

UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CONTROLE DE VELOCIDADE DE MOTOR C.C.

USANDO MICROCOMPUTADOR

Dissertação submetida à Universidade Federal de Santa Catarina para a obtenção do grau de Mestre em Engenharia.

Joni da Silva Fraga

maio 1979

"CONTROLE DE VELOCIDADE DE MOTOR DE CORRENTE CONTÍNUA USANDO  
MICROCOMPUTADOR"

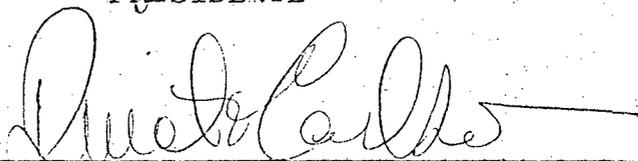
CANDIDATO: JONI DA SILVA FRAGA

Esta dissertação foi julgada para a obtenção do título de MESTRE EM CIÊNCIAS Especialidade Engenharia Elétrica e aprovada em sua forma pelo curso de Pós-Graduação.



---

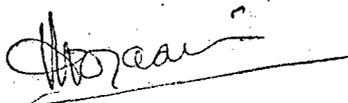
PROF. JACQUES SCHONEK, Dr. ING.  
PRESIDENTE



---

PROF. RENATO CARLSON, DR. ING.

Banca Examinadora:



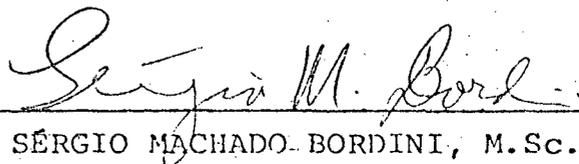
---

PROF. RAJAMANI DORAISWAMI, Ph. D.



---

PROF. CELSO P. BOTTURA, Dr. ING.



---

PROF. SÉRGIO MACHADO BORDINI, M.Sc.

## AGRADECIMENTOS

À CAPES e a UFSC pelo apoio financeiro.

Ao Prof. Jacques Schoneck, pela inestimável ajuda prestada na orientação deste trabalho.

A todos que colaboraram ou pelo incentivo para tornar possível a realização deste trabalho em particular, as seguintes pessoas:

Renato Carlson

Hans Helmut Zürn

Jean Loup Farges

Marcos Cardoso Filho

Luiz Gonzaga de Souza Fonseca

Carlos Raul Borenstein

Aos funcionários do Departamento de Engenharia Elétrica

## R E S U M O

Este trabalho consiste de um sistema de controle de velocidade de uma máquina de corrente contínua usando microcomputador.

A máquina é alimentada na armadura por um retificador a tiristores ligado à rede, com excitação de campo constante. As variações de tensão de armadura são comandadas pelo microcomputador a partir de:

- Informação de velocidade: valores da velocidade atual da máquina são obtidos com o uso de um tacômetro digital.

- Informação de corrente: valores da corrente de armadura são obtidos com o uso de um conversor analógico/digital.

- Velocidade de referência: um jogo de chaves é responsável pela introdução da velocidade de referência no microcomputador.

O algoritmo de controle de velocidade tem o comportamento de um compensador integral com o ganho variável, e inclui um limitador de corrente.

Este sistema é praticamente todo digital, constituindo-se portanto, numa regulação numérica. São apresentados resultados experimentais que atestam o bom funcionamento do conjunto e mostram um sistema bastante operacional.

## ABSTRACT

This work analyzes the speed control system of a direct current machine using a microcomputer.

The machine is armature fed by a thyristor rectifier connected to the standard network, with constant field excitation. The changes in armature voltage are controlled by the microcomputer originating from:

- Speed information: actual speed of machine is measured by a digital tachometer.
- Current information: values of armature current are obtained via an analog to digital converter.
- Reference speed: a switch array feeds the reference speed into the microcomputer.

The speed control algorithm behaves as an integral compensator with variable gain and includes a current limiter.

This systems is almost totally digital, and as such constitutes a numerical regulation (control).

Experimental results are presented to attest the good operation of the assembly.

## I N D I C E

1.	Introdução .....	01
2.	Generalidades .....	07
3.	Descrição do Sistema .....	46
4.	Resultados .....	101
5.	Conclusões .....	109
6.	Apêndice - Diagramas e Tabelas .....	111

## CAPÍTULO I

### INTRODUÇÃO

O surgimento de tecnologias como a integração em larga escala (LSI), tornou possível uma nova geração de pequenos e poderosos computadores, os minicomputadores. Com estes equipamentos, bem mais baratos que os anteriores, começaram a ocorrer grandes mudanças.

Os minicomputadores deixaram os centros de processamento e passaram a ser utilizados em oficinas e laboratórios. Foram postos, imediatamente, a trabalhar como substitutos da lógica de controle não programável.

O minicomputador forneceu uma flexibilidade para projetos, mas sempre, com desvantagem no custo, pois somente os equipamentos mais caros poderiam justificar o preço de um minicomputador. Assim, enquanto, por um lado, o minicomputador fez um considerável avanço em aplicações com máquinas e equipamentos de alto custo, por outro lado, a maioria das aplicações de controle permaneceram sob o domínio da lógica não programável.

Porém o desenvolvimento de calculadoras eletrônicas possibilitou o aparecimento de um novo elemento, o microprocessador. Com capacidade próxima a dos minicomputadores e com preços cada vez mais competitivos com os dos sistemas de lógica não programável, o que leva o microcomputador a ocupar a faixa de aplicações entre a lógica não programável e o minicomputador.

Por razões de preço e flexibilidade, é evidente que a característica principal dos microcomputadores é a aplicação dedicada. Com isto, nos sistemas onde são usados, os microcomputadores possuem os programas de controle da aplicação gravados em

suas memórias fixas (ROM). Outro aspecto importante é que a simples troca de uma ROM, por outra com novos programas, modifica o comportamento do sistema, provando a versatilidade no uso de microcomputadores. Também, a incorporação de microcomputadores em circuitos de controle pode acrescentar funções bem complexas, como alarmes e monitoração.

Por outro lado, há uma maior tendência a se utilizar cada vez menos sistemas baseados em minicomputadores, pois muitos deles apresentam mais capacidade de processamento do que é realmente necessário nos trabalhos em que são utilizados. O microcomputador embora mais lento é bastante adequado para inúmeras aplicações por ser programável, pequeno, com manutenção simples, etc... Uma destas aplicações é o controle de velocidade de um motor de corrente contínua, este tipo de regulação pode ser de grande utilidade na indústria, em máquinas de controle numérico, como fresadoras, tornos, etc...

O objetivo deste trabalho foi o desenvolvimento de uma montagem operacional que permite o controle de velocidade, sem compromissos maiores com o desenvolvimento de algoritmos de controle.

O motor de corrente contínua é controlado variando a tensão aplicada na armadura, variação esta comandada pelo microcomputador.

As primeiras idéias para este trabalho foram tiradas do artigo com referência (1) na bibliografia; onde a tensão de armadura é controlada com o uso de chopper, e não é feita limitação da corrente de armadura.

No caso do nosso trabalho, a tensão aplicada na armadura é conseguida com o uso de um retificador monofásico com a en-

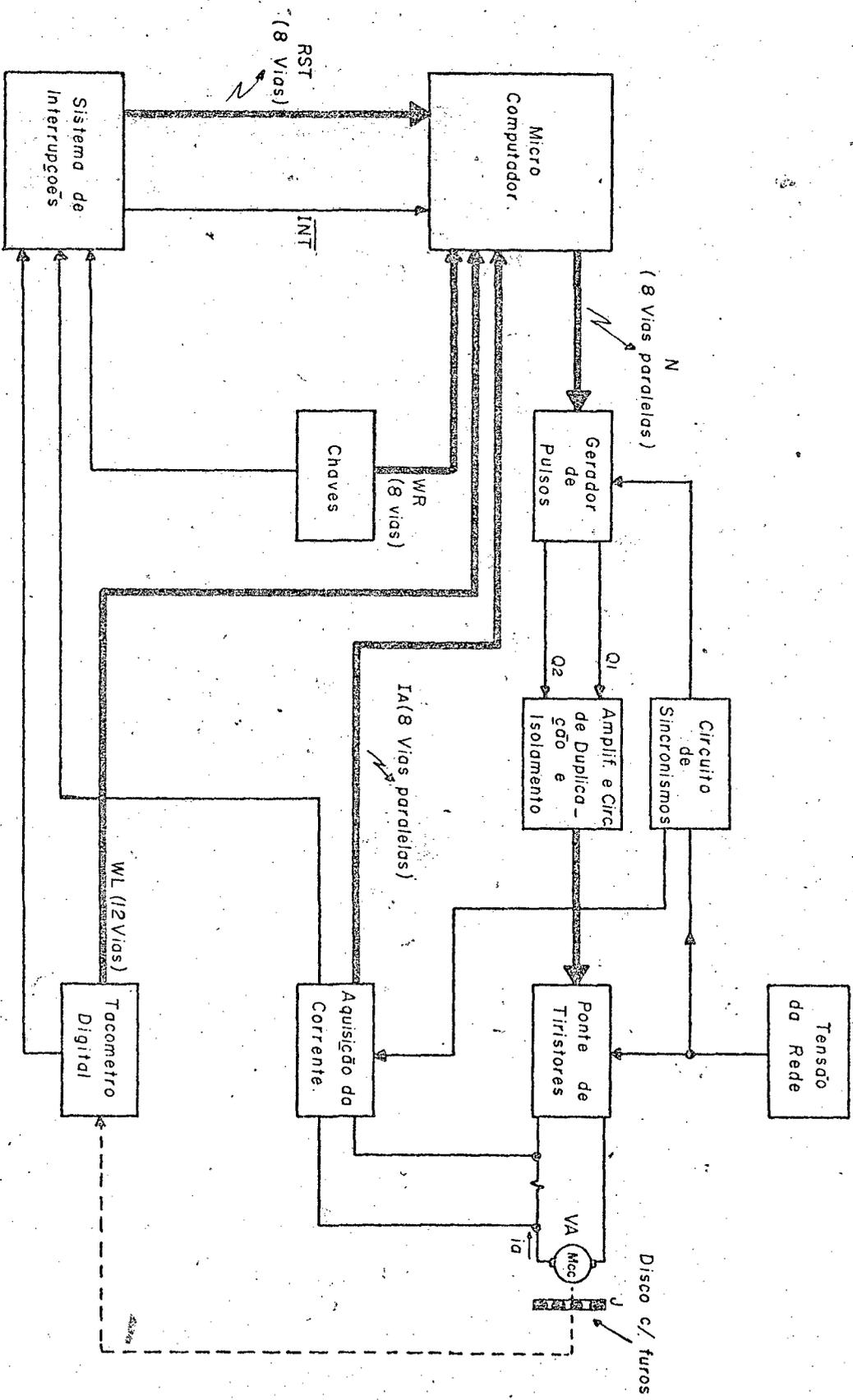


Fig. 1 Diagrama do Sistema.

trada ligada na tensão da rede, que apresenta vantagens devido a sua simplicidade. Outra vantagem é que a experiência adquirida com esta montagem (retificador monofásico), será utilizada em trabalhos futuros, com montagens reversíveis usando retificadores (controle também por microcomputador).

Para a utilização da máquina em toda a sua potencialidade foi previsto uma limitação de corrente.

São três as funções básicas a serem executadas em sistemas deste tipo:

- A transformação de uma informação digital em um sinal analógico a ser aplicado na armadura da máquina.
- As medidas que comporão a realimentação do processo.
- A parte de programação, com o algoritmo de controle que compara as medidas obtidas com as referências, ajustando de acordo com os erros a informação digital na saída do microcomputador.

Na figura 1 é mostrado o diagrama em blocos que descreve o sistema. Na parte de acionamento, uma ponte retificadora com tiristores, alimentada na sua entrada com a tensão da rede, é responsável pela tensão controlada aplicada na armadura da máquina. Este tipo de arranjo facilita bastante o uso de circuitos digitais para a sua operação. Na figura 2 são apresentadas a tensão da rede e a tensão aplicada na armadura para um certo ângulo  $\theta$  de disparo dos tiristores da ponte.

O gerador de pulsos é o circuito que possibilita o controle do ângulo de disparo dos tiristores pelo processador. Os dois trens de pulsos que saem deste circuito são amplificados e isolados tornando compatíveis os circuitos de baixa e alta potência.

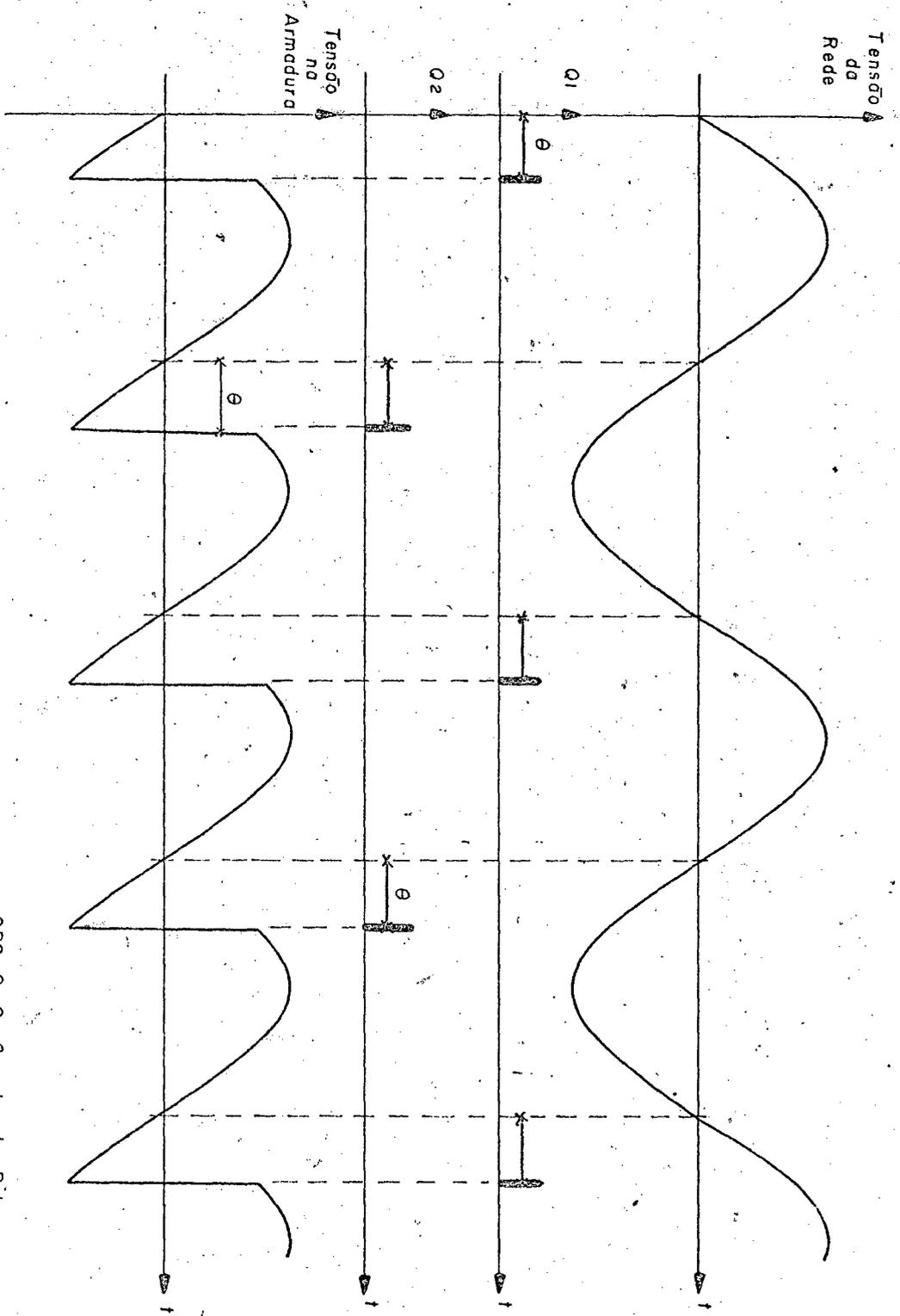


Fig. 2

OBS. Q1, Q2: Gerador de Pulsos.

Com a tensão da rede alimentando a entrada da ponte de tiristores, no controle dos ângulos de disparo as operações do gerador de pulsos e outros circuitos devem necessariamente, estar sincronizados com a tensão da rede. Para isto existe o circuito de sincronismo. Na parte referente as medidas, são realizadas a da velocidade e a da corrente. A medida da velocidade é feita com o uso de um tacômetro digital, constituído de duas partes, a primeira responsável pela obtenção de um trem de pulsos e a segunda pela contagem de pulsos.

Na aquisição da corrente de armadura foi utilizado um conversor A/D, o que inclusive possibilita uma maior flexibilidade de no tratamento do problema da corrente de armadura. A operação do sistema é bastante facilitada com o uso de um jogo de chaves que será responsável pela introdução do microcomputador de velocidade angular de referência.

A obtenção dos dados é feita com o uso de interrupções e para isto é necessário a classificação das entradas em ordem de prioridades, o que é feito no bloco sistema de interrupções do diagrama.

Um microcomputador MCS-80 (SDK-80) da INTEL é usado na implementação deste sistema de controle por amostragem. Nesta aplicação a preocupação maior sempre foi a de melhor utilização do microcomputador, aproveitando a sua potencialidade para a definição dos blocos restantes do sistema. Aspectos ligados a microcomputadores, bibliografia utilizada neste trabalho e maneiras diferentes de realização das funções dos blocos no diagrama são discutidos no capítulo referente a generalidades. As descrições e implementação das funções de cada bloco deste diagrama e a parte referente ao "software" do sistema são apresentados no capítulo II. Resultados e conclusões são apresentados em capítulos subsequentes.

## CAPÍTULO II

### GENERALIDADES

#### 2.1. - Microcomputador

##### 2.1.a. - Considerações gerais

Todos os computadores são basicamente iguais. A diferença entre eles reside apenas no tamanho e velocidade da memória, na eficiência com a qual o dado pode ser movido ou combinado para determinada aplicação, e restrições particulares a interfaces de dispositivos de entrada e saída.

Terminologias como microcomputadores, minicomputadores, etc..., já não são tão claras para a separação das máquinas em classes. Isto pode ser verificado pelo fato de alguns minicomputadores se comportarem como computadores convencionalmente chamados de grandes. E por outro lado, alguns microcomputadores se aproximam de minicomputadores mais simples. O único fato consistente para classificações é o tamanho (3):

- microcomputadores = 100 cm<sup>3</sup>
- minicomputadores = 0,1 m<sup>3</sup>
- midicomputadores = 1 m<sup>3</sup>
- computador = 10 m<sup>3</sup>
- supercomputador = 100 m<sup>3</sup>

A proliferação de memórias, controladores de interfaces e microprocessadores tem sido grande, tanto em número como em complexidade, graças à tecnologia LSI. Não é incomum periféricos LSI mais complexos (4) que o microprocessador com o qual funciona.

Os microcomputadores apresentam uma evolução constante, graças ao aumento gradativo de mais lógica por área nos

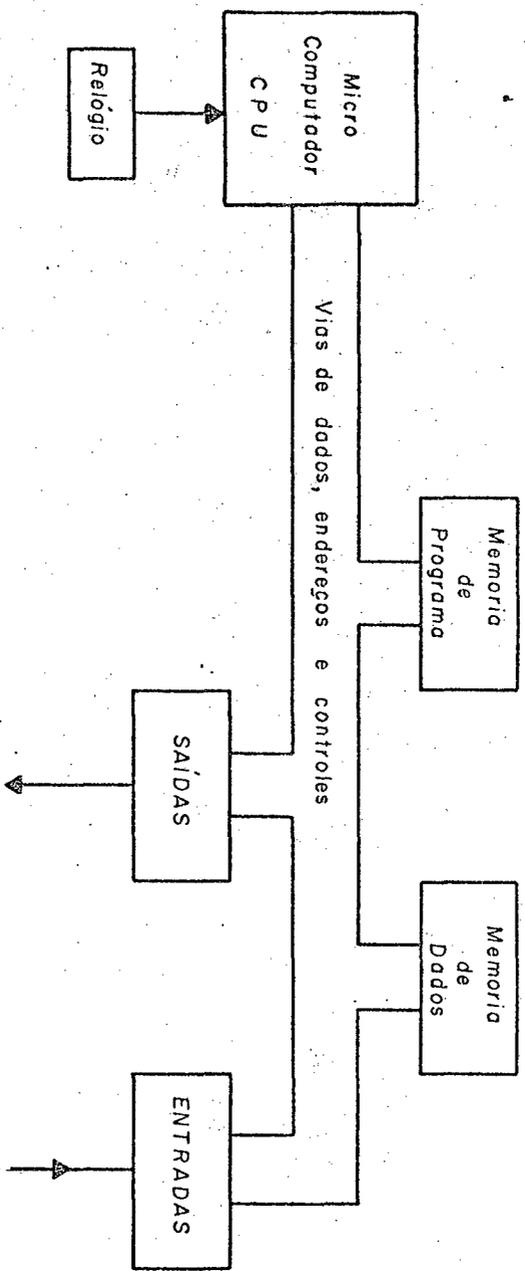


Fig. 3

circuitos integrados. A incorporação em uma simples pastilha da lógica, requerida para produzir uma unidade aritmética lógica (ALU) similar àquelas usadas pelos minicomputadores, que até então eram feitos por centenas de circuitos TTL, possibilitou o aparecimento do microprocessador. Esta ALU unida a uma unidade de controle com suas funções de comando e o respectivo conjunto de instruções, mais registradores de uso geral ou memória local, vias internas, contador de instruções, etc., formam o microprocessador (unidade central de processamento - CPU).

A quantidade e natureza dos circuitos de suporte no microcomputador variam com tipos de microprocessadores, e aplicações, mas normalmente incluem memórias de programa, memórias de dados, dispositivos de entrada e saída e geradores de "clock". A figura 3 mostra uma típica configuração destes sistemas.

#### 2.1.b. - Memórias

Normalmente como memórias, nos microcomputadores as utilizadas são as de "somente leitura" e as de "ler e escrever". Sendo que as memórias de "somente ler" ou memórias fixas são classificadas em ROMs, PROMs e EPROMs.

Em memórias ROM a programação é feita durante a sua fabricação. São utilizadas com programas padrões como controladores de TTY, etc...

A memória "somente leitura" PROM não é programada pelo fabricante, mas sim pelo usuário, com o uso de equipamentos especiais, os "queimadores de PROM" que enviam pulsos de corrente à memória. Estes pulsos irão "queimar" fusíveis internos, gravando as informações desejadas. Nesta memória uma vez gravada a informação, não se pode mudá-la.

Memória EPROM, é um tipo de memória "somente lei-

tura', que mesmo, depois de programada por dispositivos especiais, o seu conteúdo pode ser alterado. A operação de apagar a memória requer que se exponha seus circuitos, através de uma janela existente na pastilha, à luz ultra-violeta. Isto recompõe a memória e possibilita que seja programada novamente.

Estas memórias "somente leitura" por serem fixas, em aplicações dedicadas são chamadas de memórias de programa. São memórias também de acesso aleatório. A faixa de aplicação destas memórias inclui microprogramação, geração de funções, operações aritméticas, etc...

As memórias "ler-escrever" também chamados de acesso aleatório (RAM), são memórias usadas para dados, que variam durante o processamento. São memórias voláteis, ou seja, quando a tensão da alimentação é removida as informações armazenadas são perdidas.

Estas memórias "ler escrever" são divididas em RAMs estáticas e RAMs dinâmicas. A básica diferença entre RAMs dinâmicas e estáticas está na maneira como as informações são armazenadas nas suas células. Nas RAMs estáticas o armazenamento de um bit é feito com flip-flop. Enquanto a RAM dinâmica usa um capacitor interno para armazenar um bit de informação (6) (2).

Nas RAMs dinâmicas ocorrem correntes de fuga nos capacitores, e as cargas devem ser recompostas de tempos em tempos para que não se percam os dados armazenados. A este procedimento se dá o nome de "Refresh".

O "Refresh", então, consiste de um endereçamento cíclico, regenerando as cargas de modo a manter as células com suas informações. A regeneração em uma célula é feita toda vez em que é acessada para ler ou escrever. Ao mesmo tempo todas as

outras células da mesma linha são também regeneradas.

Portanto, a regeneração é feita por linha selecionada da matriz, e isto ocorre numa razão de uma regeneração em um ou dois milissegundos para cada linha. Logo para uma RAM dinâmica de 4K são necessários 64 ciclos (64 linhas X 64 colunas) de memória ou endereçamentos para a regeneração completa desta memória em dois milissegundos. Isto pode ser feito com contadores para guardar o endereço da linha a ser regenerada, relógio que inicia periodicamente o ciclo de regeneração e uma lógica para o uso do endereçamento de "refresh", juntamente com os pedidos normais de leitura ou escrita de dados pelo processador.

As RAMs estáticas são de uso mais fácil não necessitam de "Refresh", portanto, não há necessidade de um "hardware" adicional.

Em termos de potência consumida as RAMs estáticas são as que mais absorvem. Pois seus flip-flops necessitam continuamente de alimentação para conservarem seus estados. Nas memórias dinâmicas o consumo praticamente só ocorre durante a regeneração cíclica ou endereçamento das células.

Quanto ao tamanho, RAMs dinâmicas são menores se comparadas a RAMs estáticas de mesma capacidade de armazenamento. O tamanho da memória dinâmica chega a ser 20% menor que a estática correspondente (6).

Com base nas comparações feitas conclui-se que o emprego de RAM dinâmica em grandes sistemas é mais vantajoso que as estáticas, tanto pela dimensões muito menores como pelo consumo também menor. As RAMs estáticas são mais vantajosas em sistemas pequenos, pela simplicidade no seu manejo.

### 2.1.c. - Microprocessador

Os microprocessadores diferem entre si em "Software" e em "Hardware". No "Software" as diferenças estão ligadas ao projeto da unidade de controle (UC).

Entre as diversas partes componentes do processador é a unidade de controle, a encarregada de fornecer as variáveis de comando para os demais subsistemas, variáveis estas que permitem a execução de determinada operação. Essa operação é especificada através da interpretação de um conjunto de bits chamado de palavra de instrução. Uma instrução, portanto, especifica uma operação a ser executada pela máquina. Normalmente uma operação, para ser realizada dentro de uma máquina, requer um conjunto de passos elementares, encadeados entre si, e executados sequencialmente, passos elementares estes que recebem o nome de microoperações. Existe também a possibilidade de execução simultânea de microoperações, que são ditas. microconcorrentes (4).

Com base nestas variáveis e operações a serem realizadas estipula-se o conjunto de instruções que irão compor o repertório da máquina. É estabelecido um código conveniente para cada instrução, de forma a facilitar o projeto da UC.

Existem duas alternativas para unidades de controle. E uma destas alternativas, a chamada unidade de controle fixa, no seu projeto, o primeiro passo após a fixação das operações, da listagem das variáveis de controle e do repertório de instrução, é o de construir uma "carta de microoperações" para cada instrução. Esta "carta de microoperações" ou "carta de tempos" consiste na distribuição de microoperações no tempo, de modo a completar uma instrução. Uma vez concluídas as "cartas de tempos" correspondentes a todas as instruções, a próxima etapa con-

siste em sintetizar as funções desejadas correspondentes às variáveis de controle.

Para sintetizar as funções são utilizados processos clássicos de síntese e minimização de funções booleanas, o que ao mesmo tempo em que permitem grandes simplificações da UC, tornam difícil a inclusão de novas instruções ou a modificação, após o projeto concluído.

Resumindo, a unidade de controle fixo é um circuito ("hard-wired") encarregado dos sinais de controle a partir da instrução corrente.

O outro conceito é a unidade de controle microprogramada. Neste tipo de unidade de controle as instruções são encaradas como programas que interligam as microoperações. Previamente é estabelecido um conjunto de microinstruções, onde cada microinstrução é codificada de modo que, quando interpretada forneça os valores das variáveis de controle de sua microoperação correspondente e para cada instrução, também chamada neste caso de macro-instrução, existe um microprograma armazenado na memória de controle na UC, que é lido sequencialmente microinstrução a microinstrução ditando todas as microoperações necessárias para execução da instrução.

As instruções binárias são usadas para endereçar a localização do início dos microprogramas correspondentes na memória de controle, que é do tipo "apenas leitura". O número de microinstruções varia dependendo da complexidade da instrução.

