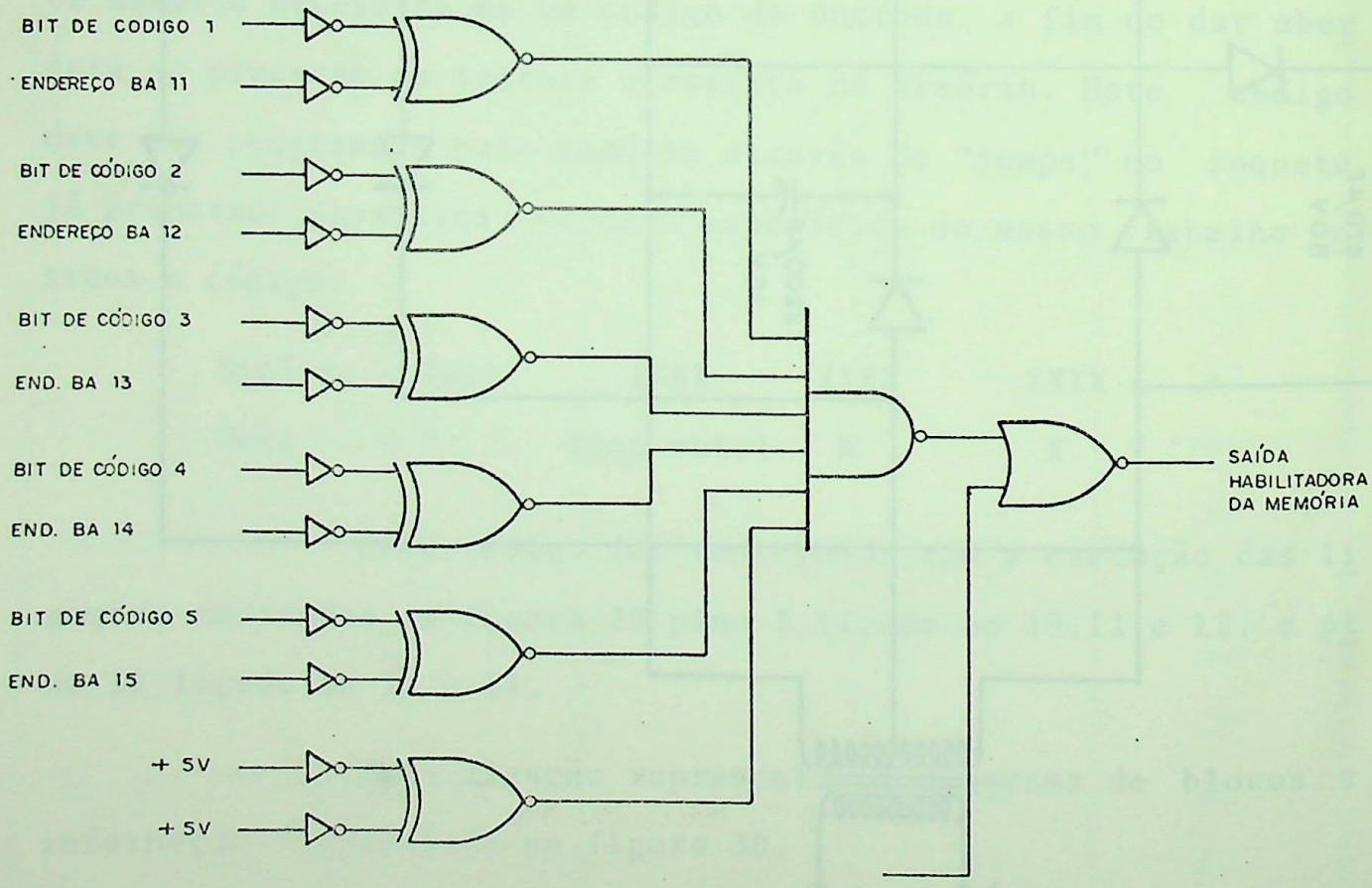

BIT DE CODIGO 1
ENDEREÇO BA 11
BIT DE CÓDIGO 2
ENDEREÇO BA 12
BIT DE CÓDIGO 3
END. BA 13
BIT DE CÓDIGO 4
END. BA 14
BIT DE CÓDIGO 5
END. BA 15
+ 5V
+ 5V
SAÍDA
HABILITADORA
DA MEMÓRIA

FIG. 30

Assim sendo, os bits de código 1,2,3 estão ligados ao +5 V, e os bits 4, 5, ao terra.

Deste modo, é possível escrever na memória desde o endereço $(3800)_{HEXA}$ até $(3FFF)_{HEXA}$, totalizando 2048 posições de memória.

É interessante observar que ao receber o código de habilitação, automaticamente, o módulo gera um sinal (MEMSEL) que desabilita a memória RAM de 256 bytes ligada a CPU, que armazena o programa, impedindo assim sobreposição de dados no mesmo endereço. Isto poderia acontecer, pois a RAM que armazena o programa só recebe os primeiros 8 bits de endereço, ficando os outros 8 flutuando.

OPERAÇÃO DO CIRCUITO

Para usar o programador de memórias é necessário executar as etapas descritas a seguir.