Como exemplo o ciclo da máquina que corresponde a busca de uma instrução na memória principal ("fetch"), pode ser realizada por uma subrotina na memória de controle de posição definida. Esta subrotina pode executar o primeiro ciclo

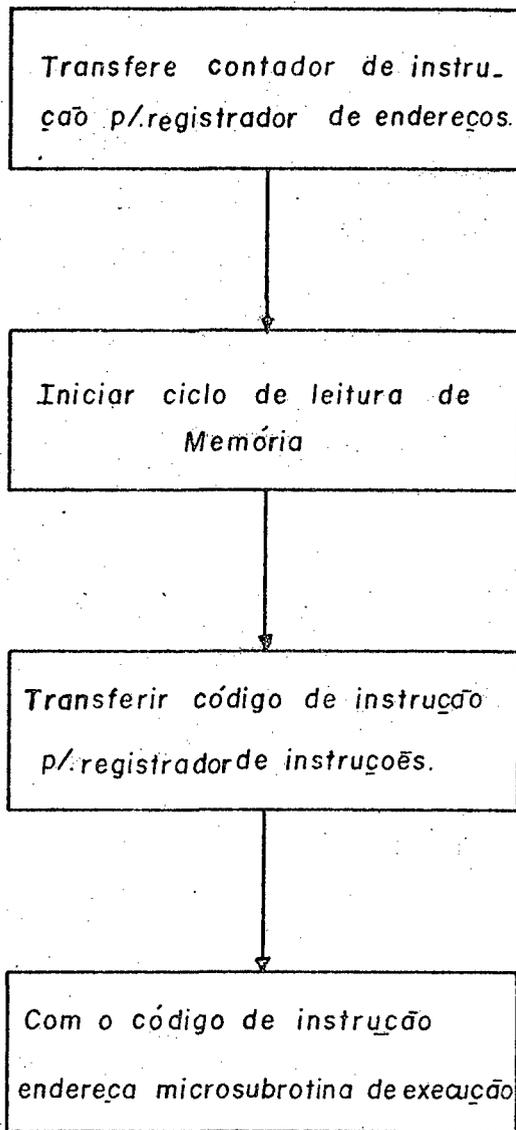


Fig.4 "Fetch"

de máquina de todas as instruções. O fluxograma da figura 4 corresponde a esta rotina.

As unidades de controle microprogramadas, podem ser encaradas como um processador dentro de um processador já que é capaz de realizar programas (microprogramas).

As desvantagens destas UC estão na necessidade da implementação de circuitos mais complexos. Para o projeto não são feitas "cartas de microoperações", e como consequência também não são feitas as minimizações, as funções se apresentam praticamente nas suas formas canônicas. E também no fato de que nas memórias de controle são utilizadas memórias ROM, o que torna lenta a execução de uma macro-instrução.

Mas apesar destas desvantagens, este tipo de UC é das mais utilizadas por ser mais maleável, pois dá a possibilidade de se acrescentar ou modificar instruções.

#### 2.1.d. - Tipos de Microcomputadores:

Com base nas diferenças entre as unidades de controle, partições dos microprocessadores e diferentes tecnologias, os microcomputadores distinguem-se em três grupos básicos (2)(5).

grupo 1 - O microprocessador que se encontra mais frequentemente é aquele usando tecnologias LSI, em que a CPU compõe uma pastilha de circuito integrado. Sua unidade de controle é fixa, ou microprogramada previamente, tendo a memória de apenas leitura dos microprogramas integrada dentro da pastilha. Isto libera o usuário de compromissos com a definição do conjunto de instruções. Microprocessadores MOS, canal N, como o 8080A da INTEL são exemplos deste grupo, usualmente operam com um barramento de dados de 8 bits e requerem um mínimo de periféricos para produzir um microcomputador.

grupo 2 - O microprocessador se apresenta dividido em várias pastilhas. Uma alternativa para estes processadores é a parte de fluxo de dados, que é composta pela unidade aritmética lógica e registradores em uma pastilha e em outra a unidade de controle. A UC ficando separada do fluxo de dados, abre a possibilidade de uso de PROMs para as memórias de microprogramas, podendo com isto o usuário adicionar alguma instrução ou copiar conjunto de instruções de algum processador. Este tipo de processador permite que na associação de vários circuitos de fluxo de dados, tenha-se microprocessadores de largura de palavra variável.

Um exemplo destes microprocessadores são os formados com o SB0400. Este integrado é feito com tecnologia bipolar LSI, que é a lógica de injeção integrada ( $i^2l$ ), apresenta baixo consumo e é um componente rápido. Para completar uma CPU é necessário só um sequenciador e uma ROM com os microprogramas, já que as microinstruções são decodificadas internamente no SB0400. Esta ROM deve ser dirigida por um circuito chamado sequenciador, tal que dada uma instrução este produz uma sequência fixa de micro-instruções. Com o uso de técnicas de microprogramação SB0400 pode ser aplicado para simular microprocessadores com palavra de 4, 8 ou 16 bits, aproveitando o "software" de apoio destes processadores.

grupo 3 - Este grupo é formado por elementos que apresentam o processador, memórias RAM e ROM e os controladores de interface em uma única pastilha de circuito integrado. Isto corresponde na verdade a um microcomputador completo em uma pastilha. São aplicáveis em pequenos sistemas, pois apresentam limitação de memória.

Um exemplo de um microcomputador em uma pastilha são os que compõe a série TMS 1000 da Texas Instrument. Este microcomputador é feito com tecnologia MOS canal P, com palavra de 4 bits, e a ROM é gravada para uma aplicação dedicada durante o processo de fabricação definindo uma única versão de microcomputador TMS 1000 para um determinado consumidor.

A escolha de um microprocessador vai depender da aplicação a que se destina. E considerações como preço, flexibilidade, se é ou não microprogramável, desempenho, tempo para desenvolvimento do sistema, etc..., devem ser levadas em conta. Se o microprocessador apresenta um conjunto fixo de instruções, todo um sistema de compiladores, montadores, editores podem ser disponíveis na tarefa de desenvolvimento de programas. Se for microprogramável o conjunto de instruções de um processador fixo pode ser microprogramado e com isto terá também disponível todo um sistema de "software" de apoio.

## 2.2. - Sistema MCS-80 da INTEL

Este microcomputador é um sistema simples que reúne em uma placa com circuitos impressos todos os componentes necessários tanto em "hardware" como em "software" para um bom funcionamento nas mais diversas aplicações.

De uma maneira geral o sistema MCS-80 tem sua configuração, conforme a figura apresentada no apêndice.

O microcomputador pode ser dividido em três blocos (8) (9).

- Unidade Central - Contém o microprocessador, a parte geradora dos "clocks" necessários e circuitos de interfaces com os barramentos.

- Memórias - As memórias que constituem o siste

ma são memórias ROM, com 1k byte para o programa monitor que controla a comunicação com um dispositivo de entrada e saída, memórias EPROM para programas específicos do usuário, e memórias de dados RAM.

- Entrada/Saída - Contém componentes da família do 8080 e circuitos que podem ser usados nas interfaces para terminais de vídeo, teclados, TTY, etc..., e também em aplicações como interfaces para motores, sensores de temperaturas, etc...

Estes três blocos se comunicam entre si por meio de três vias:

- barramento de dados - É uma via com oito linhas paralelas, bidirecional que liga a unidade central às memórias ou à Entrada/Saída.

- barramento de endereços - São 16 bits paralelos de linhas unidirecionais usadas na identificação de uma localização de memória ou de dispositivos de entrada e saída.

- barramento de controle - É um conjunto de linhas que notifica aos periféricos que tipo de atividade será executada. Estas atividades são leitura de memória representada pelo sinal  $\overline{\text{MEMR}}$  (ativo em nível zero), enviar dados a memória com a linha de controle correspondente  $\overline{\text{MEMW}}$ , leitura em dispositivos de entrada e saída (linha de controle  $\overline{\text{I/OR}}$ ), saída de dados em dispositivos de entrada/saída ( $\overline{\text{I/OW}}$ ). O sinal  $\overline{\text{INTA}}$  serve para se manipular com interrupções. Portanto, são 5 os sinais de controle.

### 2.2.a. - Memórias

O sistema MCS-80 pode acomodar de 1K a 4K bytes de memórias "apenas leitura", onde 1K byte é ocupado pelo monitor, em endereços de 0000H a 03FFH (8)..

Estas memórias podem ser EPROMs do tipo 8708, ou

8308, que são memórias MOS canal N, com organização 1024 X 8 (1024 posições de células, com oito bits cada uma, o que corresponde a 1K byte), o tempo de acesso é de 450 ns. Para acessar uma posição de memória são necessários os 10 bits menos significativos da barra de endereços. Todas as entradas e saídas são compatíveis com a lógica TTL. Nestas pastilhas as saídas apresentam a propriedade de alta impedância (terceiro estado), quando não selecionadas.

Para a seleção destas memórias de apenas leitura é usado um decodificador binário 8205, que com uma combinação de 3 bits de entrada ( $A_0, A_1, A_2$ ) é selecionada uma das oito saídas ( $0_1 - 0_8$ ), mas esta seleção só se dará com a combinação de também 3 sinais que controlam esta pastilha ( $E_1, E_2, E_3$ ).

Com este decodificador quatro pastilhas EPROMs 8708 podem ser selecionadas com as combinações das entradas  $A_0, A_1, A_2$ , que serão bits tirados da barra de endereços. Sendo as memórias de "apenas leitura", deve o decodificador usar dos sinais de controle o sinal  $\overline{MEMR}$  (leitura de memória) na lógica de controle do decodificador ( $E_1, E_2, E_3$ ).

É importante lembrar que sobram quatro linhas do decodificador que podem ser utilizadas em extensões de memórias "apenas leitura".

Para memórias de dados do sistema são usadas memórias RAMs estáticas 8111. São memórias MOS canal N, com organização de 256 posições de células com quatro bits (256 X 4). Para completar 8 bits são necessárias duas pastilhas usando uma mesma via de seleção. Na placa a capacidade máxima é de 1K bytes de RAMs, o que garante oito pastilhas da RAM 8111, com duas a duas usando a mesma via de um decodificador que no caso também é um

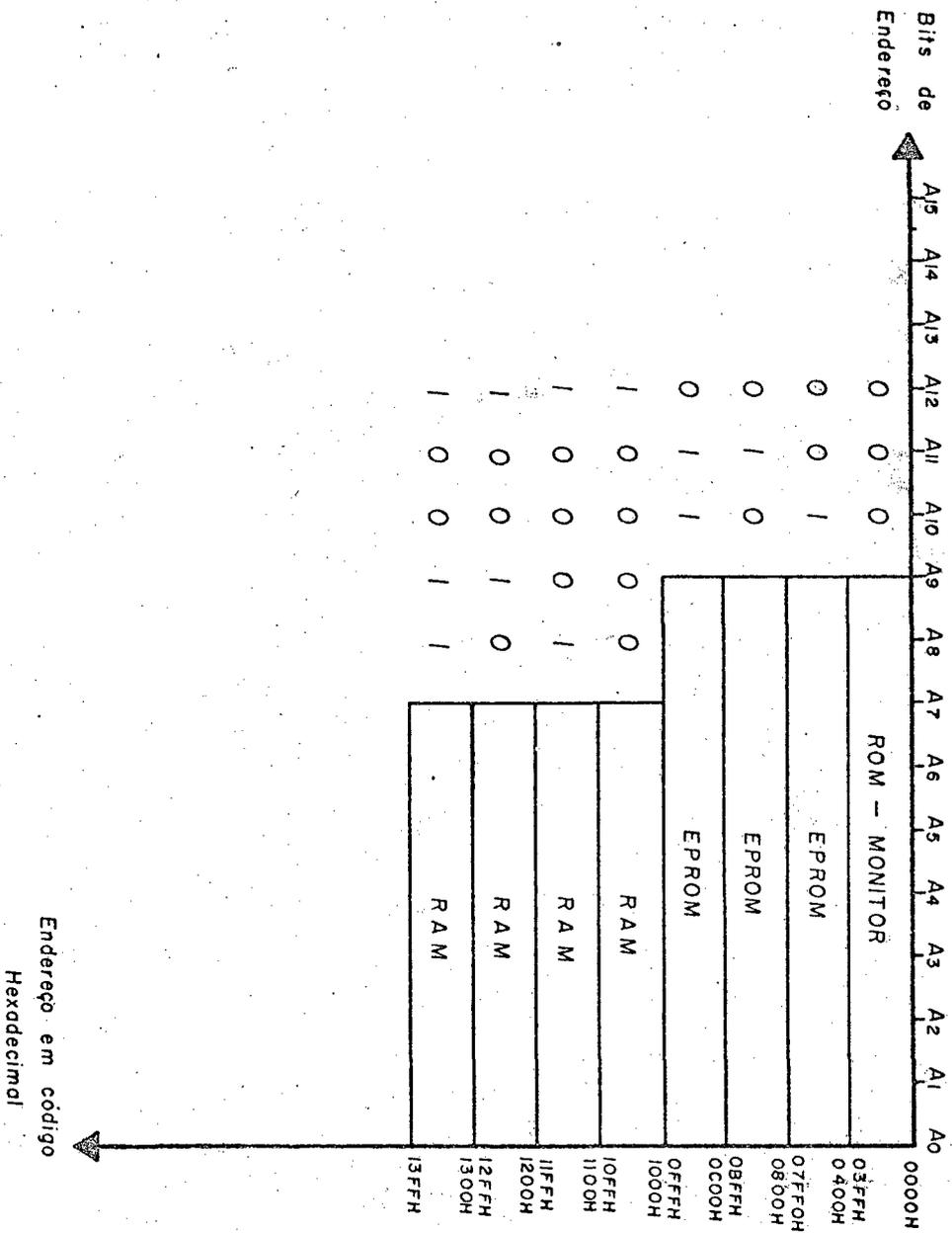


Fig.5

8205. Estas memórias apresentam tempos de acesso na ordem de 850ns, os dados tanto em entradas como em saídas usam as mesmas vias na pastilha, todas entradas e saídas são compatíveis com lógica TTL, apresentam terceiro estado. Os dois sinais de barra de controle  $\overline{\text{MEM R}}$  e  $\overline{\text{MEM W}}$  são utilizados nas lógicas de acesso das células destas memórias.

A distribuição das memórias segundo seus endereços ficou conforme a figura 5.

### 2.2.b. - Entradas/Saídas

Em controladores de interface, o MCS-80 utiliza como interface de comunicação para dispositivos de entrada e saída uma USART (9) ("UNIVERSAL"/SYNCHRONOUS/"ASYNCHRONOUS RECEIVER" "TRANSMITTER") que é o 8251. Esta unidade recebe um caráter em forma paralela da barra de dados e transmite em série para o dispositivo usado para entrada e saída, e no sentido inverso recebe em série os dados do dispositivo e transmite em paralelo para o processador.

A velocidade ou "baud rate" com que estes dados são transmitidos são controlados pelo gerador de "baud rate", que no projeto apresenta diversas alternativas de velocidades a partir de um oscilador à cristal e divisores de frequência obtidos a partir de contadores TTL.

A USART sinaliza ao processador sempre que pode aceitar um novo caráter para transmissão ou sempre que recebe um caráter para o microprocessador. A figura apresentando os blocos funcionais desta unidade está incluída no Apêndice.

Este dispositivo apresenta um "buffer" (isolador) bi-direcional com terceiro estado (alta impedância) na interface com o barramento de dados do MCS-80. O dado é recebido ou trans-

mitido com a execução de instruções IN ou OUT respectivamente. Palavras de controle, palavras de comando, e informações de "Status" (estado atual do USART) são também transferidas através do "buffer".

Os sinais que controlam todas as operações deste dispositivo são gerados pelo bloco funcional "Read/Write Control logic", a partir dos sinais das barras de endereço e de controle do sistema MCS-80. Este bloco contém registradores para armazenar as palavras de controle e comando que determinam o funcionamento da USART.

O dispositivo apresenta um "buffer" receptor que aceita dados em série e os converte em formato paralelo. Este "buffer" é controlado em suas operações pelo bloco de controle de recepção.

A transmissão também apresenta um "buffer" que aceita dados em formato paralelo da barra interna de dados, converte em formato série, inserindo bits para segurança (bit de "start" e bits de "stop") de transmissão. O bloco de controle de transmissão maneja todas as atividades associadas com a transmissão de dados.

O modo de operação deste dispositivo é programado pelo sistema por "software", definindo a forma de comunicação que se deseja. Esta programação é feita usando duas palavras de controle, a primeira palavra de MODO que define a forma geral de operação do 8251. Segue sempre uma operação de "reset". Esta palavra define se a transmissão será síncrona ou assíncrona, define o fator de "baud rate" usados na transmissão e recepção (fator que multiplica o "baud rate" usado), define o comprimento em bits do caráter a ser transmitido ou recebido, define se serão usa-

dos testes de paridade na transmissão e o número de bits de "stop" utilizados. A segunda palavra de controle é a de comando que define se serão permitidas a transmissão e recepção ou somente uma delas, define se serão zerados ou não os "flags" de erros de transmissão. Sinais usados para definição do MODEM que é outro bloco desta unidade também são controlados na palavra de comando.

No programa monitor do MCS-80 a palavra de MODO está definida para 2 bits de "stop", a paridade é desativada, caracteres de 8 bits, fator de "baud rate" igual a 64 e modo assíncrono. Na palavra de comando a transmissão e recepção estão ativas.

Para técnicas de "polling", o processador pode se utilizar da leitura de "status" que informa o estado atual da unidade.

As informações contidas numa palavra de "status" são: se a unidade está pronta para transmitir ou receber dados, se está vazia, se ocorreram erros de paridade com perda de bits do caráter, erros de "overrun" quanto há superposição de caracteres, e erro de "framing" quando um "stop bit" não é detectado.

O programa monitor do sistema foi todo desenvolvido para esta unidade de comunicação, no interface com um dispositivo de entrada e saída para dados transferidos em série.

Os outros dispositivos na placa do MCS-80, para aplicações de entrada e saída são os chamados interfaces programáveis de periféricos (P.P.I)8255. Cada unidade apresenta vinte e quatro pinos que são divididos em três portas de oito linhas paralelas, porta A, porta B e porta C. A figura com os blocos funcionais desta unidade está incluído no Apêndice. Para interface com o

barramento de dados do microcomputador, esta unidade apresenta um "buffer" bi-direcional com terceiro estado. Dados, palavras de controle e informações de "status" são transferidas através deste "buffer" executando instruções IN e OUT do microprocessador 8080. O bloco que maneja com todas as operações internas e transferências de dados, é o bloco "Read/Write Control logic". Recebe sinais dos barramentos de controle e de endereço, e transfere para o grupo de controle das portas os comandos exigidos.

Existem dois blocos que correspondem ao controle do grupo A e ao controle do grupo B. O primeiro é responsável pela porta A e quatro bits superiores da porta C, e o segundo pela porta B e os quatro bits inferiores da porta C. O MODO de operação de cada porta é programada por "software". O microprocessador envia uma palavra de controle à P.P.I. Os blocos de controle recebem sinais de comandos da "Read/Write Control logic", e aceitam as palavras de controle, definindo o comportamento de cada porta, do barramento interno de dados.

Existem três modos básicos de operação da unidade. O chamado MODO 0 em que as portas A, B e C funcionam como entradas ou saídas. Sendo que a porta C pode ter a parte superior programada de maneira diferente da parte inferior. A porta programada como saída apresenta "LATCH" (memórias com flips-flops do tipo D) para manter a informação. No MODO 1 as portas A e B podem ser programadas para entradas ou saídas e a porta C produz e aceita sinais que manuseiam com o fluxo de dados nas outras portas. A porta programada para entrada no MODO 1 possui "LATCH" que é, carregado com valores por um sinal de "STROBE" na porta C correspondente. As saídas também apresentam "LATCH" (memórias) e na porta C são produzidos sinais que podem ser utilizados

pelo dispositivo de entrada/saída. Neste modo de operação o "status" do 8255 pode ser obtido lendo os valores da porta C. No MODO 2 a porta A é usada para entradas e saídas (bi-direcional) e 5 bits da porta C para controle das operações e obtenção de informações de "status". As portas A e B apresentam memórias tanto na entrada como na saída ("LATCH"). Tanto as unidades 8255 como 8251 são compatíveis com lógica TTL.

### 2.2.c. - Unidade Central

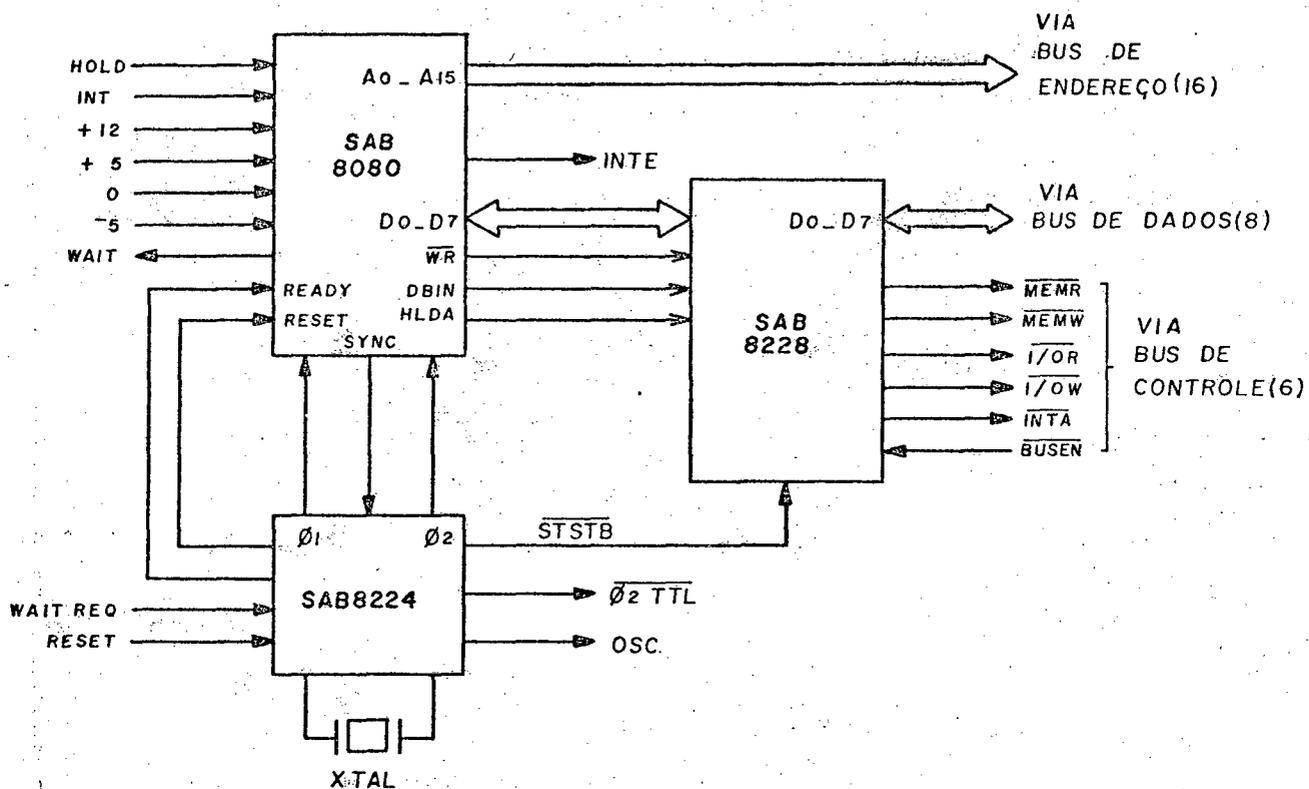
A Unidade Central, mostrada na figura 6 é constituída por (8) (9):

#### - Gerador de "Clocks" 8224

O sistema MCS-80 apresenta um gerador de "clocks" 8224, bipolar, que com um cristal de 18,432 MHz produz os "clocks"  $\phi_1$  e  $\phi_2$  responsáveis pelo "timing" do sistema, ambos com 2,048 MHz apresentam-se em níveis compatíveis com lógica TTL. Este gerador de "clock" contém um oscilador controlado a cristal, um divisor por nove gerando as frequências de  $\phi_1$  e  $\phi_2$  e algumas funções auxiliares. Os pedidos de "Reset" ao sistema e pedidos de "Wait" por algum periférico ao processador são sincronizados com  $\phi_2$  neste gerador de "clocks", antes da aplicação no processador e no restante do sistema. Também o sinal "Sync" é sincronizado só que com  $\phi_1$ , produzindo o sinal  $\overline{STSTB}$  ( $\overline{STATUS\ STROBE}$ ) que é utilizado para armazenar o "status" do ciclo de máquina nas memórias de "status" no sistema controlador 8228.

#### - Controlador de sistema e "driver" de barramento - 8228

Esta unidade é constituída por um "driver" bi-direcional que aumenta em muito o "FAN-OUT" da barra de dados. Esta unidade também é constituída por memórias chamadas "STATUS



Unidade Central

Fig.6

LATCH" que com o sinal  $\overline{STSTB}$ , armazenam o "status" do ciclo da máquina que está sendo executado pelo microprocessador. Estes bits de "status" que são armazenados, quando combinados com sinais vindos do processador (HLDA, DBIN, WR) formam os sinais do barramento de controle do sistema. Os bits de "status", os ciclos de máquina e os sinais de controle serão discutidos na descrição funcional do microprocessador 8080 A.

#### - Microprocessador 8080 A.

O microprocessador 8080 A é uma pastilha LSI usando tecnologia MOS canal N com palavra de 8 bits. Apresenta 16 bits de saída para endereços, sinais de controle e entradas para dois "clocks"  $\phi_1$  e  $\phi_2$ . Excluindo as entradas de "clock", todos os outros sinais apresentam total compatibilidade com lógica TTL.

### 2.3. - Descrição do diagrama funcional da CPU 8080

O 8080 é constituído de três unidades funcionais distintas que são registradores e lógica de endereços, unidade aritmética lógica e unidade de controle. A figura 7 mostra o diagrama funcional da 8080.

#### 2.3.a. - Registradores e lógica de endereços, ou memórias do 8080

Este bloco é constituído por seis registradores de 16 bits, organizados da seguinte forma: 6 registradores de uso geral de 8 bits (registradores B, C, D, E, H e L), 2 registradores temporários (W, Z) de 8 bits cada, um registrador para contador de instrução de 2 bytes e um registrador de 16 bits também, para ponteiro de pilha.

Pilha ou "Stack" é uma organização de memória de dados com instruções especiais para administrá-la. Nesta memória o último dado armazenado é o primeiro a ser retirado. A pilha fa

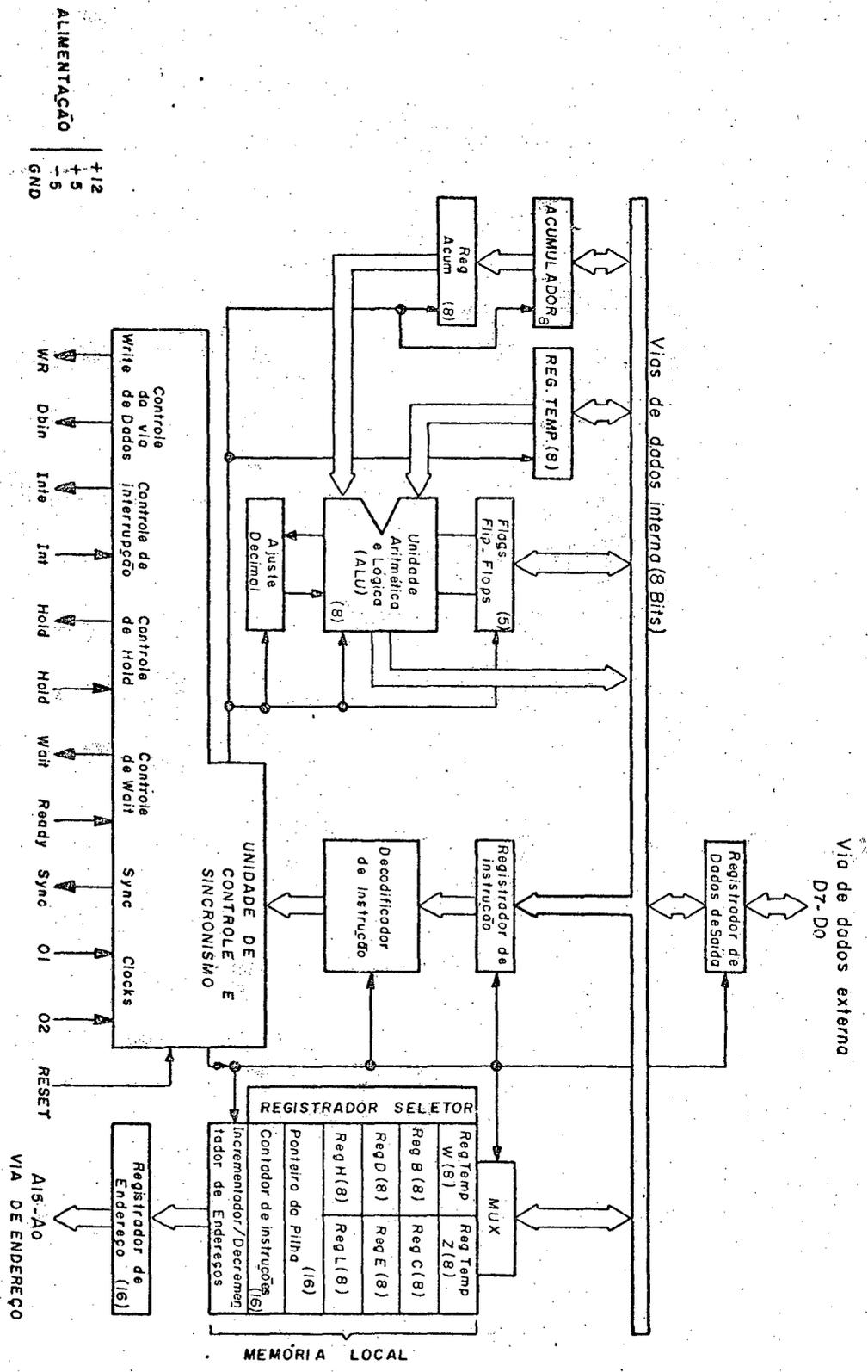


Diagrama funcional da UCP

Fig. 7

cilita muito o problema de desvios para subrotinas e desvios por interrupções, pois são estas memórias em que o endereço de retorno ao programa principal e o conteúdo de algum registrador que se pretenda preservar são guardados. O registrador ponteiro de pilha indica sempre o endereço do início da pilha. A pilha do MCS-80 é programada para o fim das memórias RAM. Os dados são sempre armazenados em 2 bytes. Quando se armazena na pilha um dado o ponteiro de pilha é decrementado de duas unidades, em caso contrário se for retirado 2 bytes de informação o ponteiro é incrementado de 2 unidades. Desta forma a pilha cresce para baixo.