Primeiro devemos montar o microprocessador SC/MP. Para isto devemos alimentar o módulo com as tensões +5 V, terra, -12 V nos pinos apropriados. Depois, devemos introduzir nos barramentos (conectores) os cartões da CPU e da memória RAM. Feito isto podemos notar que ainda ficam disponíveis duas réguas de conectores.

Agora fazemos a ligação do circuito de INTERFACE com o microprocessador SC/MP. Para isto basta ligar os fios referentes a

FLAG 0 - no pino 37 do conector disponível
FLAG 1 - no pino 38 do conector disponível
FLAG 2 - no pino 39 do conector disponível

BSOUT - no pino 34 do conector

BSIN - no pino 33 do conector

As chaves HALT MODE, HALT INST e RUN MODE devem estar posicionadas para a direita, isto é, na posição PNL,DEBUG e CONT respectivamente.

Ligamos as fontes e atuamos na tecla INIT, o display deve apresentar 0001 PC. Através da tecla F,(LOAD MEM) introduzimos o programa nas posições de memória adequadas (indicadas na listagem).

Depois,através da tecla MEM ADDR/ABORT e da tecla F (LOAD MEM),introduzimos os dados a serem gravados na EPROM na RAM, a partir do endereço $(3800)_{HEX}$.

Reatuamos na tecla INIT e o circuito está pronto para a gravação. A memória EPROM já deve estar no lugar devido. Então basta atuar na tecla RUN e o sistema executa a programação em 4,2 minutos. Ao final dos quais irá aparecer no display o endereço 006F com a indicação de instrução 08.

Neste ponto a memória EPROM já está gravada e passamos a fase de testes de gravação.

Atuamos novamente na tecla RUN e o microprocessador passa a testar todas as posições gravadas. Caso a gravação tenha se processado corretamente,irá aparecer a letra C no display após um breve intervalo de tempo, em caso contrário,irá aparecer a letra E.

Em caso de erro,basta atuar na tecla READ P3 (3) e o display apresentará o endereço que está errado, ou seja, que proporcionou o primeiro erro.

Para uma análise mais profunda, é possível monitorar com o osciloscópio as saídas de controle FLAG 0, FLAG 1 e FLAG 2 que devem apresentar a mesma forma de onda mostrada nas figuras 22 e 25.

Em caso do programa estar gravado de modo errado, a verificação pode ser feita através da tecla READ MEM (7).

C O N C L U S Ã O

Neste ponto, atingimos o fim do trabalho e, logicamente, as metas estabelecidas no início do mesmo. Como é natural, se faz necessário promovermos uma avaliação das potencialidades do sistema projetado, para detetarmos as capacidades globais do mesmo.

Após exaustivos testes do circuito com várias memórias, programando e reprogramando podemos perceber que com algumas alterações executadas em poucas instruções do programa monitor, poderíamos fazer com que nosso sistema pudesse atuar como programador de várias outras memórias, como o caso da 2716, 2048, 5204, bastando apenas mudar os limites do programa de acordo com o total de posições disponíveis da memória, bem como as características de programação. Como todos estes dados são diretamente controlados pelo microprocessador, vimos que, com poucas modificações, o nosso programador se transforma em quase um programador universal de memórias EPROM.

Por exemplo, no caso da memória 2716 que possui 2048 posições de memória, bastaria mudar a instrução.

Posição de Memória	Código de Operação
5 E	04 → 08

Este resultado era totalmente esperado e o projeto já foi feito com este intuito, porém, a novidade apareceu na capacidade que o circuito apresentou de copiar memórias já gravadas, a fim de posteriormente gravar outras. Isto simplificou enormemente o trabalho mecânico de introdução dos dados na

memória RAM do microprocessador SC/MP, sem deixar de acentuar naturalmente a facilidade de duplicar os aparelhos e sistemas' estrangeiros, cuja alma realmente se encontra em tais memórias, (objetivo este de louvor duvidável, naturalmente, cá entre nós brasileiros). Esta nova opção do circuito foi possível de se conseguir com a alteração de apenas quatro endereços que são:

posição de memória	Código de Operação	Novo código de operação
90	E7	CF
91	01	01
92	9C	08
93	2C	08

Isto basta para o circuito copiar qualquer memória de 1024 posições. Para uma memória maior, basta mudar o limite de endereços.

Esta particularidade do circuito nos foi de grande valia, pois a dificuldade nas importações de componentes tinha nos atirado exatamente, neste momento, a uma encruzilhada, quanto ao término do desenvolvimento de um sistema com microprocessador, baseado em um similar encontrado no mercado americano. Não deixando de salientar que mesmo na fase de montagem do protótipo em "proto-boards", este já se mostrou útil e necessário as ocorrências de nosso laboratório, programando algumas memórias que hoje funcionam em nossos sistemas.

Este trabalho como já mencionado no início, nasceu de uma necessidade de nosso laboratório, desenvolveu-se dentro das contingências e limites de nossa realidade, porém che

gamos a um alto grau de eficiência em seu término, cabendo a ele mais um passo na concretização de nossas aspirações como engenheiro e educador, na consolidação do crescimento do nível técnico científico de nosso país.

DATA ___/___/ 19__
PROC. ___
PED. ___
LIV. ___
NCr$ ___

I N V E N T Á R I O
BIM - EFEI

DATA	Rubrica

EFEI - BIBLIOTECA MAUÁ
8200337
NÃO DANIFIQUE ESTA ETIQUETA