O contador de instruções indica sempre a próxima instrução a ser executada no programa, e durante cada fase de busca de instrução para ser executada o contador de instrução é incrementado.

Os seis registradores de uso geral (B, C, D, E, H e L) podem ser endereçados aos pares para dados de dois bytes, formando os pares B (B e C), D (D e E) e H (H e L), ou serem utilizados separadamente para dados de um byte. Os registradores temporários são utilizados pela unidade de controle durante execução das instruções, não são acessíveis ao programador.

A comunicação entre memória local e a via interna de dados é feita pelo multiplexador, em blocos de 8 bits.

A manipulação de dados com 2 bytes só ocorre na memória local e no incrementador/decrementador de endereços.

O incrementador/decrementador de endereços é usado para incrementar o contador de instruções ou qualquer par de registradores.

O registrador de endereços recebe dados de algum dos 3 pares de registradores ou do contador de instrução, tornan

do com isto disponível na barra de endereços do microcomputador. A figura 8 mostra esta unidade funcional do 8080.

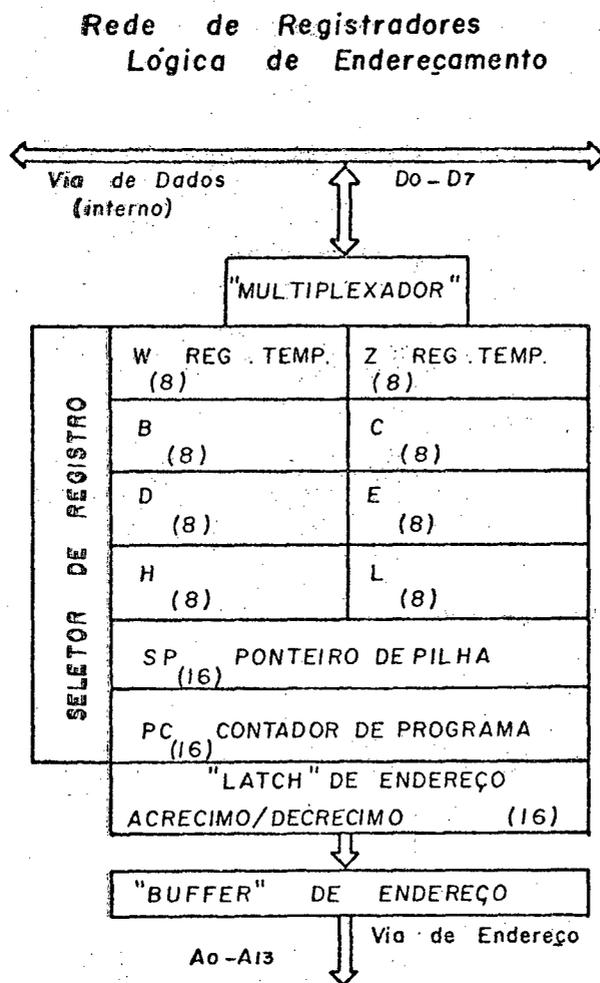


fig. 8

### 2.3.b. - Unidade aritmética lógica

Na unidade aritmética lógica são realizadas todas as operações lógicas aritméticas e de deslocamento de bits para direita ou esquerda. A ALU contém os seguintes registradores: o ACUMULADOR, o REG. ACUM. (ou "LATCH" do acumulador), e o REG TEMP (registrador temporário), todos estes registradores são de 8 bits. A ALU contém ainda o registrador de "flags" de 5 bits que são: bit de zero (Z), "vai um" (CY), sinal (S), paridade (P) e "vai um" auxiliar (AC). O conteúdo destes bits é altera

do em função dos resultados das operações. As instruções de pulo condicional utilizam o registrador de "flag" para tornar a decisão requerida.

A ALU é alimentada pelo registrador temporário e pelo acumulador temporário e o flip flop de carry ("vai um") do registrador de "flags". O resultado da operação é transferida pela via interna de dados para o acumulador, com este resultado os "flags" podem ser alterados.

O ajuste decimal é utilizado durante a execução da instrução DAA, que ajusta o acumulador em dois dígitos decimais. A ALU do 8080 é apresentada na figura 9.

#### Unidade Lógica - Aritimética

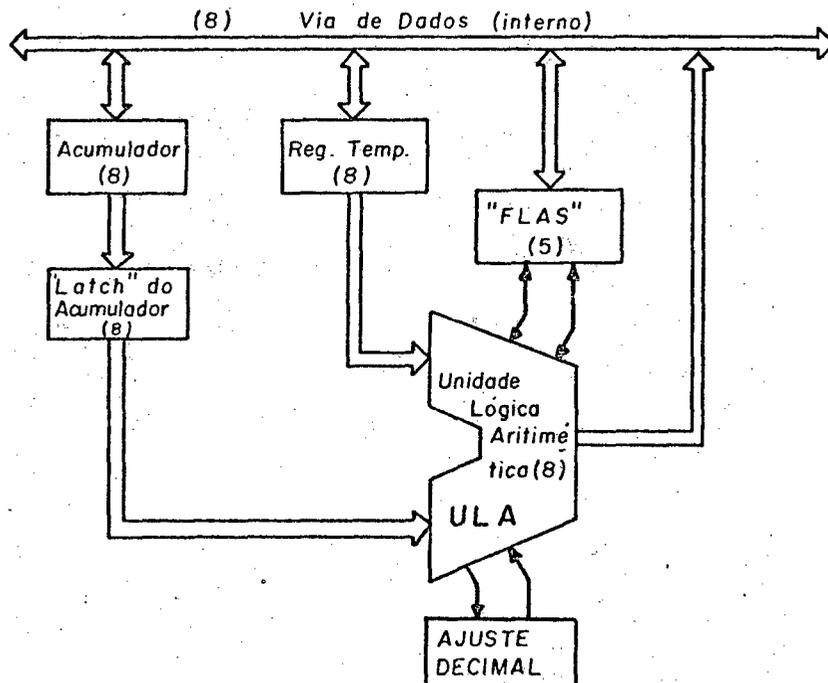


fig. 9

### 2.3.c. - Unidade de controle

A unidade de controle contém o registrador de instrução que durante um "fetch" armazena o código de operação da instrução. Este código de operação quando armazenado no registrador de instrução, fica disponível no decodificador de instruções.

A saída do decodificador alimenta a seção de controle e sincronismo que combinados com sinais externos ( $\phi_1$  e  $\phi_2$ , etc...) produzem os sinais que controlam a ALU, o fluxo de dados e registradores.

O registrador de dados é constituído de "LATCH" e "buffer" bi-direcional com terceiro estado, isto tudo garante o fluxo de dados entre a barra de dados externa e a via interna de dados. A figura 10 mostra a unidade de controle.

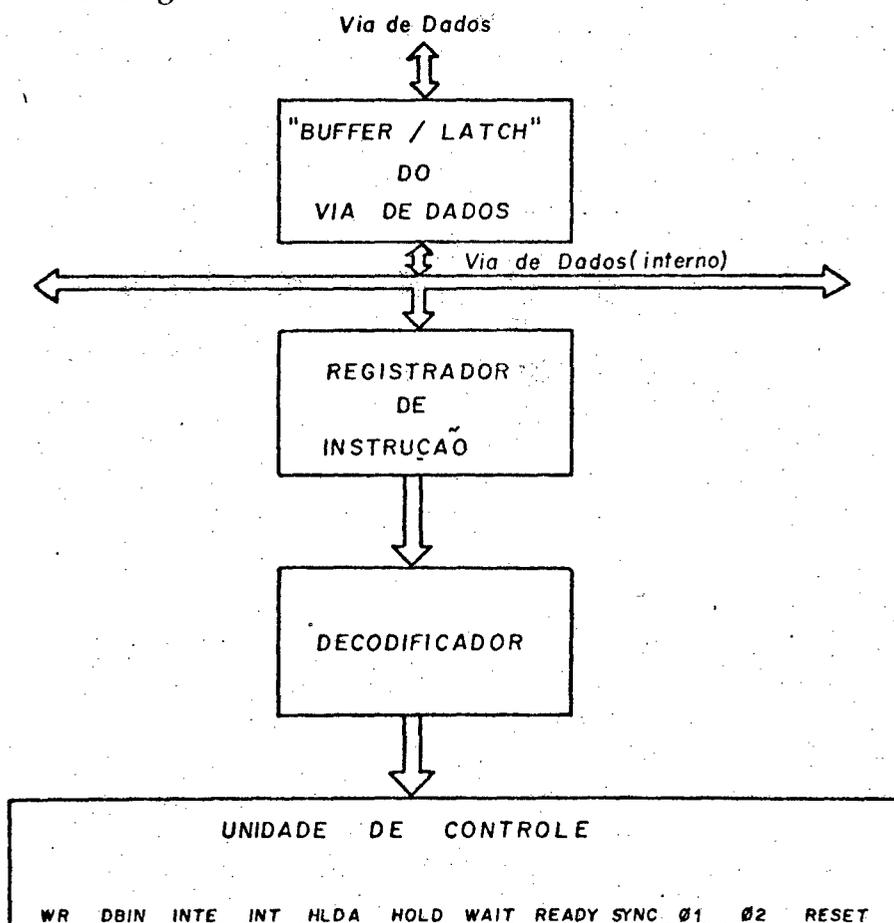


fig. 10

## 2.4. - Funcionamento do microprocessador

### 2.4.a. - Ciclos e tempos do processador (8) (9)

Um ciclo de instrução corresponde ao tempo requerido para retirar a instrução da memória e executá-la.

Um ciclo de máquina corresponde a cada acesso à memória ou portas de entrada e saída pelo microprocessador. Existem instruções de um byte (só o código de operação), de dois e três bytes. Numa instrução, um "fetch", que é a busca do código de operação, corresponde ao primeiro ciclo de máquina, dependendo da instrução ela poderá ter de 1 a 5 ciclos de máquinas. E cada ciclo de máquina é constituído de três a cinco estados.

Um estado é a menor unidade de atividade de processamento e corresponde ao intervalo entre duas subidas consecutivas de  $\phi_1$ , logo o estado corresponde a  $0,5 \mu \text{ seg.}$

O bloco de controle e sincronismos na unidade de controle com os dois "clocks  $\phi_1$  e  $\phi_2$ ", gera os sinais de controle distribuídos nos diversos estados de um ciclo de instrução. E um destes sinais, o sinal de SYNC, que é produzido na subida de  $\phi_2$  no primeiro estado do ciclo de máquina, serve para o processador identificar o início de um novo ciclo de máquina.

Para analisar o "timing" do processador vamos examinar a execução da instrução ADD R que irá somar o conteúdo do registrador R com o do acumulador e o resultado fica no acumulador. Esta instrução é de um byte só, portanto terá só um ciclo de máquina ("fetch"). Pois os operandos se encontram dentro da CPU.

Observando a figura 11, verificamos que no estado  $T_1$  na subida de  $\phi_2$  o contador de instruções tem o seu conteúdo transferido para o registrador de endereços e por consequen-

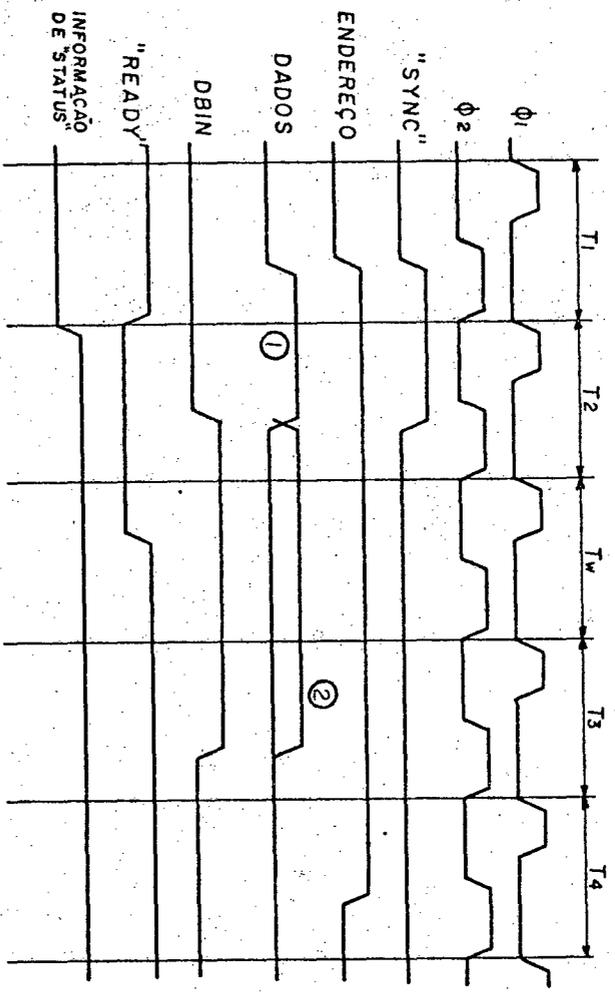


Diagrama de "timing" do ciclo de instrução da instrução ADDR.

Fig. 11

cia fica disponível na via de endereços do microcomputador. Também o sinal de SYNC é produzido pela subida de  $\phi_2$  em  $T_1$ . Quando o sinal SYNC está em nível 1, o microprocessador larga na barra de dados (1) o "status" do ciclo de máquina que está sendo executado. Este "status" será armazenado numa memória no 8228 chamada de "STATUS LATCH", com o sinal  $\overline{STSTB}$  (sinal gerado por SYNC e  $\phi_1$ ), no início de  $T_2$  e permanecerá desta maneira gerando os sinais de controle até o fim do ciclo de máquina.

Em  $T_2$  é verificado se há algum pedido de "Wait" (sinal "ready" em nível 0). Se houvesse no próximo estado a máquina entraria no estado de espera ( $T_w$ ), isto é utilizado para memórias ou periféricos lentos. Em  $T_2$  é verificado se a instrução a ser executada é um "Halt", se fosse o processador pararia (esta instrução paralisa o processador). Também é feito teste em  $T_2$  para verificar se há pedido de HOLD. No caso do exemplo examinado ADDR, em  $T_2$  com o "status" do ciclo de máquina fixado no "STATUS LATCH" se produz o sinal de  $\overline{MEMR}$  que combinado com o endereço da instrução, provoca o acesso à célula de informação.

Na subida de  $\phi_2$  no estado  $T_2$  o sinal de SYNC volta a nível 0, libera a barra de dados, que antes estava com a informação de "status" do ciclo de máquina e é acionado o sinal de DBIN ("DATA BUS IN" - Entrada de dados no microprocessador) com a subida de  $\phi_2$ .

No estado seguinte,  $T_3$ , DBIN é desativado com a subida de  $\phi_2$ , e o dado de entrada (2) deve já estar no microprocessador. Sendo, o primeiro ciclo de máquina de uma instrução, um "fetch", o dado deverá estar armazenado no registrador de instruções em  $T_3$ . Durante  $T_4$  será executada a soma do registrador R com o acumulador e o resultado ficará no acumulador. O

contador de instruções é incrementado durante um "fetch" no incrementador/decrementador de endereços. No Apêndice é apresentada uma tabela com os estados em um ciclo de máquina do 8080.

Uma instrução em que um dos operadores está na memória, é normalmente uma instrução de dois ciclos de máquina. Como exemplo ANAM, que faz um E lógico (AND) com o conteúdo da memória cujo endereço está no par de registradores HL e o conteúdo do acumulador. No primeiro ciclo de máquina desta instrução o código de operação é buscado ("fetch") em três estados ( $T_1$ ,  $T_2$ ,  $T_3$ ). No segundo ciclo de máquina o conteúdo de HL é transferido para via de endereços e o operando é lido na memória em  $T_1$ ,  $T_2$  e  $T_3$  e no último estado deste ciclo  $T_4$  é feita a função lógica E. Este ciclo de máquina será um "MEMORY READ".

Os diferentes ciclos de máquina que podem ocorrer numa instrução são:

- 1) "FETCH" (busca do código de operação)
- 2) "MEMORY READ" (leitura de memória)
- 3) "MEMORY WRITE" (escrita de memória)
- 4) "STACK READ" (leitura de pilha)
- 5) "STACK WRITE" (escrita de pilha)
- 6) "INPUT" (entrada)
- 7) "OUTPUT" (saída)
- 8) "INTERRUPT" (interrupção)
- 9) "HALT" (parado o microprocessador)
- 10) "HALT-INTERRUPT" (parado e interrompido)

O processador torna possível a identificação do ciclo de máquina pela transmissão de 8 bits para via de dados durante o intervalo em que o sinal SYNC está ativo. Esta informação ("status") é armazenada no STATUS LATCH e produz em combina-

ção com os sinais de DBIN,  $\overline{WR}$  e HLDA os sinais da barra de controle.

Os bits de uma palavra de "status" de um ciclo de máquina são descritos a seguir:

BIT DA PALAVRA	SÍMBOLO	DEFINIÇÃO
D <sub>0</sub>	INTA	Bit de reconhecimento de uma interrupção, este bit combinado com o sinal DBIN gera o sinal de controle $\overline{INTA}$ .
D <sub>1</sub>	$\overline{W_0}$	Este bit indica que a operação deste ciclo de máquina será acesso à memória ou à dispositivo E/S pelo processador (ativado em zero).
D <sub>2</sub>	STACK	Indica operações com a pilha.
D <sub>3</sub>	HLTA	Indica que a máquina está em HALT (parada).
D <sub>4</sub>	OUT	Indica operações de saída de dados para dispositivos de E/S. A combinação deste bit ativado e com o $\overline{WR}$ sinal do 8080, gera o sinal de controle $\overline{I/O W}$ e desativado junto com $\overline{WR}$ gera o sinal de $\overline{MEM W}$ .
D <sub>5</sub>	M <sub>1</sub>	Indica que o ciclo é um "FETCH".
D <sub>6</sub>	INP	Indica uma entrada de um dispositivo de E/S. A combinação com DBIN gera o sinal de barra de controle $\overline{I/O W}$ .
D <sub>7</sub>	$\overline{MEM R}$	indica que haverá leitura de memória. A sua combinação com DBIN gera

o sinal  $\overline{\text{MemR}}$  na via de controle.

Baseado nestes bits e seus significados temos a tabela da figura 12 que classifica o ciclo de máquina pelo conteúdo dos bits de palavra de "status" e dá os sinais de controle que são gerados com base nestas informações de "status".

TIPO DE CICLO DE MÁQUINA

		1	2	3	4	5	6	7	8	9	10
D0	INTA	0	0	0	0	0	0	0	1	0	1
D1	$\overline{\text{w0}}$	1	1	0	1	0	1	0	1	1	1
D2	STACK	0	0	0	1	1	0	0	0	0	0
D3	HLTA	0	0	0	0	0	0	0	0	1	1
D4	OUT	0	0	0	0	0	0	1	0	0	0
D5	MI	1	0	0	0	0	0	0	1	0	1
D6	INP	0	0	0	0	0	1	0	0	0	0
D7	MEMR	1	1	0	1	0	0	0	0	1	0

SINAIS DE CONTROLE

INTA  
I/O $\overline{\text{W}}$   
I/O $\overline{\text{R}}$   
MEM $\overline{\text{W}}$   
MEM $\overline{\text{R}}$   
MEM $\overline{\text{W}}$   
MEM $\overline{\text{R}}$

CARTA DE PALAVRAS DE STATUS . FIG 12

#### 2.4.b. - Interrupção

Um dos aspectos positivos do microprocessador 8080 é a sua capacidade de manusear pedidos de interrupções de dispositivos externos. Um dispositivo periférico pode iniciar uma interrupção enviando ao processador um pulso na entrada INT.

Este pulso que ocorre de maneira assíncrona em qualquer estado de um ciclo de instrução, só será armazenado em um flip-flop interno se a saída INTE (interrupção permitida) estiver em nível alto e só ocorre em coincidência com  $\phi_2$  no último

estado da instrução atual, o que garante a sua conclusão antes da interrupção.

Quando isto acontece ao invés de executar um ciclo de busca ("fetch"), iniciará o ciclo de máquina correspondente a interrupção, o sinal INTE assumirá nível zero bloqueando novos pedidos de interrupções.

O ciclo de interrupção tem um comportamento semelhante ao de um "fetch". O dispositivo que solicitou o pedido de interrupção, deve largar na barra de dados, quando aceito o pedido, uma instrução de RST (instrução de reinício) que armazena o conteúdo do contador de instruções na pilha ("STACK") e carrega o contador de instruções com uma posição da memória especificada em um campo de três bits na instrução. O conteúdo do contador de instruções será de oito vezes o valor contido no campo de três bits. Serão portando oito endereços possíveis: 0, 8, 16, 24, 32, 40, 48 e 56. Nestes endereços deverão estar o início ou um desvio para uma subrotina.

Desta maneira haverá condições do processador atender a oito pedidos diferentes de interrupções. No fim de uma subrotina de interrupção uma instrução de RETURN restaura o contador de instruções que estava no "STACK", e o programa principal volta a ser executado. Antes da execução de um RETURN o sinal INTE deve ser ativado novamente para que o processador volte a aceitar novos pedidos de interrupção e isto é conseguido com a execução de uma instrução EI (interrupção permitida).

O sistema MCS-80 no programa monitor só permite um vetor interrupção (só uma subrotina é endereçada nas interrupções). A pastilha 8228 é a encarregada quando no ciclo de interrupção de largar na barra de dados uma instrução RST 7, ga-

rantindo um reinício no endereço 56.

Uma interrupção pode ser encarada como uma chamada de uma subrotina por "HARDWARE". Para entradas assíncronas de dados ou aplicações em tempo real as interrupções são muito uteis.

#### 2.4.c. - "HOLD" (estado suspenso)

Para que um dispositivo periférico realize um acesso direto à memória é necessário o envio de um pulso na entrada HOLD do microprocessador 8080.

Este pedido de "HOLD" que pode ocorrer em qualquer estado de um ciclo de máquina é sincronizado internamente, isto o corre com a subida do clock  $\phi_2$  em qualquer estado.

A combinação deste sinal de HOLD reciclado, com o sinal READY em nível 1 e o clock  $\phi_2$  ativam um flip-flop interno (HOLD F/F). Com este flip-flop ativado a próxima subida do clock  $\phi_1$  leva a saída HLDA para nível 1, entrando o microprocessador em estado de suspensão.

Este pedido de "HOLD" é aceito no início de  $T_3$  (subida de  $\phi_1$ ) se o ciclo de máquina corresponder a uma entrada de dados no processador. Se o ciclo de máquina corresponder ao envio de dados para fora do microprocessador, o pedido só será aceito (HLDA = 1) no estado subsequente a  $T_3$ .

Quando é aceito o "HOLD" as vias de endereço e de dados são liberadas pelo processador, que deverá continuar a conclusão do ciclo de máquina atual, pois em  $T_4$  e  $T_5$  as operações são internas, não sendo mais necessárias entradas e saídas.

O processador deverá permanecer neste estado de suspensão até que o sinal "HOLD" externo caia a nível "0", este sinal também é assíncrono, e deverá ser sincronizado. Com isto "HLDA" assumirá também nível "0" no primeiro pulso de  $\phi_1$  e o pro

cessador reiniciará normalmente ao próximo ciclo de máquina, saindo do estado de alta impedância.

É importante observar que o pedido de "Hold" só será aceito num ciclo de máquina se ocorrer em estados como  $T_1$ ,  $T_2$  ou  $T_w$  com READY ativo. Caso contrário o microprocessador só entrará em alta impedância no próximo ciclo de máquina. Isto garante que as atividades do processador com as vias de endereços e de dados serão completadas no atual ciclo de máquina.

#### 2.4.d. - Instrução Pare (HALT)

A execução da instrução Halt faz com que o microprocessador entre em uma sequência de estados de esperas  $T_w$ , onde não existe processamento. O processador só escapa dessa situação com pedidos de RESET, ou interrupção.

#### 2.5. - Considerações sobre soluções possíveis para o sistema

As maneiras de realização das funções no diagrama da figura 1 são muitas, e estão na dependência de uma maior capacidade para projetos de circuitos ou para desenvolvimento de "software" do projetista. A seguir serão apresentadas algumas soluções lógicas para determinados blocos, com a análise de suas limitações e conveniências.

##### 2.5.a. - Tacômetro

Um problema que pode ser enfrentado com soluções diferentes é a medida de velocidade. Esta medida pode ser feita utilizando dispositivos analógicos, mas a precisão é insuficiente, isto é inerente a tecnologia dos detetores, pois muitos fatores contribuem para resultados não favoráveis, como derivas com a temperatura, queda de tensão nas escovas, degradação da informação na transmissão etc... Além da desvantagem que é a necessidade da transformação de valores analógicos para digitais.

As leituras utilizando tacômetro digitais são mais precisas (10), e a dificuldade maior é a lentidão em que são obtidas. Estes dispositivos são constituídos por um gerador de pulsos, que entrega uma informação sob a forma de um trem de pulsos de frequência proporcional à sua velocidade de rotação. Todo o problema da medida digital de velocidade consiste em converter, em um tempo muito curto esta frequência em um número utilizável para a regulação. Isto se deve à dificuldade em se obter frequências altas nestes trens de pulsos, pois há uma certa incapacidade técnica para se obter altas resoluções por giro da máquina nestes detectores. Por outro lado, o transporte dos sinais a distância implica numa limitação pela faixa passante dos cabos.

A faixa de frequências utilizadas é de:

$$0 < f < 20 \text{ KHz}$$

A conversão frequência/número, pode ser feita com uso do método mais simples, que consiste em contar pulsos sobre uma base de tempos fixa (ou tempo de contagens). Se este tempo de contagens é escolhido de maneira compatível com as faixas de utilização da regulação, os resultados obtidos proporcionarão um bom desempenho ao sistema.

Um outro método de conversão frequência/número é o baseado na observação evidente, de que a conversão mais rápida realizável na prática, tem como base de tempo o próprio período da frequência a ser convertida. A medida da velocidade é então obtida, contando sobre este período, os pulsos de uma frequência de "clock". O número N contado é proporcional ao inverso da velocidade. Este método por apresentar a resolução variável com a velocidade impõe o emprego de uma frequência de relógio muito elevada, isto limita a faixa de utilização na direção das baixas frequên-

cias, uma vez que a capacidade dos contadores é limitada (10).

A contagem de pulsos no tacômetro digital, pode ser feita por "software" no microcomputador usado no processo de controle (1). Mas este procedimento traz um grande número de desvantagens, começando pela dificuldade do aproveitamento do microcomputador no processamento de dados outros que não a velocidade; já que o processador durante todo o tempo de contagens, não pode ser desviado para outras tarefas, sob pena de erros na leitura. Considerando também, que o tempo de processamento do algoritmo de controle é praticamente nulo se comparado ao tempo de contagens; o microprocessador, portanto, será praticamente monopolizado, na aquisição da velocidade. Existe, portanto, uma certa dificuldade na aquisição de valores da corrente de armadura, o que limita o algoritmo a compromissos maiores com os incrementos de tensão que atuam sobre a armadura, ocasionando processos mais lentos de regulação. O uso de algoritmos mais complexos é problemático para montagens com este tacômetro. Uma outra desvantagem deste tipo de tacômetro é a limitação da velocidade da máquina pela velocidade do processador.

Um procedimento muito utilizado em tacômetros digitais com base de tempo fixo é o escalonamento de modo que a cada pulso do codificador ótico acoplado ao eixo da máquina corresponde a 1 rpm, possibilitando com isto, a leitura direta da velocidade nas contagens. Este procedimento deve ser feito com o cuidado de não comprometer o período de amostragem deste sistema discreto.

#### 2.5.b. - Entrada de dados

A obtenção de valores de velocidade de referencia, podem ser feitos por intermédio de chaves em estado de aces

so à memória. Para tanto é necessário gerar um pulso para requisição de "HOLD". Com isto ao reconhecer o pedido o processador no terceiro ou quarto estado do ciclo de máquina atual, libera os barramentos, entrando em estado de alta impedância. São dois os caminhos para a entrada de dados:

- O primeiro seria armazenar em localização de memória os valores das chaves; mas para isto é necessário gerar simultaneamente ao reconhecimento do pedido de "HOLD", os sinais de controle e de endereço das posições que queremos alcançar. Este método, embora rápido, necessita de circuitos adicionais, como um controlador de DMA (Acesso Direto à Memória).

- o segundo método, bem mais simples, seria a leitura direta das chaves colocadas em posição de memória.

Nos dois casos, com o processo de controle em andamento, para qualquer alteração é necessário a paralização do processador, o que não é muito favorável, principalmente no segundo caso, onde o motor de corrente contínua pode ficar alguns minutos sem regulação, que seria o tempo de se acertar nas chaves a nova velocidade de referência.

Outro método para a obtenção da velocidade de referência, seria forçando o processador entrar em estado de espera ( $T_w$ ) pela entrada READY colocada no nível zero (1). Com isto as chaves colocadas em endereço de memória ou em porta de entrada podem ser ajustadas ao novo valor de velocidade de referência. Pelos mesmos motivos apresentados no caso anterior, este método é lento.

Para entradas de dados poderia ser utilizada a técnica de "polling" que é bastante segura. Esta técnica consiste de uma interrogação cíclica das portas de entradas de dados, cons

tituindo-se numa alternativa ao método de interrupções. No caso do sistema proposto pelo diagrama da figura 1 as entradas de velocidade e corrente não estão sincronizadas entre si e as entradas de chaves ocorrem de uma maneira totalmente aleatória. No emprego desta técnica seria necessário cuidados especiais no "software" do sistema para a aquisição de dados. Outra desvantagem deste método é a sua lentidão.

## CAPÍTULO III

### DESCRIÇÃO DO SISTEMA

#### 3.1. - Considerações Gerais:

Na composição deste sistema de controle de velocidade, mostrado na figura 1, além das definições dos blocos e suas funções, foi importante definir o comportamento do microcomputador de modo a simplificar a implementação dos circuitos, de algum "hardware" adicional, e do "software" do sistema. As realizações dos circuitos foram feitas sempre procurando entre o disponível e o barato, o que implicou no uso de circuitos lógicos de tecnologias diferentes.

O sistema mede e corrige a velocidade do motor em tempos discretos. Os valores de corrente também são obtidos de maneira discreta e são utilizados em um limitador de corrente da armadura.

O sistema apresentado na figura 1 pode ser dividido em três partes:

- A parte correspondente ao acionamento da máquina; constituída pelo gerador de pulsos, amplificadores e isoladores, e a ponte de tiristores.

- A parte referente a entrada de dados; formada pelo tacômetro digital, bloco de aquisição de corrente e o painel de chaves responsáveis pelo envio da velocidade de referência.

- O "software" do sistema; constituído pelas rotinas de entrada de dados e algoritmo de controle.

#### 3.2. - Gerador de pulsos:

##### 3.2.a. - Descrição do bloco

O gerador de pulsos é o responsável pela produ

ção dos pulsos de disparos, possibilitando o controle da ponte de tiristores pelo processador. Como entrada apresenta um número digital  $N$  em oito vias paralelas, enviado pelo microcomputador, e como saídas dois trens de pulsos, ambos com frequência de 60 Hz defasados de  $180^\circ$ , com o primeiro atuando no disparo dos tiristores da ponte que correspondem ao semiciclo positivo da tensão da rede, e o outro aos tiristores que conduzem no semiciclo negativo.

O ângulo de disparo  $\theta$  (figura 13), a partir do zero da tensão da rede até a condução na ponte, está relacionado com a informação digital  $N$ .

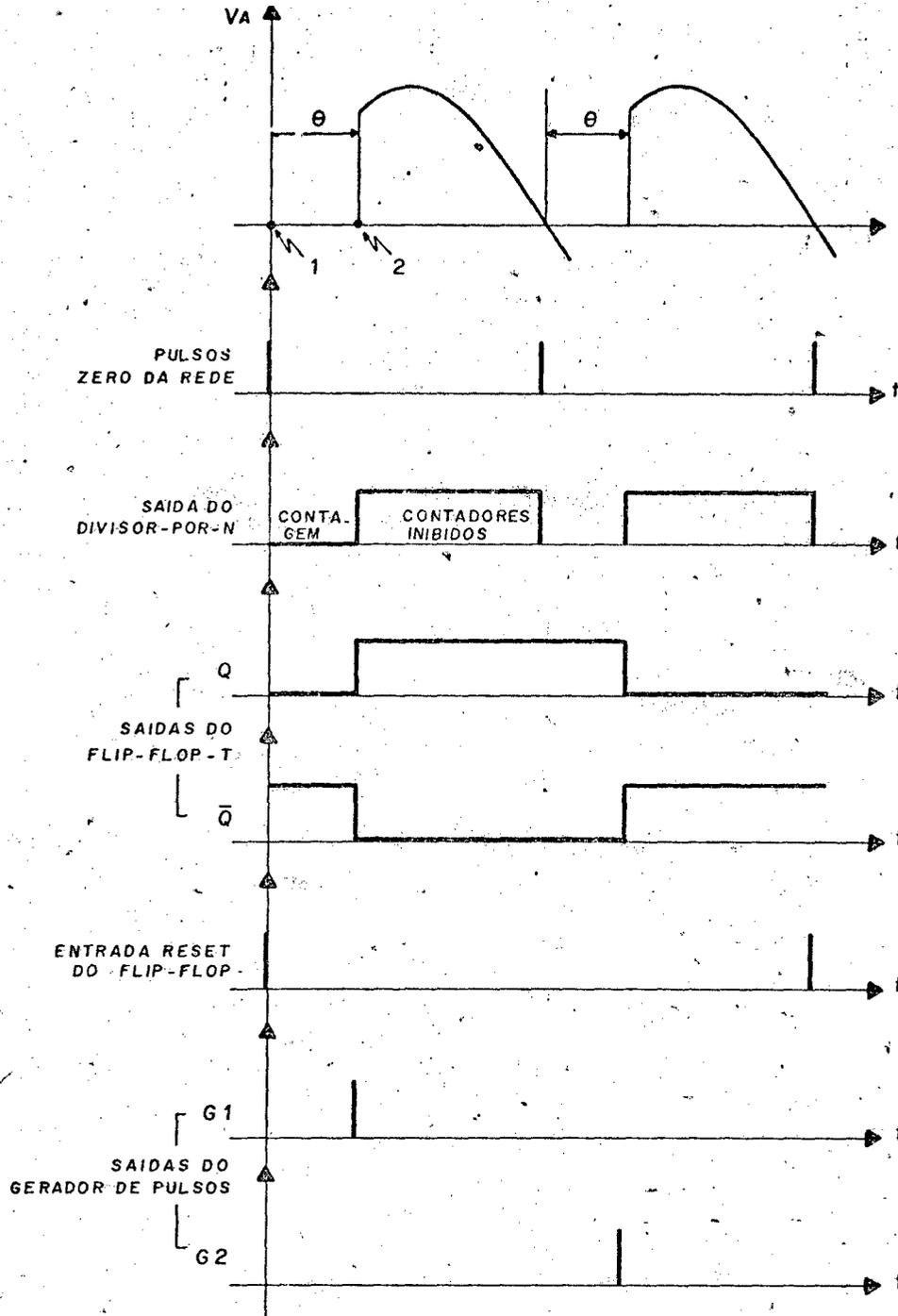
Com base nesta relação, pode-se programar o disparo dos tiristores da ponte a partir do processador.

Para isso foram utilizados contadores programáveis que dividem uma frequência, com valor previamente estabelecida, pelo número  $N$  enviado pelo microprocessador.

A frequência resultante da divisão por  $N$ , quando completa um ciclo dispara um pulso, gerando com isto os trens de pulso de 60 Hz com defasagem controlada em relação à passagem por zero da tensão da rede:

Em um ângulo de disparo mínimo, equivalente a zero graus, a divisão se fará por  $N = 1$  e para um ângulo máximo de  $180^\circ$  a divisão será por  $N = 2^n$ , onde  $n$  é o nº de vias paralelas que definem  $N$ , (o valor no protótipo é de  $n = 8$ ), correspondendo com isto a uma divisão máxima.

É importante lembrar que o carregamento da informação vinda do microprocessador (o número  $N$ ) nos contadores programáveis, bem como o início das contagens, devem ocorrer com os zeros da tensão da rede. Quando completado o ciclo da frequência



Em 1 ocorre o carregamento das informações N nos contadores e as contagens são iniciadas.

Em 2 o ciclo da frequência resultante é completado, os pulsos de disparo são produzidos e os contadores inibidos.

Fig.13

resultante da divisão por N (este é o momento dos pulsos de disparo dos tiristores), os contadores programáveis devem ser inibidos até o próximo zero da rede.

Com estas informações e sabendo que o ângulo máximo é de  $180^\circ$ , a divisão programada terá necessariamente como frequência mínima resultante 120 Hz. Com isto a frequência fixa usada na divisão terá seu valor determinado:

$$120 \text{ Hz} = \frac{f}{N_{\text{max}}}$$

$$f = 120 \times N_{\text{max}}$$

$$f = 120 \times 2^8 = 30,72 \text{ KHz}$$

A relação entre o ângulo de disparo  $\theta$  e o N pode também ser determinada:

$$\theta = \frac{180^\circ}{2^8} \times N \quad \text{ou} \quad \theta = \frac{\pi}{256} \times N \quad \text{para } 0 < N < 256$$

Na relação ângulo por número N a resolução será de  $0,703^\circ$ , ou seja, para cada variação do bit menos significativo teremos uma variação de  $0,703^\circ$  no ângulo de disparo.

$$\theta = 0,703^\circ \times N$$

Este bloco tem então uma frequência fixa de 30,72 KHz que será dividido por N a partir do zero da rede. E quando completado o ciclo de frequência resultante, os pulsos de disparo dos tiristores são produzidos e os contadores serão inibidos até o próximo zero da rede.

A figura 13 serve para esclarecer os eventos deste circuito.

### 3.2.b. - Implementação

A figura 14 mostra os circuitos correspondentes ao gerador de pulsos.

Este bloco foi implementado com contadores pro-

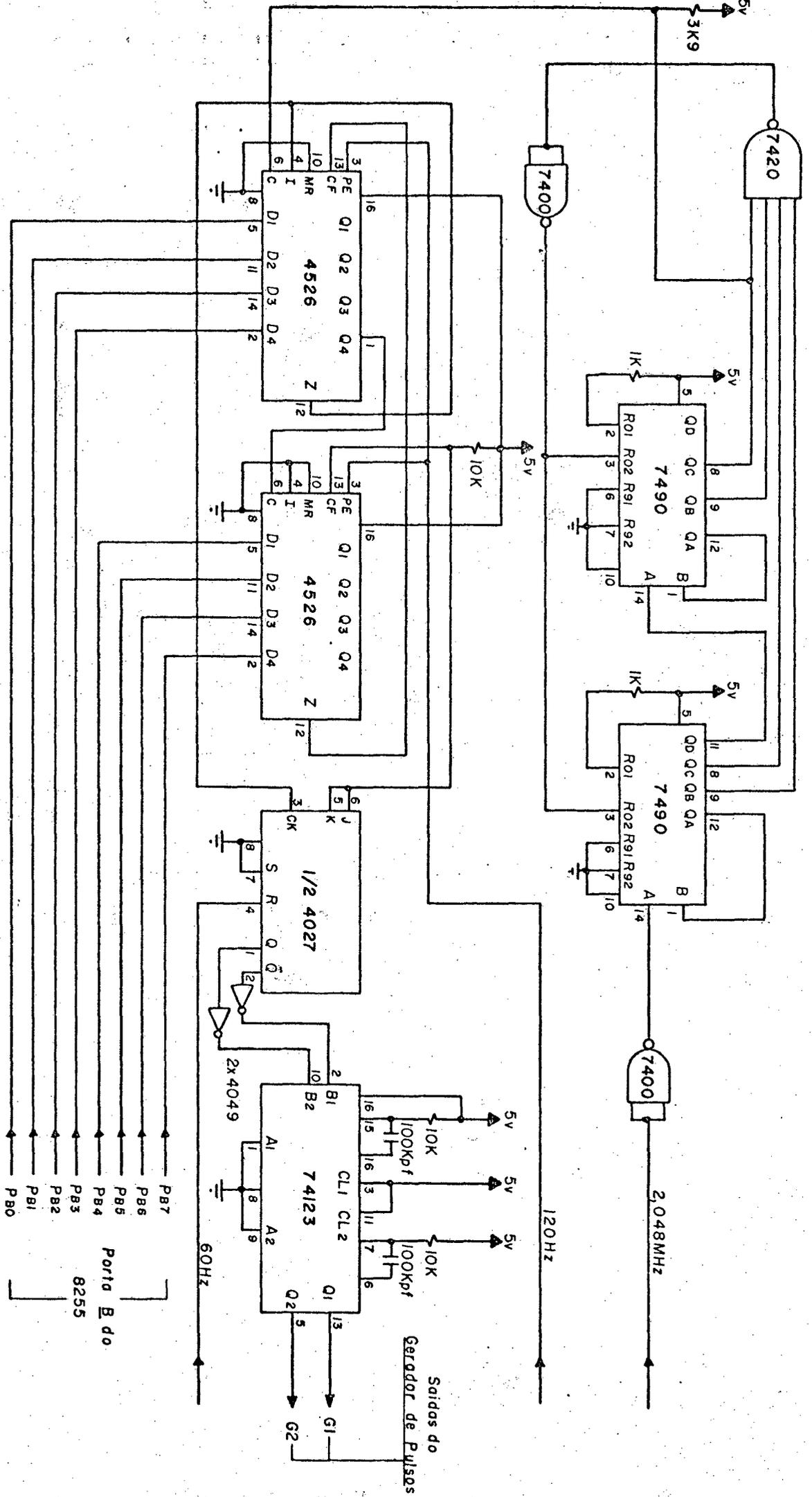


Fig.14. Gerador de Pulsos

gramáveis CMOS 4526, binário de quatro bits. Estes contadores são síncronos e usam contagem decrescente (contadores "down"). Apresentam duas entradas para relógio (C e I), uma entrada para carga paralela (PE), quatro bits ( $D_0 - D_3$ ) para entradas paralelas, uma entrada (CF), quatro saídas paralelas ( $Q_0 - Q_3$ ), uma saída chamada "zero" (Z) e um "MASTER RESET" (MR) entrada assíncrona que inicializa as contagens ( $Q_0 = Q_1 = Q_2 = Q_3 = 0$ ).

A informação nas entradas paralelas ( $D_0 - D_3$ ) é carregada no contador enquanto a entrada PE está em nível 1, independente das outras entradas, com exceção para "MASTER RESET" que deve permanecer nível "0".

Quando PE e I estão a nível "0" a entrada de "clock" ativa será C, com contagens na subida dos pulsos. Em caso contrário com PE em nível zero e C em nível 1, a contagem será na descida dos pulsos na entrada de "clock" I. A saída "Z" será 1 quando o contador está no estado zero ( $Q_0 = Q_1 = Q_2 = Q_3 = 0$ ) e CF está em nível 1.

A frequência de "clock" para estes contadores programáveis, que no caso será a frequência sujeita a divisão programada, tem como valor 30,72 KHz e no protótipo foi obtida a partir do sinal  $\phi_2$  (TTL) disponível no microcomputador MCS-80.

A obtenção do 30,72 KHz a partir de 2,048 MHz ( $\phi_2$ ) foi conseguido segundo a divisão por 66 com dois contadores em cascata, 7490, que são assíncronos TTL, ligados no formato divide-por-dez (BCD). Sendo o 7490 um contador assíncrono ou ripple com entradas para "reset", a divisão por 66 é realizada com as saídas, que serão nível 1 no último estado do ciclo da divisão, conectados em um "AND" para produzir o sinal de "reset" (zerar).

Em contagem estes contadores apresentam as entradas de "reset" como;  $R_{0,1} = 1$  e  $R_{9,1} = R_{9,2} = R_{0,2} = 0$ . Para zerar os contadores é necessário  $R_{0,2} = 1$ . O bit mais significativo desta contagem até 66 (QC) é usado como entrada para os contadores programáveis, com a frequência de 30,72 KHz.

Devido a incompatibilidade entre os níveis de tensão de saída TTL (nível "0"  $\approx 0,4V$  e nível "1"  $\approx 2,5V$ ) e entrada CMOS (nível "1"  $> 3,5$  e nível "0"  $< 1,5$ ), é necessário uma resistência de "pull-up", para subir a tensão nos valores desejados para nível "1" CMOS, na entrada do "clock".

Os contadores programáveis 4526 são em número de dois usados em cascata, formando oito bits de saída ( $Q_0 - Q_7$ ). Estão arranjados de modo a que a contagem ocorra em só um ciclo. como entrada de "clock" é usado a entrada  $\underline{C}$  (contagem na subida).

O contador 4526 mais significativo da montagem apresenta a entrada  $CF = 1$ , de modo que quando suas saídas paralelas estiverem no estado zero ( $Q_4 = Q_5 = Q_6 = Q_7 = 0$ ) a sua saída  $Z$  subirá para nível 1. Com isto a entrada  $CF$  da pastilha menos significativa passará a 1, determinando que na próxima passagem por zero das saídas paralelas desta pastilha ( $Q_0 - Q_3$ ) a subida para nível "1" também da saída  $Z$ . E, portanto a entrada  $I$  na pastilha menos significativa passará a valor "1" inibindo a contagem.

Os contadores só serão desinibidos com a entrada  $PE = 1$ , que permite que o número  $N$  que está nas entradas paralelas ( $D_0 = D_7$ ) passe para as saídas paralelas ( $Q_0 = Q_7$ ).

Isto nos garante que os contadores partirão de um número  $N$  carregado nas saídas para  $PE = 1$ , e farão uma contagem decrescente para  $PE = 0$  até as saídas  $Q_0 = Q_1 = Q_2 \dots = Q_7 = 0$ ,

completando o ciclo da frequência resultante da divisão por  $N$ . Os circuitos permanecerão inibidos até que PE assuma novamente valor 1, jogando nas saídas paralelas ( $Q_0 - Q_7$ ) o número  $N$ .

Os oito bits de informação do número  $N$  são enviados do microcomputador, através, de uma P.P.I. 8255 atuando em MODO 0, (com porta B como saídas). Nas entradas PE dos dois contadores é aplicado um trem de pulsos na frequência de 120 Hz correspondendo aos zeros da tensão da rede.

A frequência resultante da divisão por  $N$  pode ser tirada da saída "Z" da pastilha menos significativa dos contadores 4526. E da maneira como são forçados estes contadores esta saída terá sempre 120 Hz. Esta frequência será aplicada no "clock" de um flip-flop 4207, funcionando como tipo T, para divisão por dois. As duas saídas deste flip-flop ( $Q$  e  $\bar{Q}$ ) são usadas uma para cada semi-ciclo da rede. O sinal "reset" dos flip-flop é formado por pulsos com frequência de 60 Hz tirados dos zeros da tensão da rede. Esta sincronização tem como finalidade evitar que as saídas do flip-flop atuem em semi-ciclos trocados.

Estas saídas dos flip-flops são aplicadas a dois monoestáveis para a obtenção de trens de pulsos responsáveis pelos disparos dos tiristores. Para os monoestáveis é utilizado um 74123 TTL, que apresenta um par (dual) numa pastilha.

Em nível zero o "Fan-out" do CMOS 4027 ( $i_{OL} \approx 0,8$  mA para  $V_{CC} = 5$  V) é incapaz de sustentar o TTL74123 ( $i_{O1} \approx -1,6$  mA). Para solucionar o problema foi usado um 4049, que é um "Hex-Buffer" inversor, que apresenta a capacidade de "Fan-out" para dois TTL comuns.

As constantes de tempo dos monoestáveis foram calculadas de modo a que os pulsos tivessem duração suficiente

para as correntes dos tiristores se manterem.

A largura dos pulsos do 74123 são dados por:

$$T = K \times R_T \times C_{ext} \left(1 + \frac{0,7}{R_T}\right)$$

como:  $K = 0,28$        $C_{ext} = 100 \text{ KpF}$        $R_T = 10K\Omega$

$$T \approx 0,3 \text{ m seg.}$$

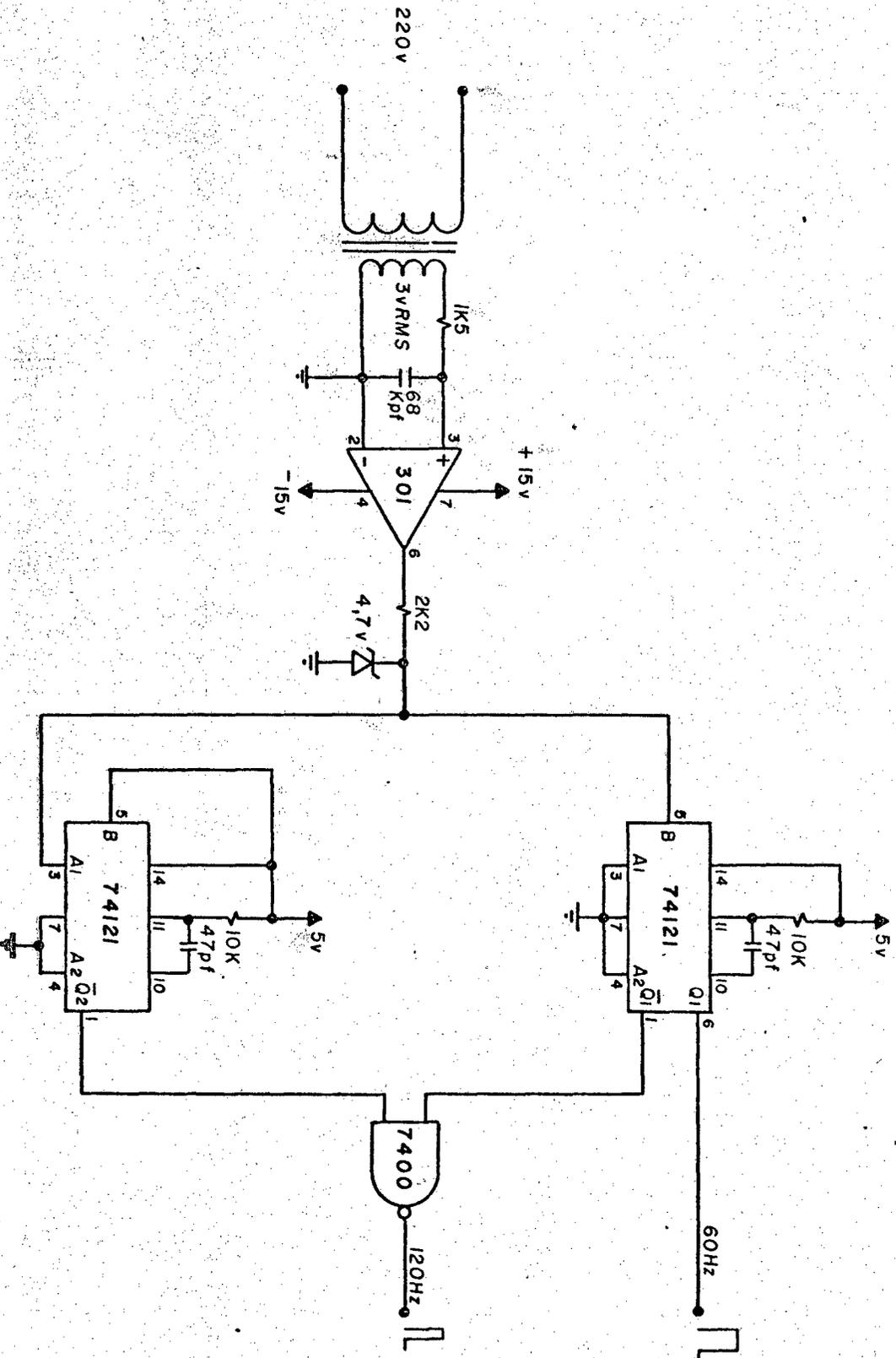
Os dois monoestáveis são disparados na subida (entrada B), produzindo cada um com sua saída Q um trem de pulsos de 60 Hz, que são os sinais  $G_1$  e  $G_2$  saídas do gerador de pulsos. A figura 13 mostra a relação dos sinais no gerador de pulsos

### 3.3. - Circuitos de sincronismos

Diversas operações no sistema apresentam a necessidade de um sincronismo com a tensão aplicada na entrada da fonte. Os trens de pulsos usados para esta finalidade serão obtidos pelo bloco "circuitos de sincronismos". Este circuito faz a detecção dos zeros da tensão da rede. Conforme é mostrado na figura 15, o circuito é constituído de um transformador que baixa a tensão de 220 V da rede para 3 V, que por sua vez é comparada com o terra do sistema. Na onda quadrada resultante desta comparação são forçados níveis de tensão aceitos por circuitos lógicos TTL, por um limitador constituído com um diodo zener de 4,7V. Para nível "1" teremos 4,7 V e como nível "0" -0,7 V. Na comparação foi usado um amplificador operacional 301.

A onda quadrada de 60 Hz é aplicada na entrada de dois monoestáveis TTL 74121, um disparado na subida e outro na descida.

A duração dos pulsos foi calculada de modo a que tivessem a mínima duração possível para serem usados nos contado



Detecção dos zeros da rede. Fig.15

res programáveis CMOS 4526 na entrada PE.

Para monoestáveis 74121 a duração dos pulsos é dado por:

$$T = 0,7 RC \quad \text{como} \quad R = 10 \text{ K}\Omega \quad \text{e} \quad C = 47 \text{ PF}$$

$$T = 329 \text{ ns}$$

As saídas  $\bar{Q}$  ( $\bar{Q}_1$  e  $\bar{Q}_2$ ) dos dois monoestáveis são aplicadas em um NAND, resultando na saída de pulsos na frequência de 120 Hz.

$$(\overline{\bar{Q}_1 + \bar{Q}_2}) = Q_1 + Q_2$$

Pulsos com a frequência de 60 Hz são tirados diretamente de um dos monoestáveis, das saídas Q.

O filtro na entrada do comparador evita o ruído introduzido na tensão da rede pela comutação dos tiristores. Estes ruídos serão função da carga da máquina.

### 3.4. - Circuitos de amplificação e isolação e a ponte de tiristores

Estes dois blocos do diagrama da figura 1, são os responsáveis pelo acionamento da máquina de C.C. Os dois trens de pulsos, saídas do bloco gerador de pulsos, devem ser amplificados e duplicados, para serem aplicados nos circuitos de disparos dos quatro tiristores da ponte.

A duplicação é feita por dois transformadores de pulsos cada um com dois secundários, o que também serve de isolação entre os circuitos de baixa e alta potência.

A amplificação dos pulsos é feita com um amplificador com dois estágios a transistor. A figura 16 mostra estes circuitos.

A tensão aplicada na armadura da máquina de C.C. apresenta um comportamento não linear em função do ângulo de dis

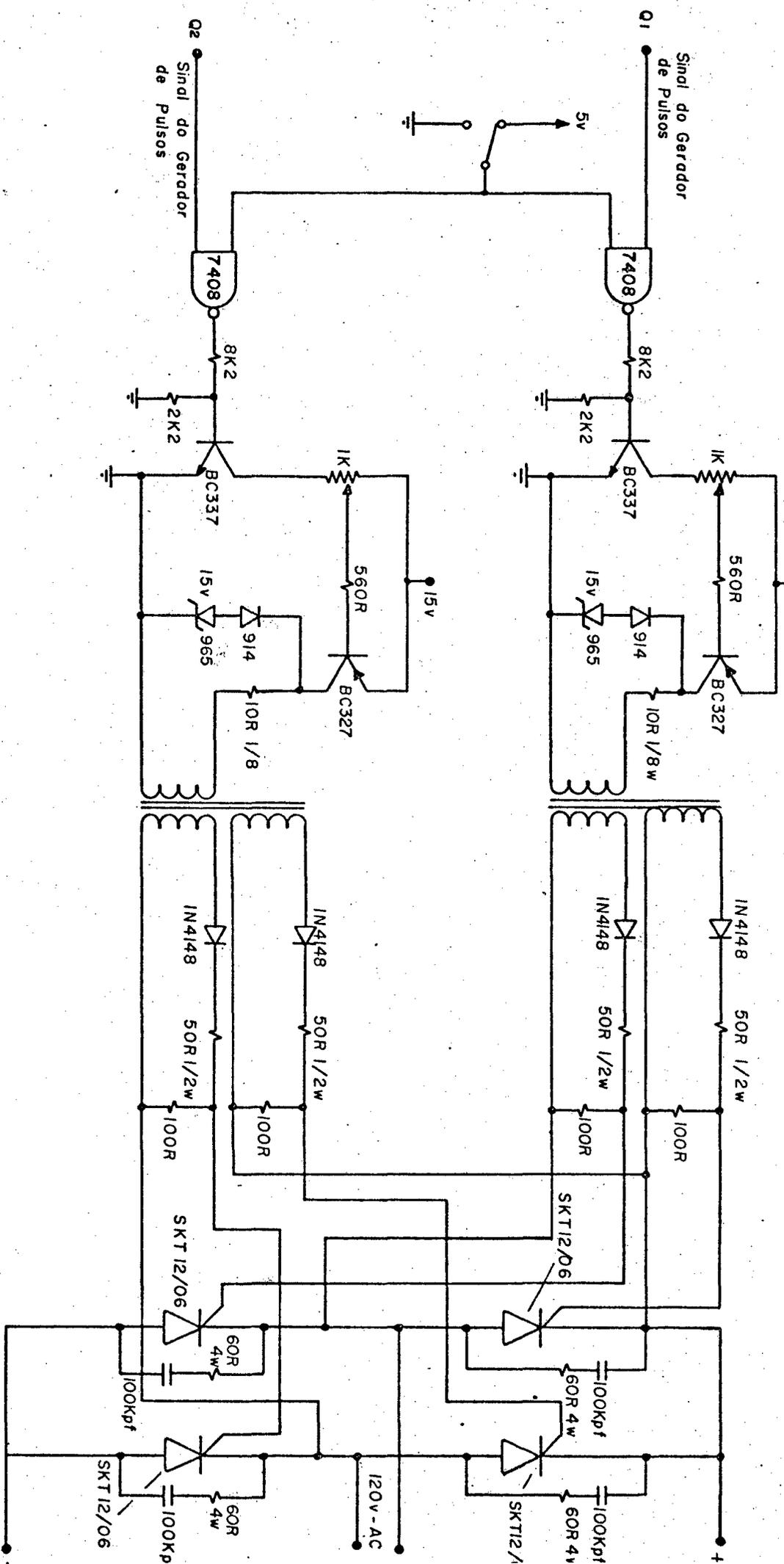
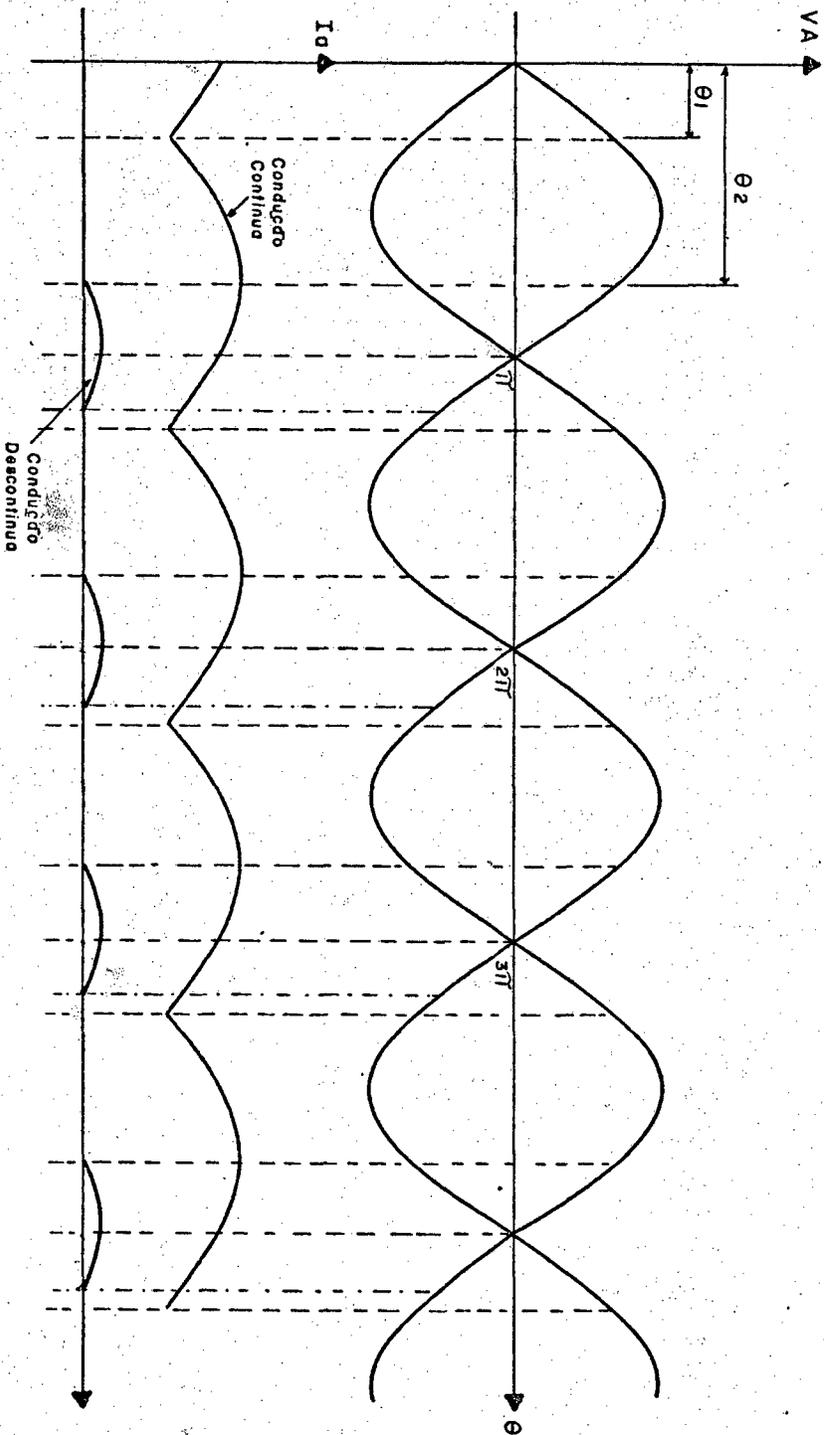


Fig.16. Estagio de Amplificaçãõ dos Pulsos de Gate.



OBS:  $\theta_1$  — Condução Contínua  
 $\theta_2$  — Condução Descontínua

Fig.17

Na conversão do modo de referência

para  $\theta$ .

Esta função pode ser dividida em duas regiões distintas, que definem o comportamento da corrente de armadura:

- Região de condução contínua; onde  $V = K \cos \theta$ . Nesta região a corrente de armadura não se anula.

- Região de condução descontínua; onde  $V$  é uma função bastante complexa do ângulo de disparo e a corrente de armadura é descontínua.

As duas regiões dependem, além de  $\theta$ , das características da máquina. A figura 17 mostra condução contínua para  $\theta_1$  e descontínua para  $\theta_2$ .

### 3.5. - Tacômetro Digital

#### 3.5.a. - Considerações iniciais

O tacômetro digital pode ser dividido em duas partes: numa primeira onde pulsos com a frequência proporcional à velocidade angular da máquina são obtidos, e na segunda parte, constituída por contadores binários que realizam contagens destes pulsos, com base de tempo fixa.

A obtenção destes pulsos é feita com um disco com 240 furos, de inércia desprezável, fixado no eixo do motor. O disco girando entre um par led-fototransistor produz este trem de pulsos proporcional à velocidade angular da máquina. Os pulsos devem ser amplificados e conformados entre valores que possibilitam a contagem.

O tacômetro foi idealizado de modo a possuir uma resolução de 1rpm ou seja, 1rpm para cada pulso na entrada dos contadores (1rpm para cada variação do bit menos significativo dos contadores) (1), e com faixa de velocidade de 0 - 4096 rpm. Os

valores das contagens, portanto, os valores da velocidade angular lidos serão enviados ao microprocessador em 12 vias paralelas, o que determina a limitação em 4096 rpm ( $2^{12} = 4096$ ).

Esta resolução é determinada da seguinte maneira:

seja

W velocidade angular em rpm

F número de pulsos ou furos por revolução

N número de pulsos

P período de amostragem em segundos

O número de pulsos no período P, para velocidade angular W será de :

$$N = W \frac{\text{revolução}}{\text{minuto}} \times \frac{1 \text{ minutos}}{60 \text{ segundos}} \times F \frac{\text{pulsos}}{\text{revolução}} \times P \text{ (segundos)}$$

$$N = \frac{W \times F \times P}{60} \text{ pulsos}$$

A resolução entre velocidade por pulso produzida será:

$$\frac{W}{N} = \frac{60}{F \times P} \frac{\text{rpm}}{\text{pulso}}$$

fazendo

$$\frac{60}{F \times P} = 1 \quad \therefore F \times P = 60$$

Assumindo o período de contagem como 62,5 m seg para o tacômetro digital, fica determinado  $F = 960$  furos.

Devido às limitações de ordem técnica o disco foi realizado com 240 furos. Como o aumento do tempo de contagens traria desvantagens para o sistema (lentidão do processo), optou-se pela multiplicação por quatro do trem de pulsos que sai do fototransistor, conseguindo-se, portanto, 960 pulsos por revolução da máquina. E a resolução será de 1 rpm para cada pulso na entrada dos contadores, ou seja, cada variação de 1 LSB (bit menos significativo dos contadores) corresponde 1 rpm. Outra maneira

ra de se encarar este problema da resolução seria considerar que para cada pulso do fototransistor obtêm-se 4 rpm nos contadores.

### 3.5.b. - Circuito de obtenção da frequência da máquina

A figura 18 apresenta o circuito de obtenção da frequência proporcional a velocidade angular da máquina. Neste circuito uma corrente de 100 mA polariza o diodo emissor de luz.

O fototransistor se apresenta cortado e a incidência de luz o coloca na zona de condução.

No caso do disco com furos o seu giro produz pulsos mais um nível DC. A medida que a velocidade da máquina aumenta o sinal de coletor do fototransistor tem a amplitude dos pulsos diminuída.

O potenciômetro ligado na entrada inversora é ajustado de modo a compensar estas variações, possibilitando a detecção de todos os pulsos pelo comparador.

O sinal na saída do comparador é colocado em níveis TTL pelo limitador formado com o diodo Zener.

Este trem de pulsos é aplicado em dois monoestáveis um disparando na subida e outro na descida, as duas saídas  $\bar{Q}$  dos dois monoestáveis são aplicadas numa porta "NAND", garantindo um sinal com o dobro da frequência do sinal de entrada. Repetindo o mesmo processo com o sinal de frequência duplicado, os pulsos resultantes terão quatro vezes a frequência dos pulsos gerados no coletor do fototransistor. Dois TTL 74123 foram usados neste processo de multiplicação.

Nesta multiplicação devem ser tomados cuidados com a definição das constantes de tempo dos monoestáveis principalmente na frequência máxima permitida pelos contadores, para evitar superposição dos pulsos disparados na subida com os na des

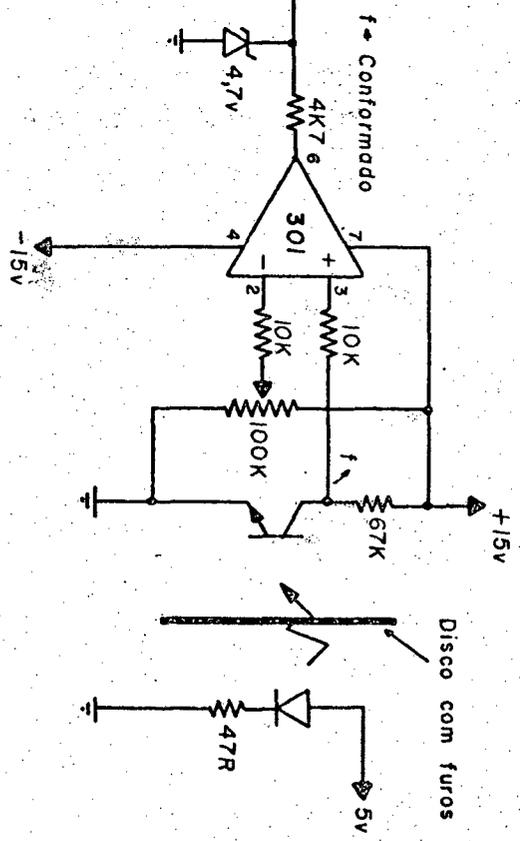
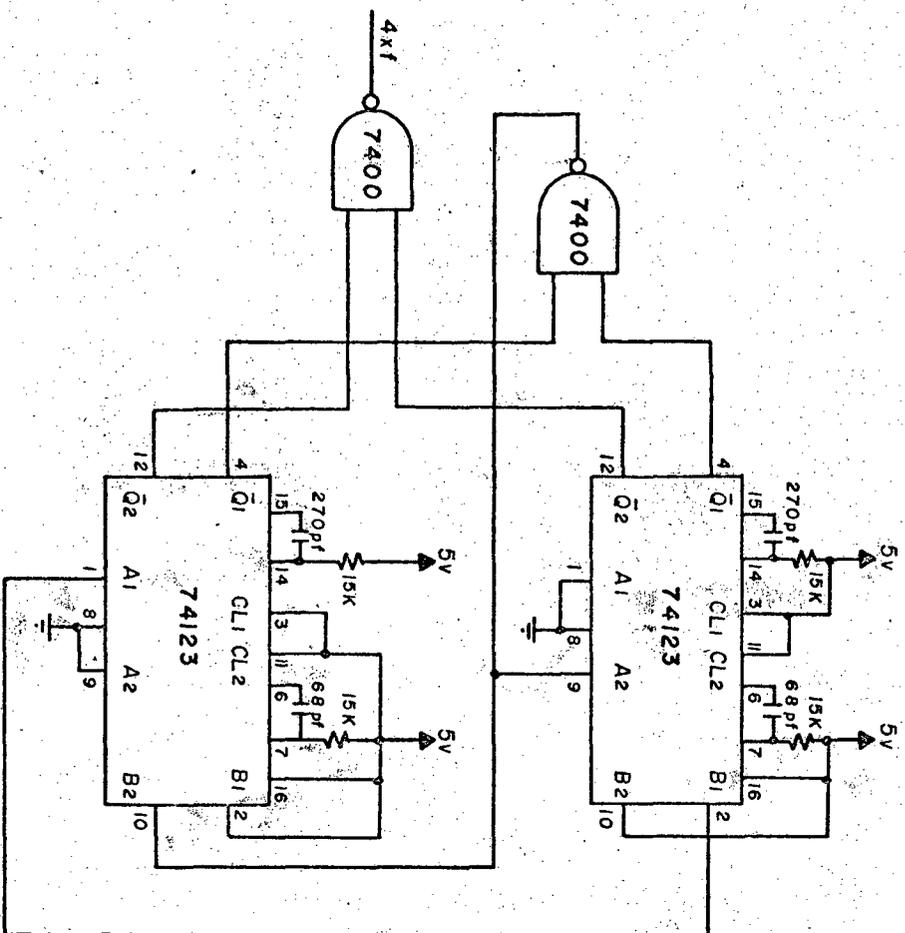


Fig.18. Circuito de Obtenção da Velocidade da Máquina

cida.

O valor máximo para 12 bits de contadores binários é 4096, isto num tempo de amostragem de 62,5 mseg. Portanto a frequência para esta contagem será:

$$F = \frac{4096}{62,5} \text{ KHz} = 65,53 \text{ KHz}$$

Que corresponde a quatro vezes a frequência da saída do comparador.

$$F_C = \frac{F}{4} = 16,38 \text{ KHz}$$

Esta frequência máxima na saída do comparador garante um período mínimo de 61  $\mu$  seg.

Este sinal na saída do comparador não forma exatamente uma onda quadrada, mas assim mesmo, devido ao fato do disco ter o espaçamento de furos e não furos iguais, a largura dos pulsos não são muito diferentes da metade do período.

Um fator importante verificado foi a influência do "slew rate" (taxa de variação do sinal de saída por unidade de tempo  $S = \left| \frac{dV_O}{dt} \right|$ ) do amplificador operacional 301 usado no circuito como comparador. Em frequências com valores próximos a 16,38 KHz os pulsos apresentam uma duração maior que a metade do período.

De qualquer maneira para uma frequência máxima de 16,38 KHz na saída do comparador, a largura dos pulsos terá valores no mínimo da ordem de 30  $\mu$  seg (metade do período). Com base nisto foi estipulado como constante de tempo para os dois primeiros monoestáveis o valor de:

$$\tau = 270 \text{ pF} \times 15 \text{ K}\Omega = 4050 \text{ n seg}$$

O que corresponde a pulsos com duração de 1,620  $\mu$  seg, assegurando, com isto a não superposição dos pulsos dispa

rados na subida e descida na duplicação da frequência 16,38 KHz .

Para os outros dois monoestáveis que garantem a quadruplicação da frequência foi estipulado como constante de tempo:

$$\tau = 15 \text{ K}\Omega \times 68 \text{ pF} = 1020 \text{ nseg}$$

E os pulsos correspondentes terão valores de 408 nseg. Garantindo finalmente, que na frequência máxima, a cada pulso que sai do comparador corresponde a quatro na saída do circuito. A figura 19, mostra a relação dos tempos entre os pulsos no circuito de obtenção da frequência proporcional a velocidade da máquina.

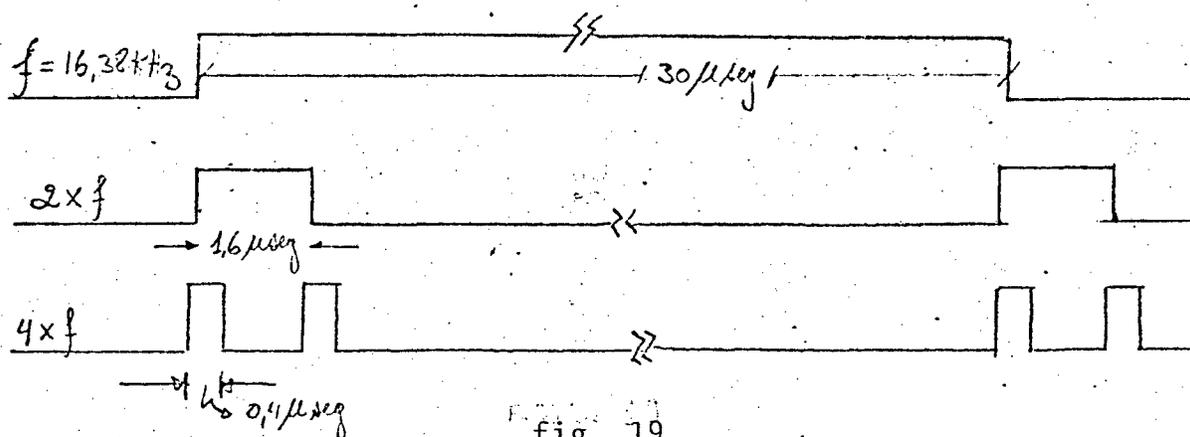


fig. 19

Este circuito para a faixa de valores de frequência em que foi calculado, não apresentou problemas.

### 3.5.c. - Circuito contador da frequência da máquina

Neste circuito, da figura 20, a contagem da velocidade da máquina é feita por três contadores binários 7493 (divide-por-16), aproveitando que o microcomputador também utiliza o código binário. Com isto, a entrada de dados é facilitada, tornando desnecessário o uso de conversores.

Os doze bits dos contadores 7493 limitam a velocidade ao valor 4096 ( $2^{12}$ ). A frequência  $4 \times f$  entra numa porta

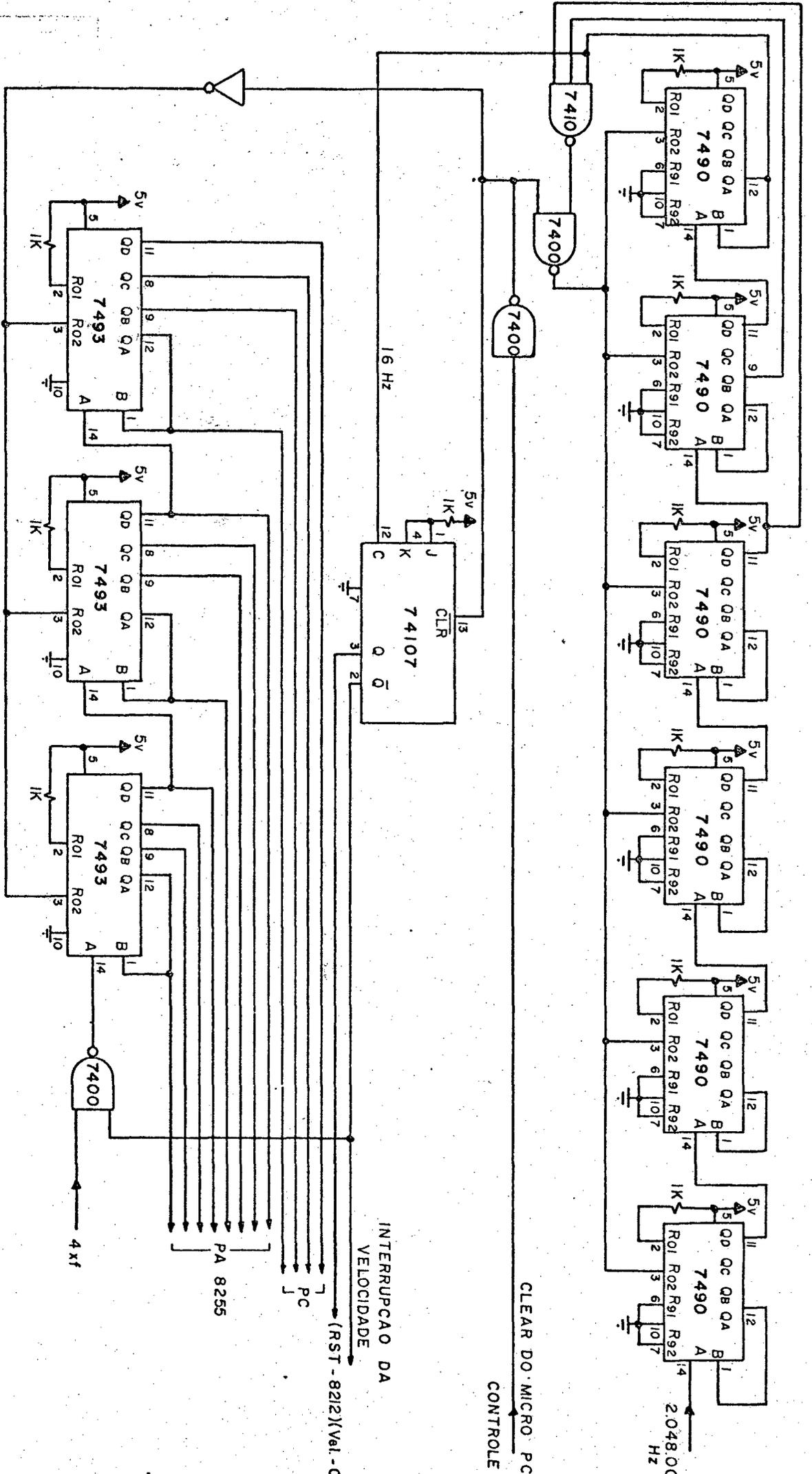


Fig. 20. Contador da Frequência da Máquina

NE ("NAND") que é controlada pela base de tempo (62,5 mseg). Terminado este tempo de contagem, esta porta NE é bloqueada e os doze bits de saída dos contadores são enviados a uma porta (A) e meia (C) no controlador de interface 8255 do microcomputador. Esta unidade é programada em MODO 0, e neste modo de funcionamento a entrada das portas não apresentam memória ("LATCH").

Para evitar mais componentes neste circuito, no caso, memórias, a alternativa foi colocar a base de tempo sob controle do processador, de maneira que ao encerrar o tempo de contagem, o "NAND" é bloqueado, os contadores permanecem com seus valores e a seguir é enviado ao microcomputador um pedido de interrupção. A subrotina acionada pela interrupção deve obter os valores nas portas A e C da 8255. A mesma subrotina, depois da leitura da velocidade, está encarregada de zerar os contadores 7493 e reiniciar um novo tempo de contagens, inicializando para isto os contadores que controlam este tempo. Estas duas operações, tanto de zerar os contadores 7493, como inicializar as contagens são feitas com o envio de um pulso pela saída 4 da porta C.

A resolução do tacômetro foi definida de modo que valores da velocidade são obtidos em rpm, sem a necessidade de nenhuma conversão. A precisão destas medidas implica na garantia de base de tempo corretas. Estes tempos, por isto, são obtidos a partir do oscilador a cristal disponível no microcomputador, com frequências de 2,048 MHz ( $\phi_2$  TTL).

Na obtenção do tempo de 62,5 mseg, que é base de tempo para as contagens, é feito a divisão de 2,048MHz por 128 000, o que resulta na frequência de 16 Hz (período igual a 62,5 mseg). Uma nova divisão por dois usando um flip-flop T garante a obten-

ção de um pulso com a duração de 62,5 mseg. A figura 21 mostra as frequências de 16 Hz e 8 Hz, sem o interferência do processador.

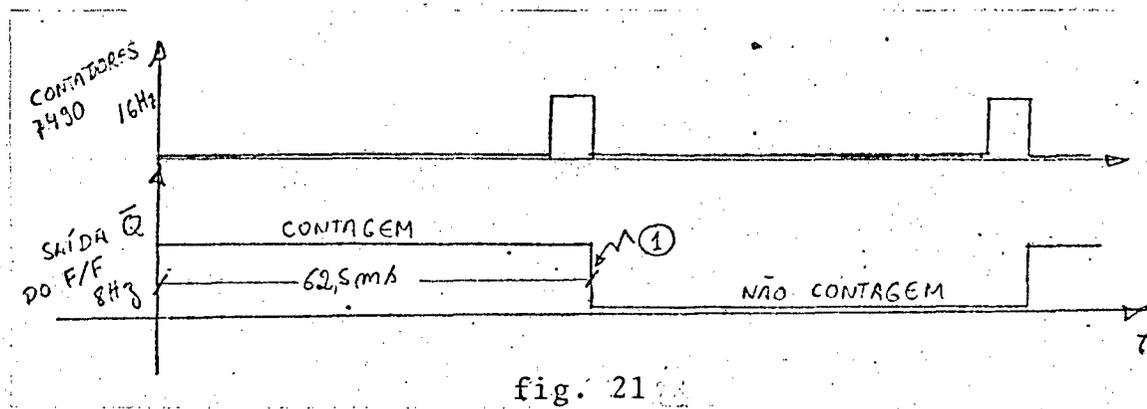


fig. 21

Usando a saída  $\bar{Q}$  do flip-flop como base de tempo (aplicada no "NAND" que controla as contagens de velocidade), este sinal, enquanto, estiver em nível alto haverá a passagem de  $4 \times f$  para os contadores; na descida deste sinal em (1), o micro é chamado para a leitura de  $W_a$ , devendo após, inicializar os contadores usados na divisão de  $\phi_2$  para 16 Hz. Este mesmo sinal de inicialização enviado pelo microprocessador deve também atuar no "CLEAR" do flip-flop, garantindo sempre a saída  $\bar{Q}$  do flip-flop em nível alto quando da inicialização dos contadores pelo processador.

O resultado desta ação do microcomputador sobre os contadores e o flip-flop é a alteração da frequência de saída  $\bar{Q}$  do flip-flop. Podendo praticamente suprimir a parte "LOW" do sinal  $\bar{Q}$ , que assumiria a frequência de 16 Hz (esta é uma das situações possíveis). Isto porque a duração da execução da subrotina que corresponde ao nível zero em  $\bar{Q}$  não vai além de 80  $\mu$ seg, o que em comparação a 62,5 mseg é praticamente nulo. Os contadores que fornecem a frequência de 16 Hz devem também sofrer uma pequena variação desprezível (80 $\mu$ ) em seu ciclo. A figura 22 mostra esta situação.

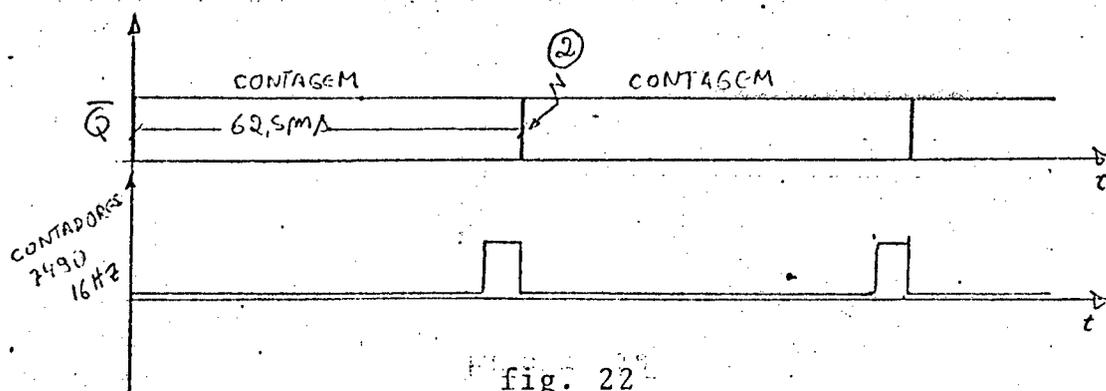


fig. 22

Na descida em (2) ocorre a interrupção, a leitura da velocidade e logo em seguida pela porta C na saída 4 de uma 8255 é enviado um pulso que atua no "clear" do flip-flop, no "reset" dos contadores 7490 e também zera os contadores 7493 usados na medida da frequência. Com isto o sinal  $\bar{Q}$  inicia novamente as contagens.

Estas interrupções devem necessariamente ocorrer, pois os contadores 7493 precisam ser zerados para que não haja o acúmulo de valores antigos de  $W_a$  com novas contagens.

O controle do microprocessador está sobre o tempo entre duas interrupções consecutivas, que pode ser variado por "software" (na subrotina de leitura de velocidade) de 62,5 mseg a 125 mseg. A base de tempo deve permanecer inalterada no seu valor de 62,5 mseg ( $\bar{Q} = 1$ ), para que não ocorram erros de leitura.

Na subrotina da interrupção de velocidade usada no protótipo o período de amostragens dos valores de velocidade no microcomputador foi programado para 62,5 mseg.

Para a divisão de 2,048 MHz na obtenção de 16 Hz foram usados seis contadores 7490, onde as saídas, que assumem valor "1" no último estado do ciclo na contagem até 128 000, são entradas num "NAND" 7410, formando o pulso que deve iniciar novo ciclo. Este sinal invertido é entrada em outra porta "NAND" junto com o pulso também invertido enviado pelo processador em PC 4

(porta C bit 4 da 8255). Isto garante a soma dos dois sinais na entrada "reset" dos contadores 7490. É importante verificar que esta entrada está sujeita ao processador por  $PC_4$ , que pode aumentar o ciclo ou inibir estes contadores.

Considerando B o resultado do "AND" entre as saídas dos contadores em "1" no último estado da contagem até 128000 e  $PC_4$  a saída em que o micro envia seus pulsos, teremos:

$$\text{"Reset" 7490} = (\overline{PC_4} \overline{B}) = PC_4 + B$$

$$\text{"CLEAR" F/F} = \overline{PC_4}$$

$$\text{"RESET" 7493} = PC_4$$

O sinal  $Q_A$  do contador 7490 mais significativo é aplicado no "clock" do flip-flop 74107, que está arranjado como tipo T. Esta entrada de "clock" é ativada na descida.

A saída Q do F/F é usada na base de tempo para contagens de  $4Xf$ , ficando em nível alto durante 62,5 m seg e quando na descida este sinal é usado para pedidos de interrupção ao processador. A saída Q é usada na composição das instruções RST. Quando  $Q = 1$  e ocorrer uma interrupção o processador interpreta como interrupção de velocidade, se  $Q = 0$  será uma interrupção de corrente, o que garante prioridade das interrupções de velocidade sobre as de corrente. Isto será discutido com detalhes no bloco sistema de interrupções vetoradas.

A obtenção da frequência de 16 Hz a partir de 2,048 MHz pode ser feita com o uso de contadores binários 7493. O que corresponderia a uma divisão por 1F400H, isto traria a economia de um contador, se comparado com a mesma divisão mas usando contadores 7490. Porém a grande vantagem seria o uso de contadores 4040 CMOS, que são contadores binários assíncronos de 12 bits. Para esta divisão, portanto, bastariam só duas pastilhas

destes contadores. Na parte de contagem da frequência da máquina as três unidades do 7493 podem ser substituídas por apenas um 4040 e com isto, este circuito com grande número de componentes seria reduzido a só três contadores 4040 e um flip.flop CMOS. Isto não foi feito desta maneira pela não disponibilidade destes componentes na época do projeto deste circuito.

### 3.6. - Circuito de obtenção da corrente

#### 3.6.a. - Considerações iniciais

Este sistema não só estabiliza a velocidade da máquina mas também impede que a corrente de armadura ultrapasse de determinado valor, quando da correção de grandes erros de velocidade. A figura 23 mostra o circuito de obtenção de valores de corrente.

Apesar do objetivo em se tratando de corrente da armadura seja um limitador; utilizou-se para isto um conversor analógico digital. Este dispositivo deu uma maior flexibilidade no trato da corrente por programa, possibilitando inclusive que se faça no futuro algo além de uma limitação de corrente.

#### 3.6.b. - Conversor analógico digital

O conversor utilizado é o ADC 800, que é um conversor A/D monolítico implementado com tecnologia MOS canal P. Este dispositivo contém um comparador, 256 resistores e chaves, memórias de saída com terceiro estado, saída de dados ( $D_0 - D_7$ ) compatível com TTL e uma lógica de seleção das chaves e controle. O diagrama desta pastilha está no Apêndice.

Este conversor utiliza o método de aproximações sucessivas, que é um método que diminui em muito o tempo de conversão pois esta é feita com uma comparação bit a bit, enquanto outros processos se baseiam em uma contagem contínua, que depen-

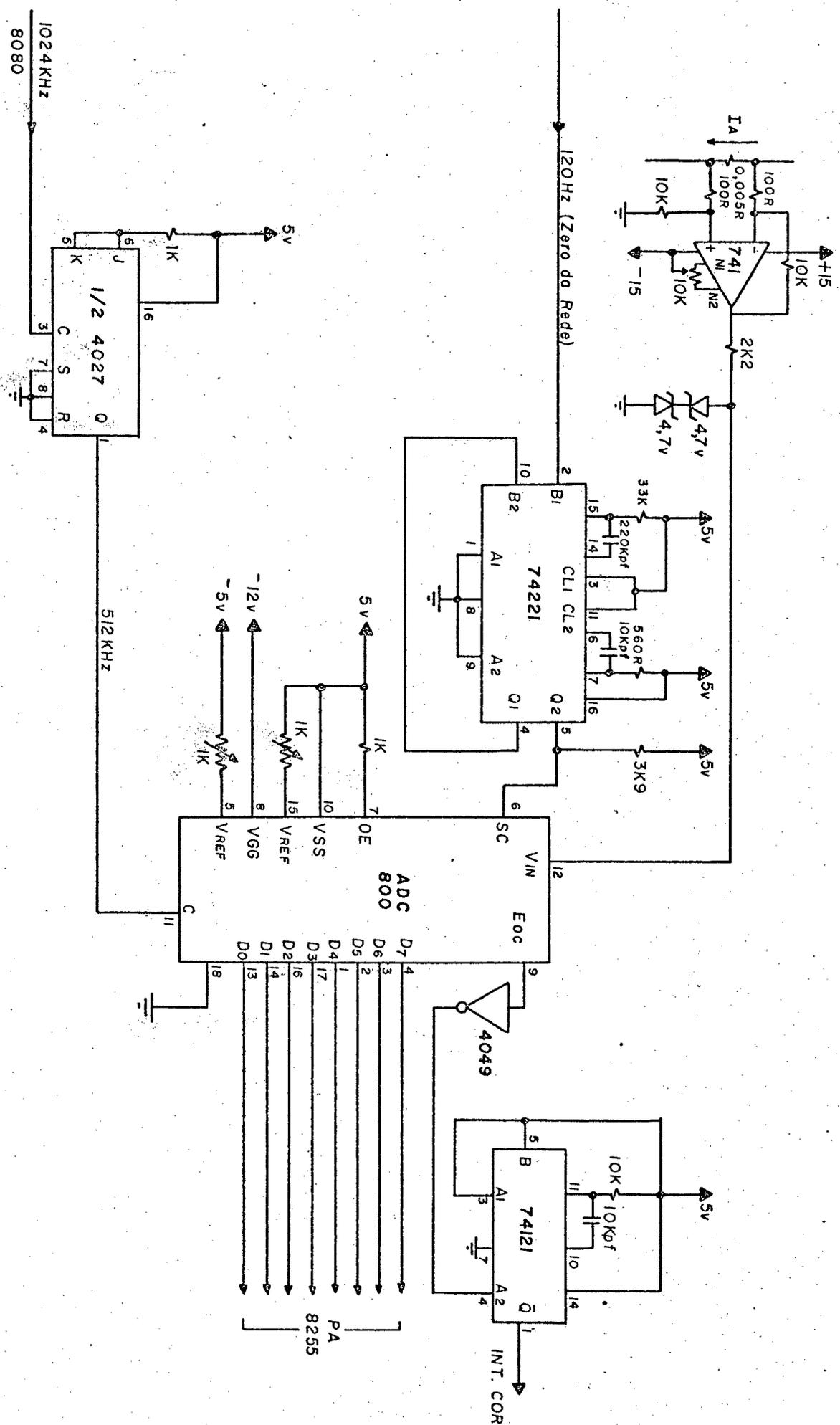


Fig. 23. Obtenção da Corrente

dendo da resolução pode se alongar. Esta técnica consiste na comparação do sinal analógico com a metade da tensão de referência ( $V_R$ ). Se o sinal for maior o bit mais significativo (MSB) do conversor recebe o valor 1 caso contrário será zero. A seguir o próximo bit mais significativo é determinado com a comparação do sinal analógico com  $\frac{3}{4} V_{ref}$  se MSB = 1, ou com  $\frac{1}{4}$ , se MSB = 0. Este processo segue com a comparação do sinal com frações de  $V_R$  (referência), sempre levando em conta o valor do bit anterior, até a determinação do bit menos significativo. A precisão de uma conversão que utiliza este processo depende da fração mínima do  $V$  referência, depende da  $V$  referência (erros de escala), e da resolução do conversor. Circuitos de "amostra-e-mantém" ('sampled-and-hold') são desnecessários para sinais de baixa frequência.

O ADC 800 contém uma rede de 256 resistores de  $300\Omega$  em série em cujas extremidades são aplicados as tensões de referência. As chaves quando acionadas ligam a junção entre dois resistores da rede à entrada do comparador. A tensão  $V_{in}$  (sinal analógico) é comparada no início com  $V_{ref}/2$  no comparador determinando o MSB e o processo segue normalmente até completada a conversão. A lógica de seleção de chaves e controle é quem recebe o resultado do comparador e seleciona as chaves para as comparações.

A conversão tem início com um pulso de "início de conversão" aplicado na entrada SC do ADC 800, a duração máxima deste pulso é de três períodos de "clock". O tempo de conversão corresponde a  $40 \times T$ , onde  $T$  é o período do sinal usado como "clock". Nesta pastilha no fim da conversão o sinal EOC é ativado, visando que o valor convertido está disponível nas memórias de saída, que deve permanecer nas memórias até nova subida de EOC.

### 3.5.c. - Descrição do circuito de aquisição de corrente

A tensão aplicada na armadura da máquina produz uma corrente com dois tipos de comportamento, conforme pode se verificar na figura 17, que são a condução descontínua e a condução contínua. Ambas dependem dos ângulos de disparo e característica da máquina, mas nos dois casos a corrente apresenta uma ondulação.

Esta ondulação determina valores de máximo da corrente de armadura com uma frequência de 120 Hz. Em relação aos zeros da rede estes valores ocorrem com atrasos que variam entre 4,16 mseg e 8,33 mseg, sendo que para ângulos pequenos de disparo nos tiristores a corrente de armadura é elevada e os máximos são próximos do atraso de 4,16 mseg.

O circuito de aquisição foi projetado para que os valores convertidos ocorram o mais perto possível dos máximos das correntes de ângulos pequenos de disparo. Para isto é necessário que o pulso "início de conversão" apresente uma frequência de 120 Hz e um atraso em relação ao zero da rede próximo de 4,16 mseg.

O circuito de aquisição da corrente, mostrado na figura 23, apresenta o conversor ADC 800 com a sua rede resistiva ligada a + 5V e - 5V, o que possibilita a conversão de sinais bipolares entre estes valores.

As ligações do conversor tanto para a tensão + 5V como para - 5V são feitas através de potenciômetros que servem para ajuste de plena escala e a juste de zero respectivamente.

A frequência usada neste dispositivo no "clock" é de 512 KHz, obtida a partir de frequências disponíveis no microcomputador. E o tempo de conversão, com isto, fica determina-

do com valor de aproximadamente  $80 \mu\text{seg}$  que é praticamente zero se comparado com o período do sinal que é amostrado.

O pulso de "início de conversão" é conseguido a partir da saída de 120 Hz do circuito de sincronismos. Esta frequência é usada no disparo de um monoestável que produz um atraso de 5 mseg em relação ao zero da rede, e a largura do pulso de "início de conversão" é controlada por um segundo monoestável.

Isto é conseguido com o uso de uma pastilha 74221, constituída por dois monoestáveis, onde o primeiro é disparado na subida da frequência de 120 Hz, com constante de tempo de 7,2 mseg, correspondendo com isto a um pulso de 5 mseg que é usado para o atraso. A saída  $\bar{Q}$  deste monoestável dispara na subida o outro do par do 74221. A saída Q deste último, terá pulsos de 120 Hz, com largura de  $3,5 \mu\text{segundos}$  e atraso de 5 ms em relação dos zeros da rede.

Estes pulsos devem ser colocados em níveis aceitos pela entrada SC do conversor (nível '1'  $> 4\text{ V}$ , nível "0"  $< 0,8\text{ V}$ ), o que é conseguido com uma resistência de "pull-up".

Estes pulsos de "início de conversão", poderiam ser obtidos com circuitos mais precisos tal como contadores, mas para isto seriam necessários um maior número de componentes. No caso do protótipo foi usado somente um 74221 e os resultados conseguidos foram satisfatórios.

Os valores de corrente na armadura são obtidos a partir de uma resistência de  $0,005 \Omega$  que transforma a corrente em tensão. Esta tensão é amplificada com o uso de um amplificador diferencial com o ganho de -100. A saída do amplificador está limitada entre +5,5 V e -5,5 V por dois diodos Zeners de 4,7 V.

Para uma saída de -5 V o sinal de entrada é de

50 mV, o que garante uma corrente de armadura de 10 A. Como a corrente nominal da máquina usada no protótipo do sistema é de 8 A, a faixa  $\pm 5$  V na saída do amplificador garante uma boa margem para as variações de corrente. No Apêndice, vai em anexo o levantamento de valores analógicos na entrada do amplificador, valores na saída e valores digitais na saída do conversor, dando uma ideia da linearidade da conversão analógica digital.

Os dados convertidos são armazenados nas memórias de saída com a subida do sinal EOC e estes dados são permanentemente disponíveis, pois o terceiro estado está desativado com a entrada OE ("OUTPUT ENABLE") ligado ao nível "1". Esta saída de dados ( $D_0 - D_7$ ) é entrada em uma porta A de uma 8255 programada em MODO 0. Os dados referentes a corrente são adquiridos pelo processador por interrupções.

Os pedidos de interrupções são produzidos a partir do sinal EOC, que quando do início da conversão apresenta nível "0", e sinaliza com nível "1" quando os dados estão disponíveis nas memórias de saída. Este sinal é usado no disparo de um monoestável na subida para nível "1". Como esta saída EOC não tem capacidade de acionar um TTL, foi usado um "buffer" inversor 4049 CMOS, que por sua vez provoca a necessidade do disparo do monoestável na descida do sinal invertido ( $\overline{EOC}$ ).

O pedido de interrupção é a saída  $\overline{Q}$  do monoestável, e a duração deste pulso é de 80  $\mu$ seg.

### 3.7. - Sistema de chaves

O objetivo das chaves é introduzir no microcomputador o valor da velocidade angular de referência ( $W_r$ ) em doze bits de informação. Esta informação é enviada em 8 vias paralelas ao processador por meio de interrupções. Isto traz algumas vanta

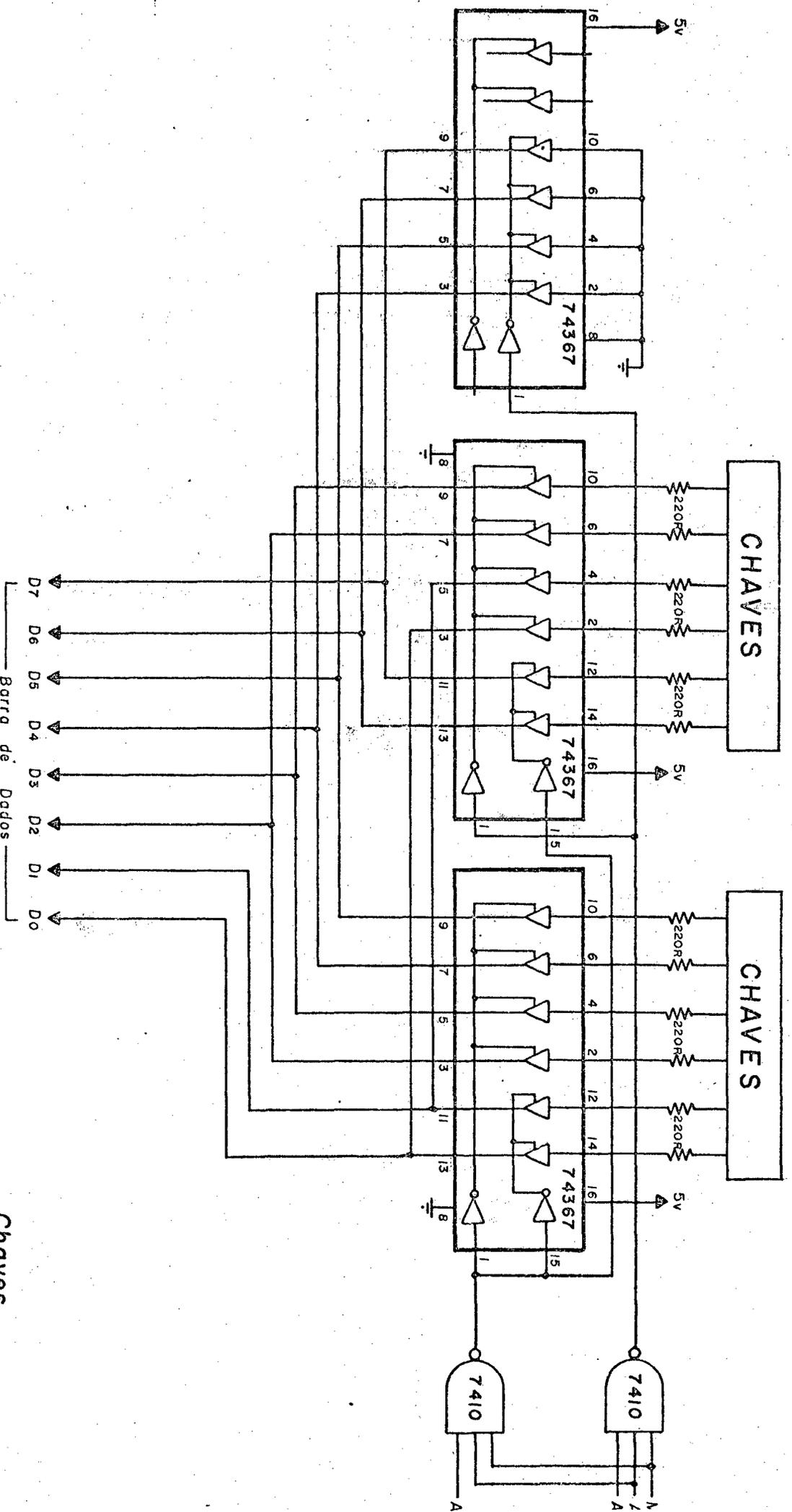


Fig. 24

Chaves.

gens, como na modificação de  $W_r$ , onde não é necessário que se pare o processo de controle de velocidade em cada alteração da velocidade de referência.

Os valores de  $W_r$  serão selecionados em doze chaves, onde cada uma pode assumir nível "1" ou "0". A figura 24 mostra os circuitos deste bloco.

O circuito é constituído de três pastilhas TTL 74367, sendo que cada unidade é constituída de seis "buffers" com a possibilidade de terceiro estado nas saídas. As pastilhas se apresentam divididas em dois grupos de "buffers", um com quatro e o outro com dois, onde ambos os grupos apresentam entradas de seleção separadas.

Aproveitando-se que o controlador de barramentos 8228, no microcomputador, é de tecnologia bipolar, portanto, aceita níveis TTL e que as pastilhas 74367 apresentam terceiro estado, as saídas dos "buffers" foram colocadas diretamente na barra de dados. Para a seleção destas pastilhas foram usados três bits da via de endereços e mais o sinal de controle MR (leitura de memória - MEMR).

A obtenção pelo processador da  $W_r$  segue os seguintes passos:

- 1) Seleção da velocidade de referência, acionando as doze chaves.
- 2) Pedido de interrupção, usando uma chave que está no bloco "sistema de interrupções".
- 3) A subrotina correspondente a este pedido de interrupção busca os dois bytes da informação, determinados pela combinação dos bits de endereço e sinal de controle MR.

O byte menos significativo que corresponde a seis "buffers" de um 74367 e mais dois, de outro, é colocado na barra de dados por qualquer instrução que acione os bits de endereço  $A_{15}$  e  $A_{13}$ , e mais o sinal de controle MR (leitura de memória). O endereço deste byte é A000H.

O byte mais significativo corresponde a dois grupos de quatro "buffers" de unidades de 74367 diferentes. Neste byte os quatro bits mais significativos estão zerados porque as necessidades no projeto são de um byte e meio (12 bits). E seu endereço de memória é C000H, determinado pelos bits  $A_{15}$  e  $A_{14}$ .

Na seleção das pastilhas foi usado o sinal de controle MR, porque dá maiores facilidades no trato com "software", pois qualquer instrução que tenha um operando na memória, ou que acione MR pode acessar a estes dados. Enquanto que se o sinal de controle usado fosse um "I/O R" (entrada) a única instrução que teria possibilidade de manipular estes dados é a instrução IN.

### 3.8. - Sistema de Interrupção

#### 3.8.a. - Aspectos gerais

A interrupção é importante em inúmeras aplicações de tempo real. O mais comum é haver várias fontes solicitando interrupções e que endereçam diretamente suas rotinas de tratamento (interrupções vetoradas). Também é usual neste tipo de interrupção, a prioridade de atendimento dos pedidos ser decidido por circuitos externos.

Um outro tipo seria o chamado "POLLED", que quando aceito o pedido de interrupção pula sempre para o mesmo endereço onde uma rotina comanda um teste em todas as portas para verificar de onde veio o pedido. Depois de determinada a fonte que

solicitou a interrupção desvia para a rotina correspondente. Este processo é lento em relação as interrupções vetores.

O microcomputador MCS-80 da INTEL apresenta somente um vetor interrupção, que durante um pedido endereça por "hardware" a posição de memória 38H, executando a instrução RST 7. Para a entrada de dados no microcomputador do sistema foi escolhido o processo de interrupções vetorados, o que implicou na necessidade de mais vetores de interrupções. Isto foi conseguido com modificações no programa monitor, deixando os espaços endereçados pelas instruções RST (reinício) com o início ou o desvio para subrotinas.

Quando uma interrupção é aceita, o ciclo de máquina de interrupção equivale a um "FETCH" (busca de instrução) e portanto a instrução RST deve estar preparada na barra de dados durante o segundo e terceiro estado do ciclo, que é quando o processador aceita entradas (DBIN = 1).

O bloco sistema de interrupções mostrado na figura 25, apresenta três vetores interrupções. Como os pedidos não são sincronizados entre si e portanto, podem ocorrer ao mesmo tempo, é necessário que se estabeleça níveis de prioridades.

Neste trabalho para a maior prioridade foi escolhida a entrada por chaves, já que são elas que definem ou redefinem a velocidade de referência de máquina. Logo um pedido de interrupção de chave deve ser atendido se ocorrer junto com uma de velocidade ou de corrente.

As interrupções de velocidade receberam a segunda prioridade porque a rotina que trata estas interrupções, está encarregada de tarefas como inicializar tempos de amostragens e zerar os contadores de frequência da máquina, e se isto não ocor

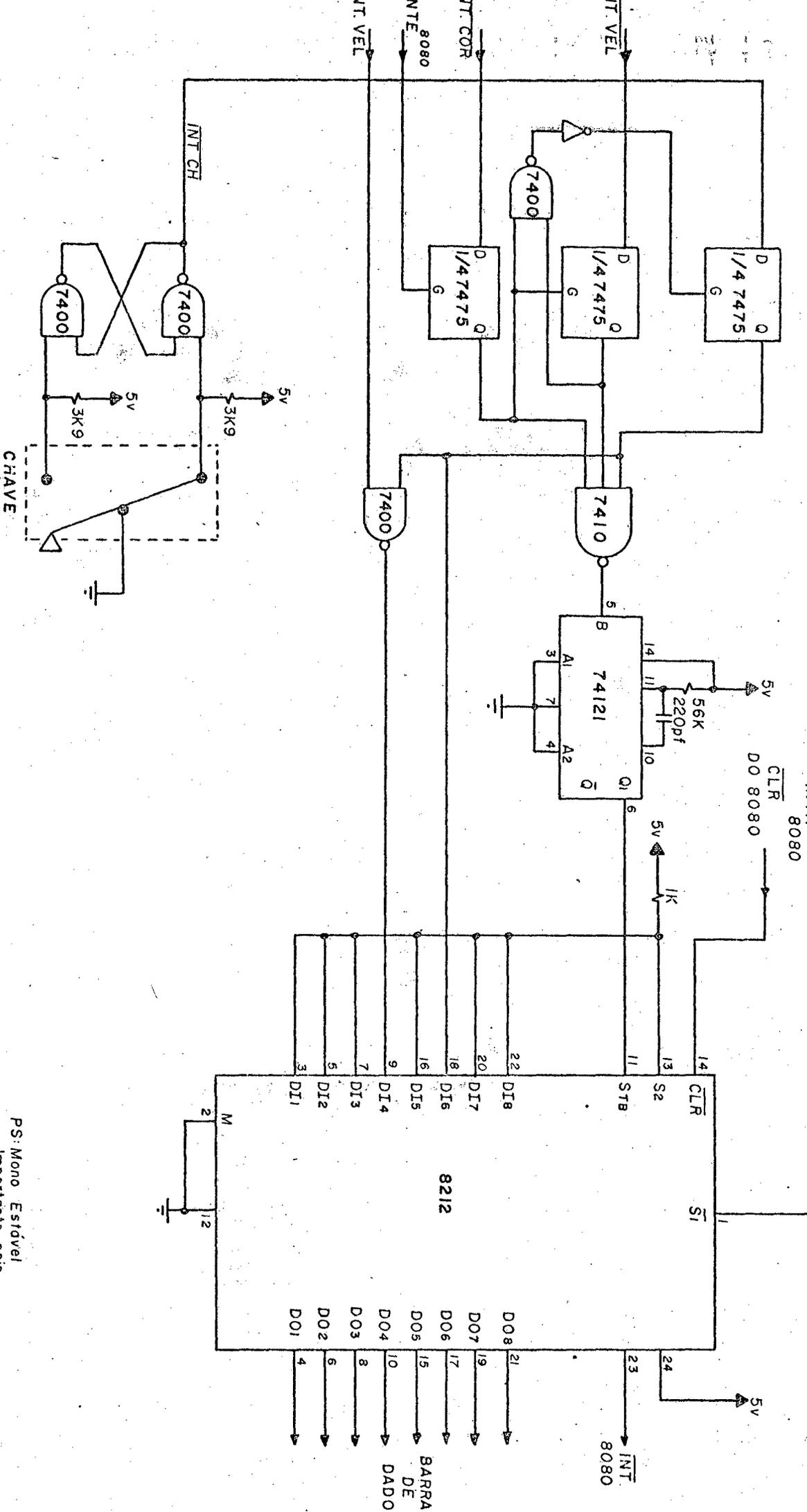


Fig. 25. Sistema de Interrupções

PS: Mono Estável  
 Importante, pois  
 STB engatilha na descida.

rer o processo estará sujeito a perturbações.

O menor nível de prioridade é o das interrupções de corrente. Os valores de corrente são usados em um limitador que tem tempo suficiente para corrigir os picos de corrente, pois entre duas interrupções de velocidade ocorrem oito para corrente.

### 3.8.b. - Descrição funcional do 8212

Na implementação deste bloco foi utilizado uma pastilha TTL 8212, que é constituída de uma memória de oito bits ("LATCH") na entrada ( $DI_1 - DI_8$ ), e na saída ( $D_{01} - D_{08}$ ) "buffers" com terceiro estado. O funcionamento deste dispositivo depende da combinação dos sinais da lógica de controle como  $S_1$ ,  $S_2$ , STB, M e  $\overline{CLR}$ . Esta pastilha apresenta também um flip-flop utilizado para armazenar os pedidos de interrupções.

A lógica de controle desta pastilha foi selecionada de modo que na chegada de um pulso na entrada STB a instrução de RST, que deve estar preparada na entrada, é armazenada na memória ("LATCH"). Este registro de entrada só aceita dados quando STB está em nível alto. Este mesmo sinal na descida zera a saída Q do flip-flop ("CLEAR"), que por sua vez é usada na entrada  $\overline{INT}$  do microcomputador, solicitando, desta maneira, interrupção (nível baixo). A saída ( $D_{01} - D_{08}$ ) é tirada do terceiro estado, acessando a barra de dados do microcomputador, quando aplicado nível "0" da entrada  $\overline{S_1}$ . Isto é feito pela  $\overline{INTA}$ , sinal da barra de controle do microcomputador, que notifica que o processador aceitou o pedido de interrupção. O sinal  $\overline{INTA}$  está disponível no início do segundo estado do ciclo de máquina de interrupção e portanto garante que a instrução de RST armazenada no registro do 8212 estará na barra de dados no momento que o processador estiver aceitando entrada nesta barra. A entrada  $\overline{SI}$  ( $\overline{INTA}$ ) quando a-

tivada (nível baixo) leva o sinal  $\overline{\text{INT}}$  novamente a nível "1", deixando o flip-flop pronto para receber novo pedido de interrupção.

Abaixo temos os sinais STB e  $\overline{\text{SI}}$  definindo o comportamento da pastilha no circuito.

STB	MEMÓRIA	SAÍDA (D <sub>01</sub> - D <sub>08</sub> )	$\overline{\text{INT}}$
0	—	Terceiro Estado	1
1	armazena	Terceiro Estado	1
	—	Terceiro Estado	0
$\overline{\text{SI}}$	MEMÓRIA	SAÍDA (D <sub>01</sub> - D <sub>08</sub> )	$\overline{\text{INT}}$
1	—	Terceiro Estado	0
0	—	na barra de dados	1

OBS: O sinal STB predomina sobre  $\overline{\text{SI}}$  no caso do flip-flop (INT).

### 3.8.c. - Descrição do bloco.

O bloco sistema de interrupções pode ser dividido em duas partes onde a primeira trata dos pedidos de interrupções. E na segunda parte simultaneamente aos pedidos são preparadas as instruções de RST correspondentes.

Na composição das RST e nos circuitos de pedidos de interrupção foram aproveitadas condições existentes em outros blocos.

No caso da velocidade o sinal, que em nível alto produz a base de tempo do frequencímetro, e na sua descida para nível baixo bloqueia as contagens, é perfeito para analisar o pedido de interrupção. Este sinal pode ser usado na composição da RST de velocidade, já que sinaliza duas situações distintas. Quando é "1" estão ocorrendo contagens, portanto, as interrupções que ocorrerem não são de velocidade, e quando é "0" a interrupção é de velocidade.

Este sinal, que é uma saída  $\overline{\text{Q}}$  de um flip-flop e

que neste bloco é chamada de  $\overline{\text{INTVEL}}$ , não pode ser aplicada diretamente a entrada STB do 8212, porque quando de sua descida o sinal  $\overline{\text{INT}}$  do 8212 é ativado (nível baixo) e necessariamente a RST correspondente a velocidade teria que ser preparada antes desta descida, pois a memorização no registro de entrada só ocorre com  $\text{STB} = 1$ . Esta mesma situação ocorre para os outros sinais de pedidos de interrupção, como os de corrente e os de chaves.

As interrupções de corrente são provocadas pelo sinal que indica o fim de conversão no conversor A/D (EOC). O pulso ( $\overline{\text{INTCOR}} = 0$ ) é saída de um monoestável disparado pelo sinal EOC.

Para interrupções de chaves o pedido ( $\overline{\text{INTCH}} = 0$ ) é feito usando a chave incluída neste bloco que quando ativada zera a saída Q do flip-flop SR formado por dois 7400.

O uso de um monoestável facilitou a preparação das instruções RST com alguns destes mesmos sinais utilizados nos pedidos de interrupção. Durante um pedido de interrupção o monoestável 74121 terá sua saída Q (STB) passando para nível "1" e a instrução RST já estará composta por este sinal que provocou o disparo do monoestável e portanto, em condições de ser memorizado no registro de entrada do 8212, quando  $\text{STB} = 1$ .

Na descida deste pulso do monoestável, a saída  $\overline{\text{INT}}$  do 8212 passa a nível baixo, portanto solicitando interrupção ao processador, e com a RST correspondente a este pedido já armazenada no registro.

O pulso do monoestável, com uma duração de  $8\mu\text{seg}$ , é disparado na subida pela soma dos três sinais de pedidos de interrupção, numa porta "NAND" TTL 7410. Os pedidos de interrupções que chegarem juntos serão classificados segundo suas prioridades de atendimento.

A finalidade dos flip-flops D é evitar que dois ou os três pedidos de interrupção, se sobreponham, produzindo só uma subida na saída do 7410 e, portanto, só um disparo do monoes tável. Sem os flip-flops, um pulso de pedido de interrupção, quando acionado, bloqueia a entrada do "NAND" 7410 pelo espaço de tempo correspondente a sua duração que varia de 10 mseg para as chaves até 80  $\mu$ seg para a corrente. Qualquer outro pedido que chegasse com um atraso pouco maior que 8  $\mu$ seg, de modo, a não poder mais impor o seu nível de prioridades, era perdido.

Com a colocação dos flip-flops D os sinais passaram a ter tratamentos diferentes.

Analisando primeiro o caso dos pedidos de interrupções das chaves. Verifica-se que o sinal produzido pela chave ( $\overline{\text{INT CH}}$ ) localizada neste bloco só chegará na entrada da porta 7410 se  $G = "1"$ . Como  $G$  neste flip-flop é dado por:

$$G = \overline{\text{INT COR}} \overline{\text{INT VEL}}, \text{ ("AND" entre as duas variáveis)}$$

Com isto, qualquer pedido de interrupção de menor prioridade ( $\text{INT COR} = 0$  ou  $\text{INT VEL} = 0$ ), chegando antes ao "NAND" bloqueia a passagem no flip-flop do pedido das chaves ( $\overline{\text{INT CH}} = 0$ ). Mas este pulso não se perde porque a sua duração é de 10 mseg, enquanto que,  $G$  permanece em nível baixo em 80  $\mu$ seg que são os tempos dos pulsos  $\overline{\text{INT COR}}$  e  $\overline{\text{INT VEL}}$  respectivamente.

Este procedimento evita, que se perca um pedido de interrupções de chaves, por chegar depois de um pedido de velocidade ou corrente, após o armazenamento da respectiva RST no registro do 8212.

Em caso contrário, ou seja, a chegada de um pedido das chaves antes ou junto de qualquer outro, por ter maior

prioridade deve ser executado o das chaves. Neste caso não há preocupação de salvar de possíveis bloqueios na porta 7410 aos pedidos de interrupção de corrente e de velocidade, porque o processo vai ter  $W_r$  mudada, redefinindo o seu comportamento.

Para interrupções de velocidade, o pulso  $\overline{\text{INT VEL}}$  (ativo em nível baixo), só terá a sua passagem permitida pelo seu respectivo flip-flop D, se instantes antes não houver ocorrido a chegada à porta 7410 do sinal  $\overline{\text{INT COR}}$  (ativo em nível baixo), pois  $G = \overline{\text{INT COR}}$ . O pedido de interrupção de velocidade também não se perde, porque a duração deste pulso depende da rotina que atende a esta interrupção. Esta rotina está encarregada de inicializar as contagens e enquanto isto não ocorre este sinal fica em nível baixo até no máximo 62,5 mseg. Normalmente a duração deste pulso é da ordem de 80  $\mu$ seg que é a duração da rotina que trata da velocidade, ou no caso de bloqueada pelo pedido de corrente é aproximadamente a soma da duração de  $\overline{\text{INT COR}} = 0$  mais a duração da rotina de interrupções de velocidade, dando pulsos da ordem de 150  $\mu$ seg.

No caso inverso, chegando antes ou junto os pulsos de  $\overline{\text{INT VEL}}$  com  $\overline{\text{INT COR}}$ , a interrupção atendida é a de velocidade.

Em pedidos de interrupção de corrente o sinal  $\overline{\text{INT COR}}$  tem sua passagem pelo flip-flop D controlada pelo microcomputador pois  $G = \text{INTE}$ , onde INTE corresponde a uma saída do processador, que avisa se as interrupções são ou não permitidas. Sempre que ocorre uma interrupção este sinal INTE assume nível baixo, avisando a não permissão, e só passará a nível alto, no caso, sinalizando interrupções permitidas, quando da execução de uma instrução EI (interrupção permitida) pelo processador. Logo

sempre que uma interrupção for aceita a passagem de  $\overline{\text{INT COR}}$  fica bloqueada no flip-flop D.

As rotinas que tratam as interrupções executam sempre instruções EI no seu final, portanto a duração do bloqueio equivale ao tempo de execução das rotinas e no caso da rotina de velocidade este tempo é de 80  $\mu$ seg.

Quando o pedido de interrupção de corrente chega depois do processador ter aceito uma de velocidade ( $\text{INTE} = 0$ ), o pedido para a corrente não se perde porque  $\overline{\text{INT COR}}$  tem duração de 80  $\mu$ seg.

#### 3.8.d. - Instrução RST

As interrupções vetoradas necessitam que junto aos pedidos de interrupção sejam endereçadas as suas rotinas de atendimento. E este endereçamento é feito usando um campo de 3 bits disponível nas instruções RST. Quando uma destas instruções é executada, o programa reinicia no endereço indicado pelo resultado do produto de oito com o conteúdo do campo de 3 bits (de 0 a 7). Portanto são oito os endereços que servem para as interrupções vetoradas: 00H, 08H, 10H, 18H, 20H, 28H, 30H, 38H.

A figura 25 mostra como as instruções RST são montadas nas entradas ( $\text{DI}_1 - \text{DI}_8$ ) da pastilha 8212. O campo de 3 bits que define o endereço para reinício corresponde as entradas  $\text{DI}_4$ ,  $\text{DI}_5$  e  $\text{DI}_6$ . Os bits restantes desta instrução apresentam o valor lógico 1, portanto, as entradas correspondentes estão ligadas a tensão 5V.

Para a montagem das RST, segundo as necessidades do projeto, onde são três os vetores interrupções, basta o uso de dois bits do campo variável, assumindo valores em função dos pedidos de interrupção. No caso os bits usados foram  $\text{DI}_4$  a  $\text{DI}_6$ .

O bit  $DI_5$  é mantido no nível lógico "1".

Dos circuitos de pedidos de interrupção se obtêm:

$\overline{INT\ VEL} = 0$  pedido de interrupção para velocidade

$\overline{INT\ COR} = 0$  pedido de interrupção para corrente

$\overline{INT\ CH} = 0$  pedido de interrupção para chave

Com base nisto e nos circuitos da figura 24, são definidas as seguintes funções:

$DI_4 = \overline{INT\ VEL} + INT\ CH$

$DI_5 = 1$

$DI_6 = \overline{INT\ CH}$

E com estas funções, as interrupções vetores indicam os seguintes endereços:

a) Interrupção de corrente

com  $\overline{INT\ VEL} = 1$ ,  $\overline{INT\ COR} = 0$ ,  $\overline{INT\ CH} = 1$

$DI_6 = 1$ ,  $DI_5 = 1$ ,  $DI_4 = 1 \implies 7$

endereço: 38H (8 · 7)

b) Interrupção de velocidade

$\overline{INT\ VEL} = 0$ ;  $\overline{INT\ COR} = 1$ ;  $\overline{INT\ CH} = 1$

$DI_6 = 1$ ,  $DI_5 = 1$ ,  $DI_4 = 0 \implies 6$

endereço: 30H (8 · 6)

c) Interrupção de chave

$\overline{INT\ VEL} = 1$ ,  $\overline{INT\ COR} = 1$ ,  $\overline{INT\ CH} = 0$

$DI_6 = 0$ ,  $DI_5 = 1$ ,  $DI_4 = 1 \implies 3$

endereço: 18H (8 · 3)

### 3.8.e. - Níveis de prioridades

Se analisadas as funções estipuladas para  $DI_4$ ,  $DI_5$ ,  $DI_6$ , quando ocorrerem os três pedidos de interrupção no mesmo instante, ou seja,  $\overline{INT\ CH} = 0$ ,  $\overline{INT\ VEL} = 0$  e  $\overline{INT\ COR} = 0$ , verifica-se que:

$$DI_4 = 1, DI_5 = 1 \text{ e } DI_6 = 0,$$

o que resulta no endereço 18H, que é a posição de memória endereçada pelo vetor interrupção das chave. Logo este vetor tem o mais alto nível de prioridades.

Se ocorrerem os pedidos de interrupção de corrente e velocidade no mesmo instante, ou seja,  $\overline{INT\ VEL} = 0$  e  $\overline{INT\ COR} = 0$ , verifica-se pelas funções que:

$$DI_4 = 0, DI_5 = 1 \text{ e } DI_6 = 1.$$

O que resulta no endereço 30 H, que é a posição de memória endereçada pelo vetor interrupção de velocidade e com isto fica garantido para este vetor o segundo nível de prioridades.

Os resultados obtidos com este bloco foram satisfatórios.

A família do 8080 apresenta uma pastilha 8214 que é usado para classificar em níveis de prioridades os pedidos de interrupções e gerar conforme esta classificação o campo de 3 bits da instrução RST, que endereça as rotinas que tratam destas interrupções.

Pelo preço esta unidade é justificável para sistemas com grande número de vetores interrupções.

### 3.9. - Programa de controle do sistema

#### 3.9.a. - Considerações gerais

O programa de controle deste sistema apresenta duas partes distintas. Uma é responsável pela aquisição de dados e a outra é composta pelos algoritmos que processam estes dados quase que simultaneamente às suas entradas no microcomputador. Ficando, portanto caracterizado um sistema em tempo real.

Este programa começa definindo as portas de en-

tradas e saídas do microcomputador e dá partida ao processo de controle, enviando valores iniciais aos circuitos externos. A seguir o processador executa uma instrução de "halt", entrando em estado de espera ( $T_w$ ).

As entradas no microcomputador da velocidade medida, da corrente de armadura e da velocidade de referência são feitas usando três vetores interrupções, que endereçam suas respectivas rotinas de aquisição. A seguir serão discutidos aspectos ligados às três entradas de dados.

As interrupções de velocidade tiram o processador do estado de espera a cada 62,5 mseg. Na rotina endereçada por estas interrupções os valores de velocidade medidos são enviados das portas de entrada para suas posições de memória, os contadores de frequência são zerados e é inicializado novo período de contagens no tacômetro digital. No retorno ao programa principal é iniciado a execução do algoritmo de velocidade, cuja duração é de aproximadamente 150  $\mu$  segundos.

O tempo de execução do algoritmo de velocidade é muito pequeno se comparado com 62,5 mseg, que é o tempo entre duas entradas de valores de velocidades consecutivas.

Portanto, o dado N, na saída do microcomputador pode ser variado em instantes discretos de tempo, caracterizando um sistema amostrado, com período de amostragem de 62,5 mseg. Depois da execução do algoritmo de velocidade o microcomputador entra em estado de espera ao executar novamente uma instrução de "halt".

As interrupções de corrente ocorrem a cada 8,33 mseg e nestas interrupções são feitas as buscas de valores da corrente de armadura. Quando do retorno ao programa principal o

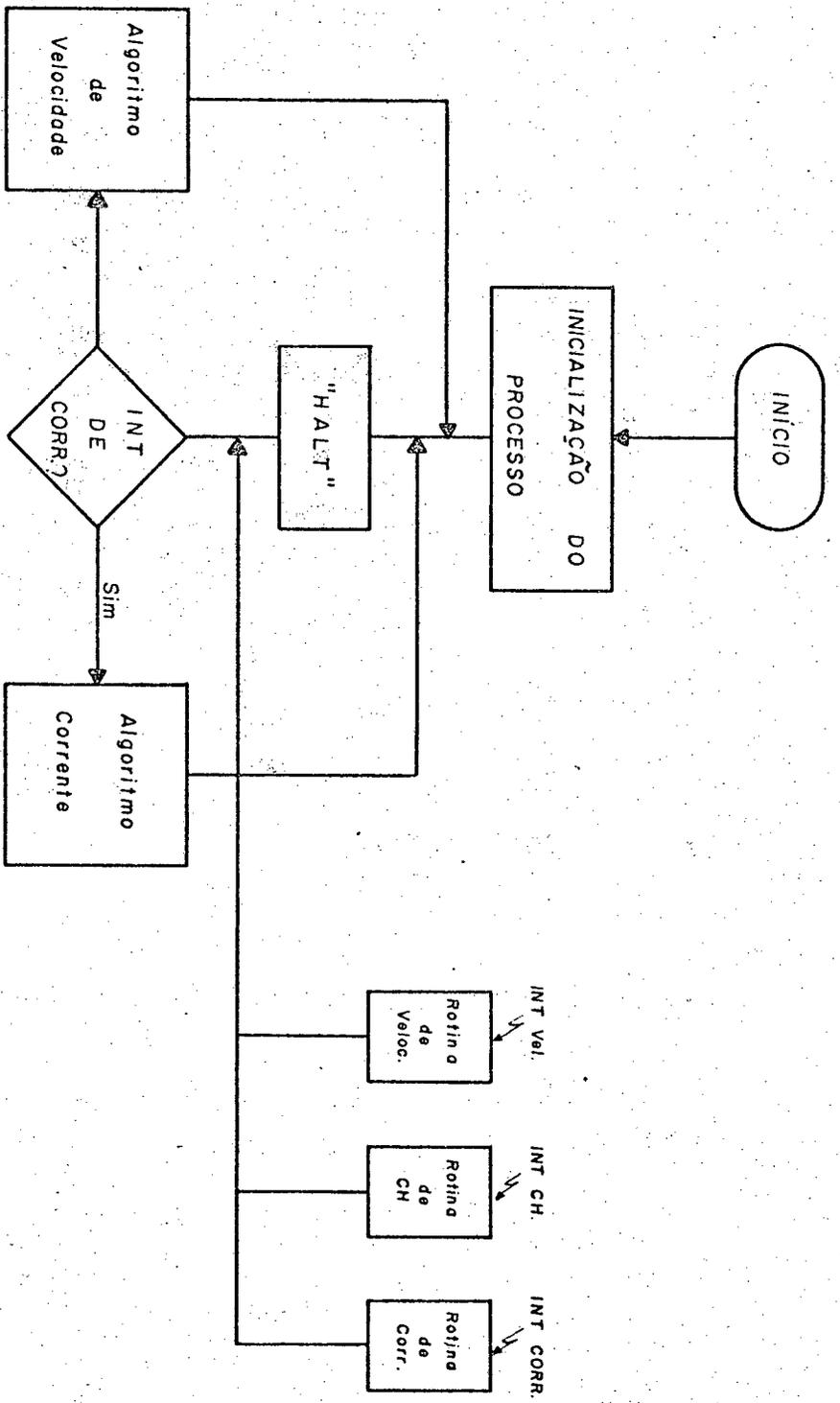


Diagrama simplificado do programa de controle.

Fig. 26

microprocessador executará o algoritmo de corrente, cuja finalidade será comparar a corrente medida com o valor escolhido para corrente máxima na armadura, e conforme o resultado da comparação haverá ou não alteração do número N na entrada do gerador de pulsos. No fim deste algoritmo o microprocessador volta ao estado de espera com a execução de um "halt".

Os pedidos de interrupção de chaves, ocorrem de maneira totalmente aleatória. A rotina endereçada nestas interrupções, é responsável pela busca da velocidade de referência. Como estes valores definem novo comportamento ao sistema, na volta ao programa principal o processador deve executar o algoritmo de velocidade.

A figura 26 mostra o diagrama simplificado do "software" do sistema.

### 3.9.b. - Algoritmo de velocidade

Este algoritmo deve, a partir do erro de velocidade modificar a informação digital N entrada no gerador de pulsos. A velocidade atual é comparada com a velocidade de referência, e o resultado desta comparação determinará se N será aumentado ou diminuído.

As três possibilidades desta comparação e suas consequências serão:

— A velocidade de referência e velocidade atual são iguais; N deve permanecer inalterado.

— A velocidade de referência é maior que a velocidade atual; N deve ser diminuído. Com isto, ocorre uma diminuição de  $\theta$  (ângulo de disparo da ponte), um aumento na tensão de armadura, maior corrente de armadura e por consequência maior velocidade do motor de corrente contínua.

- A velocidade de referência é menor que a velocidade atual; N deve ser aumentado. Portanto,  $\theta$  aumenta, a tensão da armadura diminui, a corrente também e por consequência a velocidade da máquina diminui.

Nos dois casos, tanto no sentido de aumento de velocidade como no de diminuição, o erro de velocidade será testado para três faixas de erros onde N será incrementado ou decrementado com valores diferentes.

São definidos dois parâmetros  $K_2$  e  $K_1$  onde  $K_1 < K_2$ . E os incrementos ou decrementos  $N_1$ ,  $N_2$  e  $N_3$  onde  $N_1 < N_2 < N_3$ . Com estes parâmetros e valores de incrementos ou decrementos determinados adequadamente temos:

- Para  $|\text{erro de velocidade}| < K_1$ , N é incrementado ou decrementado, se for o caso, por  $N_1$ .

- Para  $K_1 < |\text{erro de velocidade}| < K_2$ , N é incrementado ou decrementado por  $N_2$ .

- Para  $K_2 < |\text{erro de velocidade}|$ , N é incrementado ou decrementado por  $N_3$ .

Este algoritmo tem um comportamento de um compensador integral com ganho variável.

### 3.9.c. - Algoritmo de corrente

A finalidade deste algoritmo é limitar a corrente de armadura, a um determinado valor. Se o valor medido da corrente de armadura superar a um outro escolhido como máximo, este algoritmo incrementará N, o ângulo de disparo dos tiristores ( $\theta$ ), com isto, aumentará, resultando numa diminuição da tensão de armadura e por consequência a corrente diminui.

Um incremento na tensão de armadura pode ocorrer em uma interrupção de velocidade e se para este incremento a



corrente de armadura superar o valor permitido, o algoritmo de corrente atuará sobre N, de modo a diminuir a tensão de armadura, no máximo oito vezes, trazendo a corrente para a faixa permitida, antes que ocorra outra interrupção de velocidade.

As alterações de N em relação ao algoritmo de corrente só ocorrem em situações especiais, como variações bruscas de carga no motor. Em regime pode-se considerar o período de amostragem do sistema discreto como sendo 62,5 mseg, período este estipulado pelas interrupções de velocidade.

A figura 27 mostra o diagrama em blocos geral do programa de controle do sistema.

### 3.9.d. - Listagem do Programa de Controle

#### Programa Principal

##### Iniciação do Processo

1000	F3	DI		; Interrupção não permitido.
1001	3E91	MVI	A, 91 H	; Palavra de controle da 8255.
1003	D3F7	OUT		; Envio da palavra de controle.
1005	3E83	MVI	A, 83 H	; Palavra de controle da 8255.
1007	D3EF	OUT		; Envio da palavra de controle.
1009	3E50	MVI	A, 50 H	;
100B	D3F6	OUT		; Zerar contadores
100D	1E01	MVI	E, 01 H	; Variável "E" = 01.
100F	2601	MVI	H, 01 H	; Variável "H" = 01.
1011	D5	PUSH	D	; Armazenar variável "E" no "Stack".
1012	3EA0	MVI	A, A0 H	;
1014	325013	STA	NMEM	; Armazenar "N" na memória.
		OUT		
1017	D3F5	OUT	N	; Enviar "N" para gerador de pulsos.
1019	FB	EI		; Interrupção permitida.

101A	76	HLT		; "Halt" (estado de espera).
				Teste para execução dos Algoritmos
101B	3E00	MVI	A, 00 H	;
101F	BD	CMPL		; Teste da variável L.
1020	C25410	JNZ	ALCOR	; Desvio para algoritmo da corrente.
			ALVEL	
1023	3A6013	LDA	VH	;
1026	57	MOV	D, A	;
1027	3A6113	LDA	VL	; Carregar o par de registradores
102A	5F	MOV	E, A	; com o valor da velocidade atual.
102B	79	MOV	A, C	;
102C	93	SUB	E	; Subtração entre velocidade de
102D	5F	MOV	E, A	; referência (par BC) e velocidade
102E	78	MOV	A, B	; de atual (par DE), resultado no
102F	9A	SBB	D	; par DE.
1030	57	MOV	D, A	;
1031	DA8F10	JC	ALTA	; Velocidade atual maior que a ve
				locidade de referência.
			BAIXA	
1034	3E00	MVI	A, N1H	; Compara erro com N <sub>1</sub> .
1036	BA	CMP	D	; Reg. D comparado com N1H.
1037	CA6A10	JZ	TSN1L	; Desvio para teste de N1L.
103A	DA7010	JC	TSN2H	; Desvio para teste de N2H.
103D	D1	POP	D	; Retirar do "Stack" variável "E".
103D	1D	DCR	E	; Decrementar "E".
103F	C26610	JNZ	SA1	; Saída do Algoritmo de velocidade.
1042	1E04	MVI	E, 04 H	; "E" = 4.
			DECR1	
1044	D5	PUSH	D	; Colocar "E" no "Stack".

1045	3A5013	LDA	NMEM	; Colocar "N" no acumulador.
1048	D601	SUI	DEC1	; Decrementar "N" de "dec 1".
		PROT I		
104A	D24F10	JNC	SA1	; Teste para proteção inferior.
104D	3E05	MVI	A, 05 H	; Proteção inferior "N" = 05.
104F	325013	STA	NMEM	; Armazenar "N".
		ALCOR		
1052	2E01	MVI	L, 01 H	; "L" = 1.
1054	3A5A13	LDA	"IA"	; "IA" no acumulador.
1057	FE14	CPI	IMAX	; Comparar corrente com valor máximo.
1059	3A5013	LDA	NMEM	; "N" no acumulador.
105C	D21710	JNC	OUT	; Desvio para "OUT"
105F	3C	INR	A	; "N" = "N" + 1.
1060	325013	STA	NMEM	; Armazenar "N".
1063	C31710	JMP	OUT	; Desvio para "OUT".
		SA1		
1066	D5	PUSH	D	; Armazenar "E".
1067	C35210	JMP	ALCOR	; Desvio para algoritmo de corrente.
		TSN1L		
106A	3E05	MVI	A, 05 H	;
106C	BB	CMP	E	; Compara erro (Rec. E) com "N1L".
106D	D23D10	JNC	DECR1	; Desvio para DECR1.
		TSN2H		
1070	3E10	MVI	A, 10 H	;
1072	BA	CMP	D	; Compara erro (Reg. D) com "N2H".
1073	CA8110	JZ	TSN2L	; Desvio para teste com "N2L".
1076	DA8710	JC	DECR3	; Desvio para DECR3.
		DECR2		
1079	3A5013	LDA	NMEM	; "N" no acumulador

107C	D601	SUI	DEC2	; Decrementar "N" de "dec 2".
107E	C34A10	JMP	PROT I	; Desvio para PROT I
	TSN2L			
1081	3E00	MVI	A, 00 H	;
1083	BB	CMP	E	; Compara erro (Reg. E) com "N2L".
1084	D27910	JNC	DECR2	; Desvio para DECR2.
	DECR3			
1087	3A5013	LDA	NMEM	; "N" no acumulador.
108A	D607	SUI	DEC3	; Decrementar de "N" "dec 3".
108B	C34A10	JMP	PROT I	; Desvio para PROT I
	ALTA			
108F	3E00	MVI	A, 00 H	;
1091	93	SUB	E	;
1092	5F	MOV	E, A	;
1093	3E00	MVI	A, 00 H	;
1095	9A	SBB	D	;
1096	57	MOV	D, A	; Complemento 2 do erro.
1097	3E00	MVI	A, 00 H	; N'1H no acumulador.
1099	BA	CMP	D	; Compara erro (Reg D) com N'1H.
109A	CABC10	JZ	TSN'1L	; Desvio para TSN'1L.
109D	D25210	JNC	ALCOR	; Desvio para algoritmo de corrente.
	TSN'2H			
10A0	3E00	MVI	A, 00 H	;
10A2	BA	CMP	D	; Compara erro (Reg D) com N'2H.
10A3	CAC510	JZ	TSN'2L	; Desvio para TSN'2L.
10A6	DACB10	JC	INCR2	; Desvio para INCR2.
	INCR1			
10A9	25	DCR	H	; Decrementar "H".
10AA	C25210	JNZ	ALCOR	; Desvio para algoritmo de corrente.

10AD	2605	MVI	H, 05 H	; "H" = 5.
10AF	3A5013	LDA	NMEM	; "N" no acumulador
10B2	C601	ADI	INC1	; "N" é incrementado de "inc 1".
		PROT S		
10B4	D24F10	JNC	ALCOR	; Desvio para ALCOR.
10B7	3EA0	MVI	A, A0 H	; Proteção Superior.
10B9	C34E10	JNC	ALCOR	; Desvio para ALCOR
		TSN'1L		
10BC	3E28	MVI	A, 28 H	;
10BE	BB	CMP	E	; Compara erro (Reg. E) com N'1L.
10BF	D25210	JNC	ALCOR	; Desvio para ALCOR.
		TSN'2L		
10C5	3EA0	MVI	A, A0 H	;
10C7	BB	CMP	E	; Compara erro (Reg. E) com N'2L.
10C8	D2A910	JNC	INCR1	; Desvio para INCR1
		INCR2		
10CB	3A5013	LDA	NMEM	; "N" no acumulador
10CE	C601	ADI	INC2	; "N" é incrementado de "inc"2.
10D0	C3B410	JMP	PROT S	; Desvio para PROT S.

#### Rotina da Velocidade

1300	F5	PUSH	PSW	; Salvar acumulador e "Flags".
1301	DBF6	IN		; Entrada velocidade (parte alta).
1303	E60F	ANI	OFH	; Máscara 4 bits.
1305	326013	STA	VH	; Armazenar na Posição VH.
1308	DBF4	IN		; Entrada velocidade (parte baixa).
130A	326113	STA	VL	; Armazenar na posição de memória VL.
130D	3E10	MVI	A, 10 H	;
130F	D3F6	OUT		; Zerar contadores

```

1311 3E00 MVI A, 00 H ;
1313 D3F6 OUT ; Iniciar período de contagens.
1315 2E00 MVI L, 00 H ; "L" = 0
1317 F1 POP PSW ; Restaure acumulador e "flags".
1318 FB EI ; Permitida interrupção.
1319 C9 RET ; Retorno.

```

#### Rotina da Corrente

```

1200 F5 PUSH PSW ; Salvar acumulador e "flags"
1201 DBED IN ; Entrada da corrente.
1203 325A13 STA IA ; Armazenar na memória na posição "IA".
1206 F1 POP PSW ; Restaurar "flags" e acumulador.
1207 FB EI ; Interrupção permitida.
1208 C9 RET ; Retorno.

```

#### Rotina das Chaves

```

1210 F5 PUSH PSW ; Salvar acumulador e "flags".
1211 E5 PUSH H ; Salvar variáveis "H" e "L"
1212 2100C0 LXI H, 00C0H;
1215 46 MOV B, M ; Colocar em B velocidade de referência
; (parte alta)
1216 2100A0 LXI H, 00A0H;
1219 4E MOV C, M ; Velocidade de ref. (parte baixa) em C.
121A 3200 MOV A, 00 H ;
121C D3F6 OUT ; Inicializar contagens.
121E E1 POP H ; Restaurar "H" e "L".
121F F1 POP PSW ; Restaurar acumulador e "flags".
1220 FB EI ; Permitida interrupção.
1221 C9 RET ; Retorno

```

1360	VH	EQU	1360H	;
1361	VL	EQU	1361H	; Posição de memória para vel. atual.
1350	NMEM	EQU	1350H	; Posição de "N" na memória
135A	IA	EQU	135AH	; Posição de memória para corrente de armadura.

## CAPITULO IV

### RESULTADOS

O motor de corrente contínua utilizado neste sistema de controle tem a velocidade angular nominal de 1800 rpm e corrente nominal de 8 A. Os resultados obtidos, levando-se em conta que não foi objetivo neste trabalho o desenvolvimento de algoritmos de controle mais elaborados, pode-se considerar bons. As figuras de 28 a 40 mostram as curvas velocidade X tempo e corrente X tempo na partida do motor a vazio e com carga e em resposta uma variação de carga em forma de degrau.

Nas curvas corrente X tempo o efeito do limitador de corrente durante a partida do motor aparece de uma maneira bem clara. Limitou-se a corrente no valor nominal.

A montagem utilizando na parte de acionamento uma ponte de tiristores apresentou certas desvantagens. Quando a velocidade da máquina supera a velocidade de referência o motor precisa ser frenado e portanto, há a necessidade de um torque em sentido contrário e para isto, de uma corrente em sentido contrário na armadura. Como a ponte não é reversível (corrente só num sentido), a redução da velocidade depende quase que só do atrito na máquina e da carga.

Nesta mesma situação de velocidade atual maior que a velocidade de referência, os incrementos dados a  $N$  sofrem um cuidado especial, o aumento do ângulo de disparo deve seguir vagarosamente a variação da velocidade da máquina, para evitar oscilação e instabilidade do sistema.

Um outro problema da montagem, usando ponte de tiristores, está ligado com o funcionamento em condução descontínua do retificador, para pequenos valores de  $\theta$ . Nestes casos po-

dem ocorrer falta de corrente durante um certo intervalo de tempo (ver figura 31).

O processo é bastante lento sobretudo devido a não reversibilidade da montagem. Os tempos de estabilização para variações de carga ou na partida do motor são de valores entre dois a quatro segundos. Observando as curvas que se seguem, conclui-se que a faixa de bom funcionamento do sistema é para valores entre 500 e 2000 rpm. Para valores maiores de velocidade o funcionamento melhora (ondulação, tempo de resposta).

As figuras 28 e 29 mostram respectivamente o motor em 1750 rpm, com partida a vazio, e a corrente correspondente a esta velocidade, onde o efeito do limitador na partida do motor é bastante visível. A escala é 0,65 cm/seg.

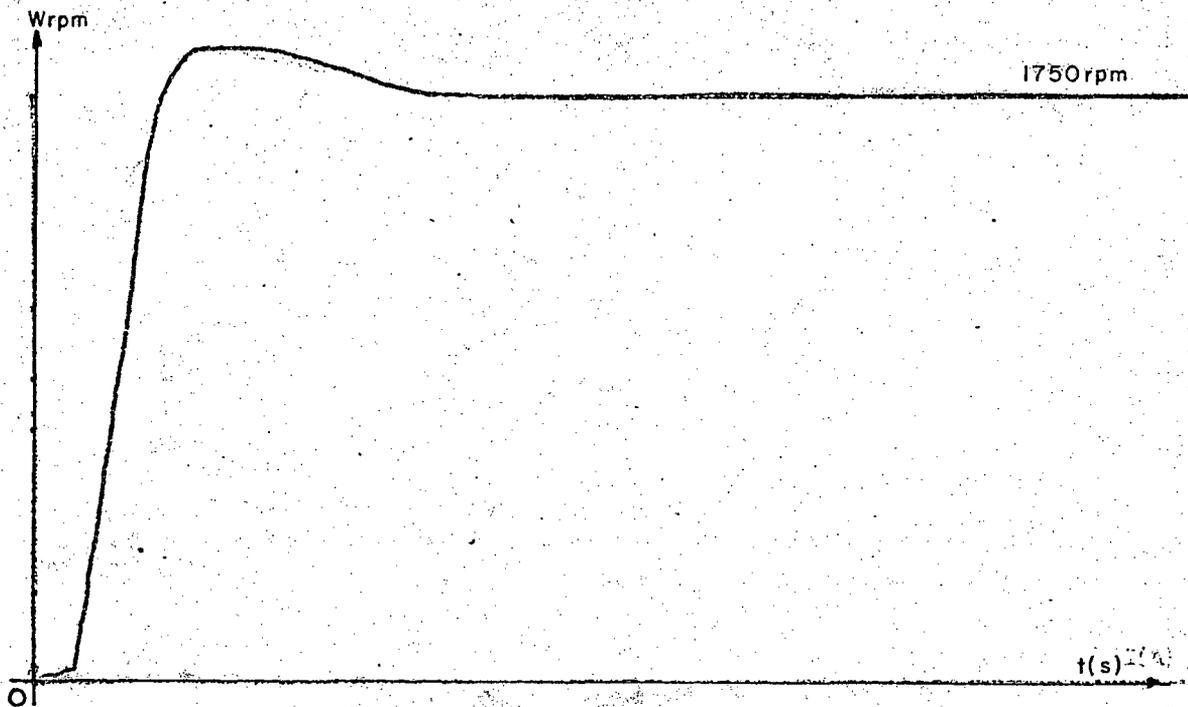


fig. 28

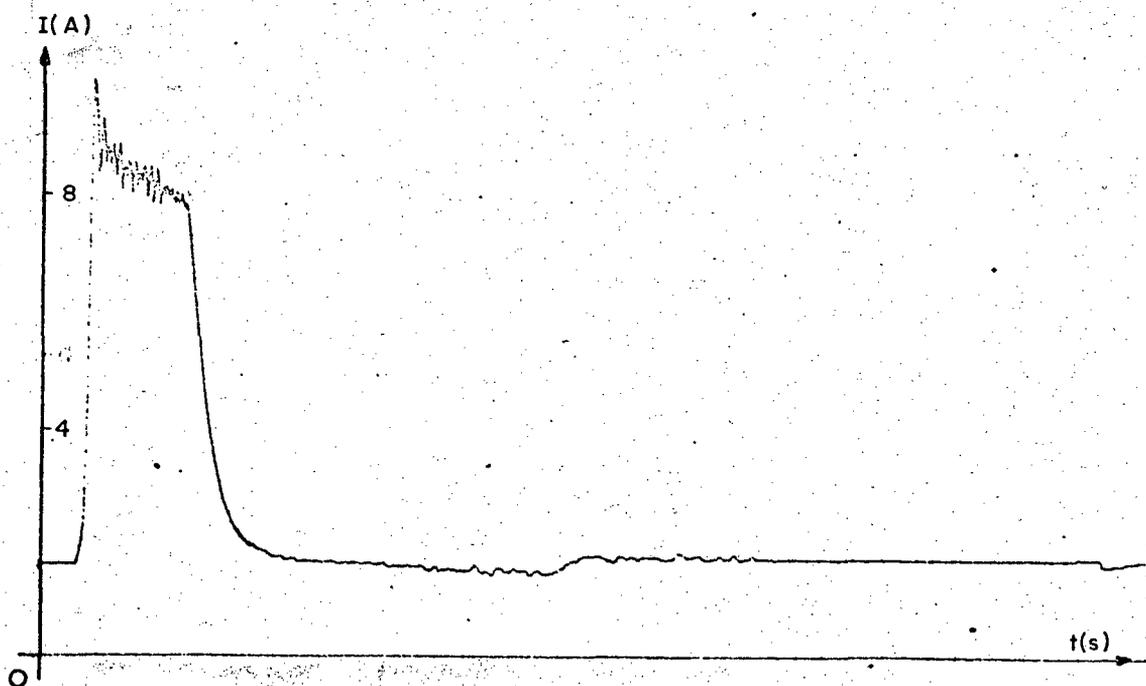


fig. 29

As figuras 30 e 31 mostram o motor para velocidade de 1500 rpm com a partida a vazio. Nesta caso ocorre descontinuidade na corrente para  $\theta$  pequeno. Escala de 0,65 cm/seg.

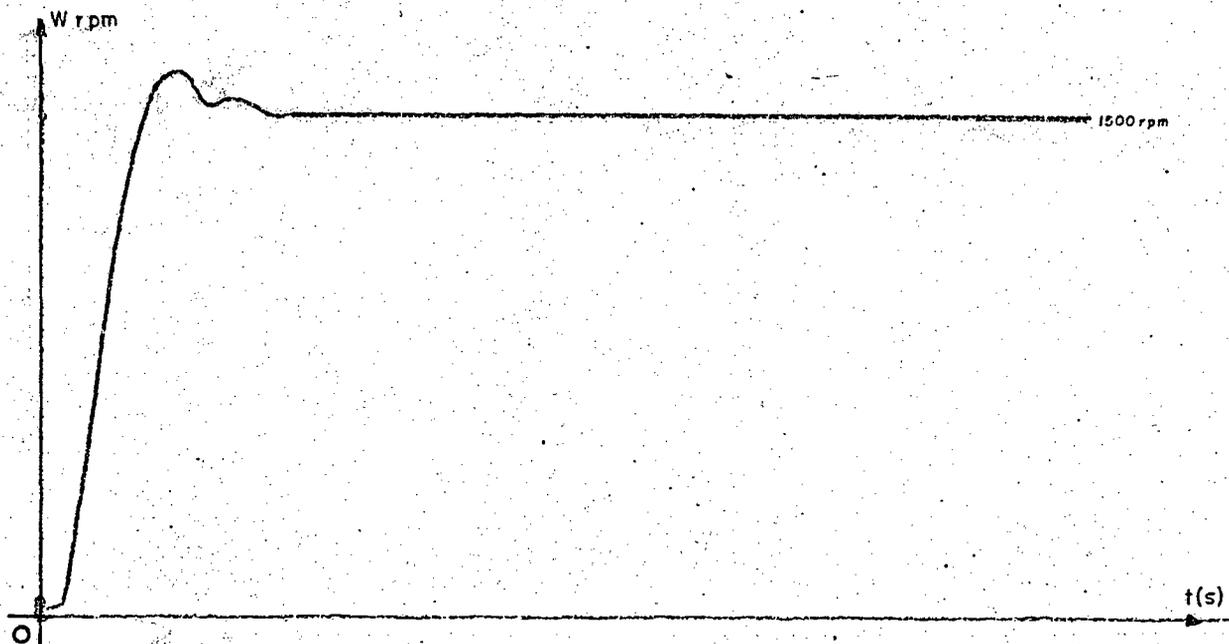


fig. 30

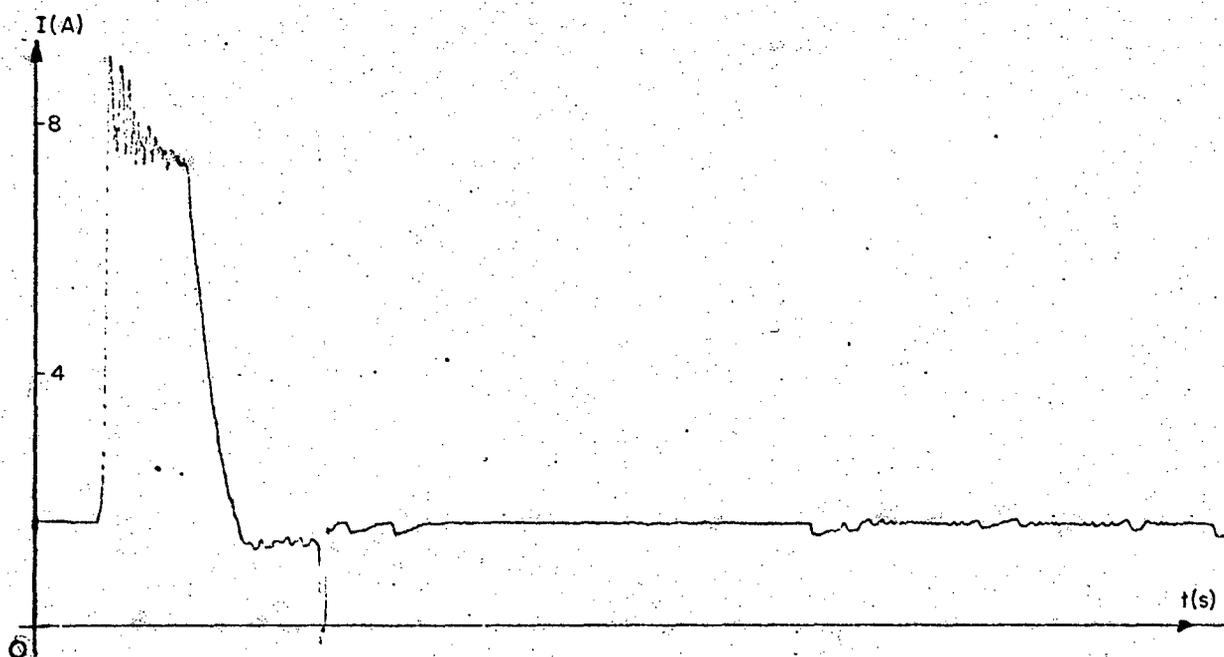


fig. 31

Nas figuras 32 e 33 aparecem as curvas de velocidade e corrente para a velocidade de 1000 rpm com partida a vazio. Escala de 0,65 cm/seg.

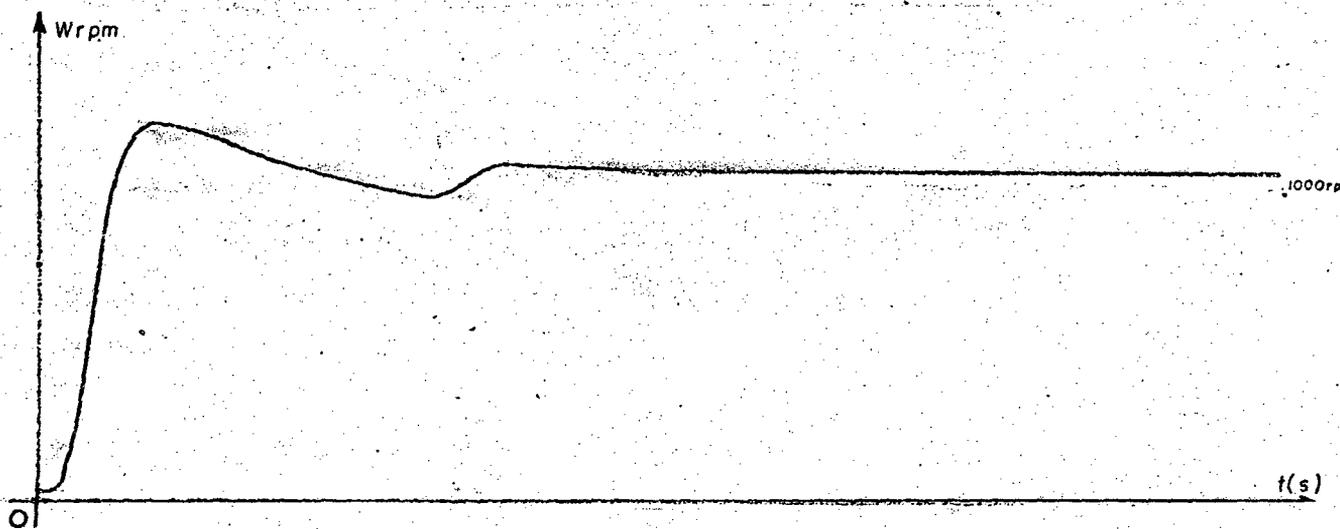


fig. 32

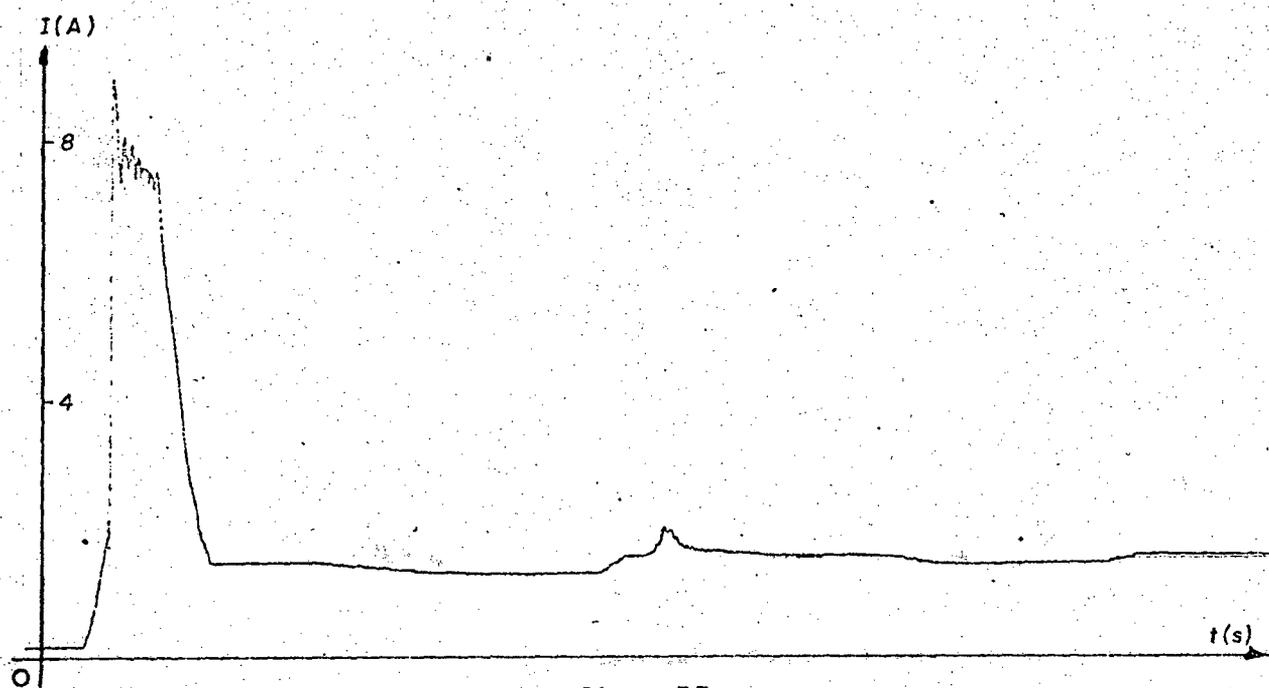


fig. 33

Nas figuras 34 e 35 aparecem as curvas de velocidade e corrente respectivamente, na partida com carga. A velocidade é de 1500 rpm. (Corrente na armadura 5 A). Escala de 1,3 cm/seg.

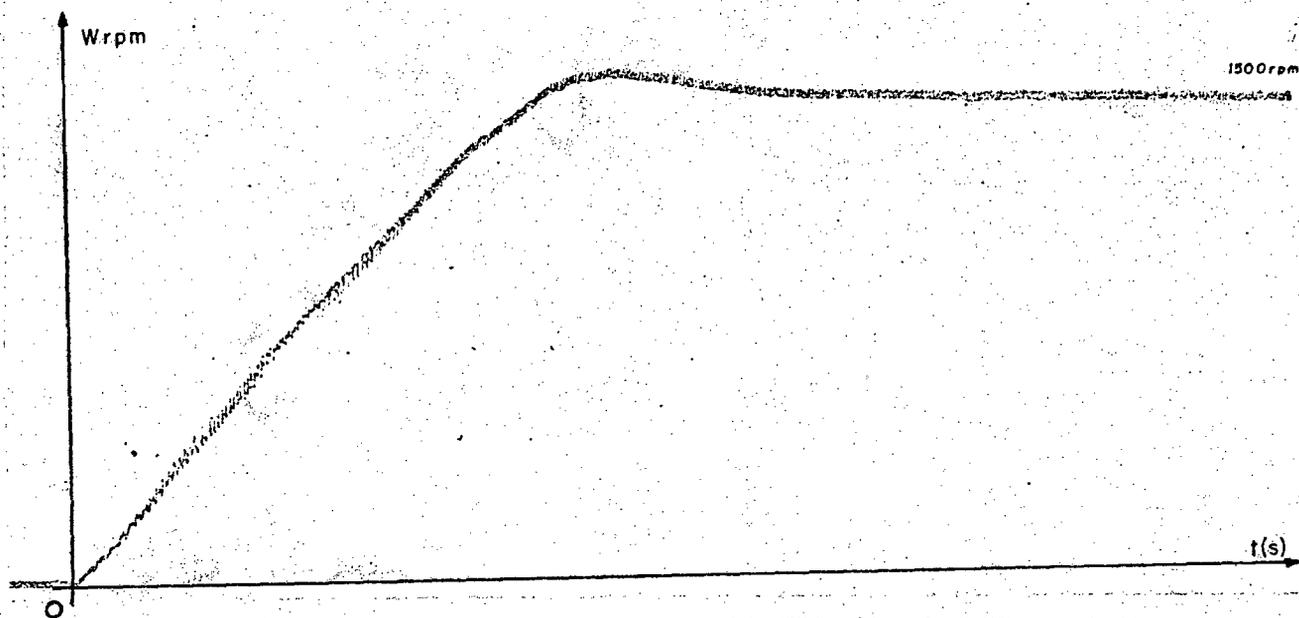


Fig. 34

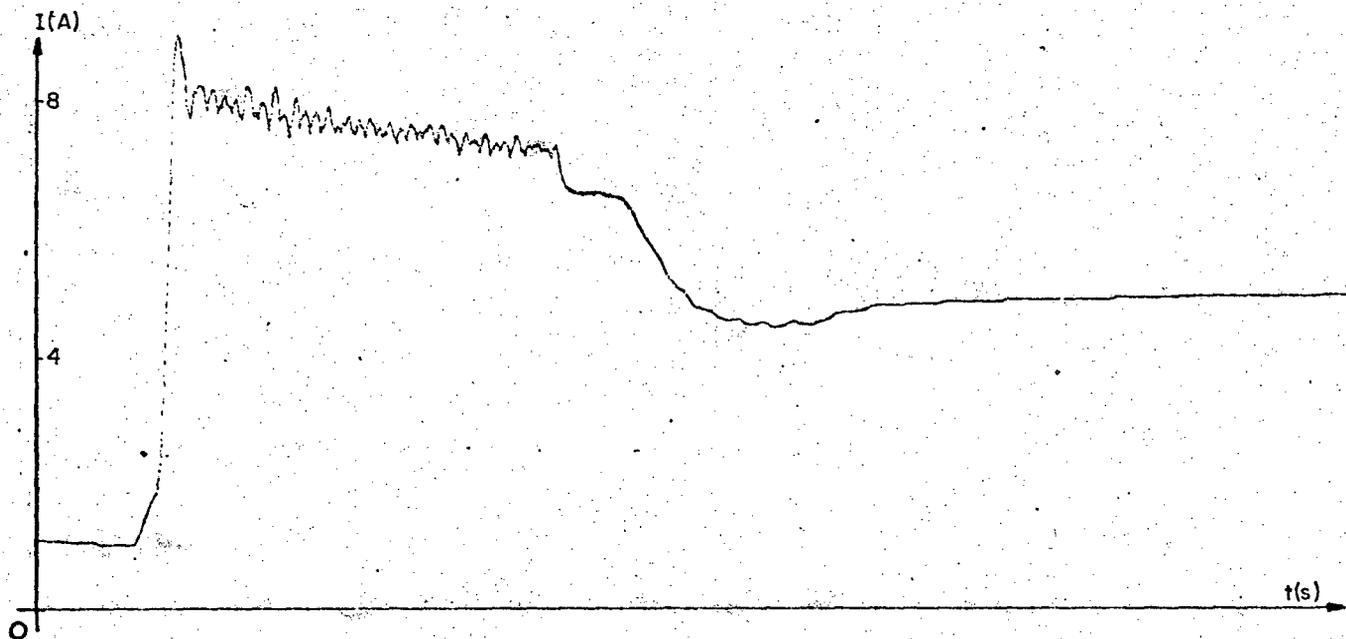


Fig. 35.

As figuras 36 e 37 mostram as curvas de velocidade de e corrente de armadura respectivamente na partida com carga para velocidade de 1000 rpm (Corrente na armadura 5 A). Escala de 1,3 cm/seg.

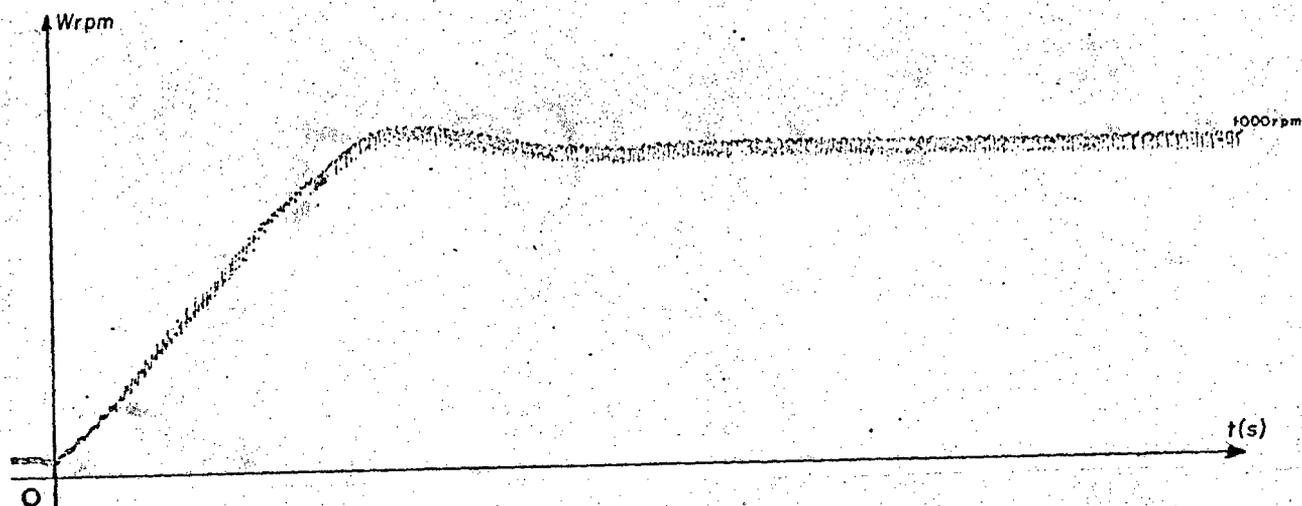


Fig. 36.

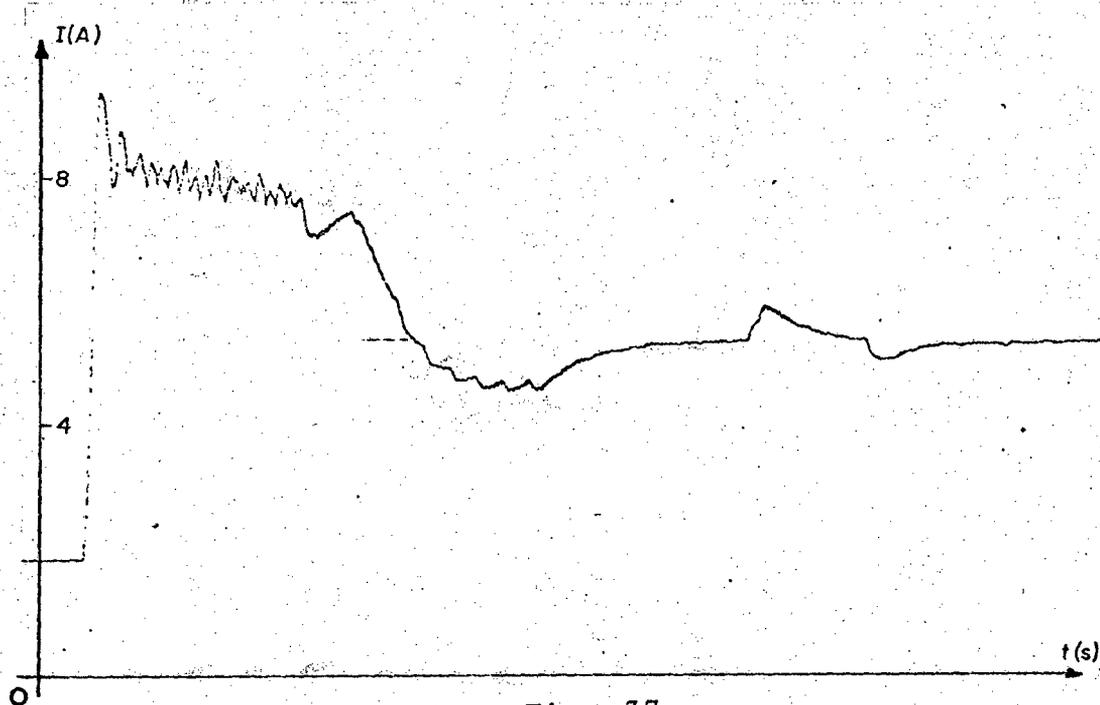


Fig. 37.

As figuras 38, 39 e 40 apresentam a resposta da máquina a um degrau de carga tal que a corrente de máquina seja de 5A em regime permanente, para velocidades de 500, 1000 e 1500 rpm. A escala é de 0,65 cm/seg.

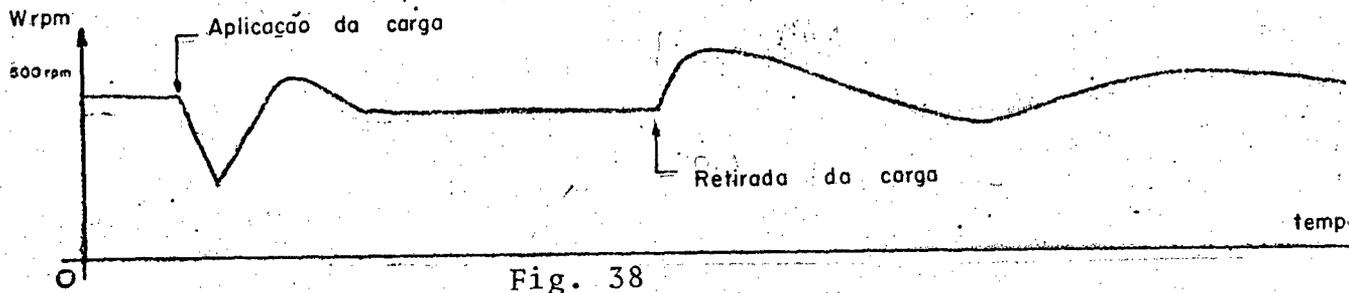


Fig. 38

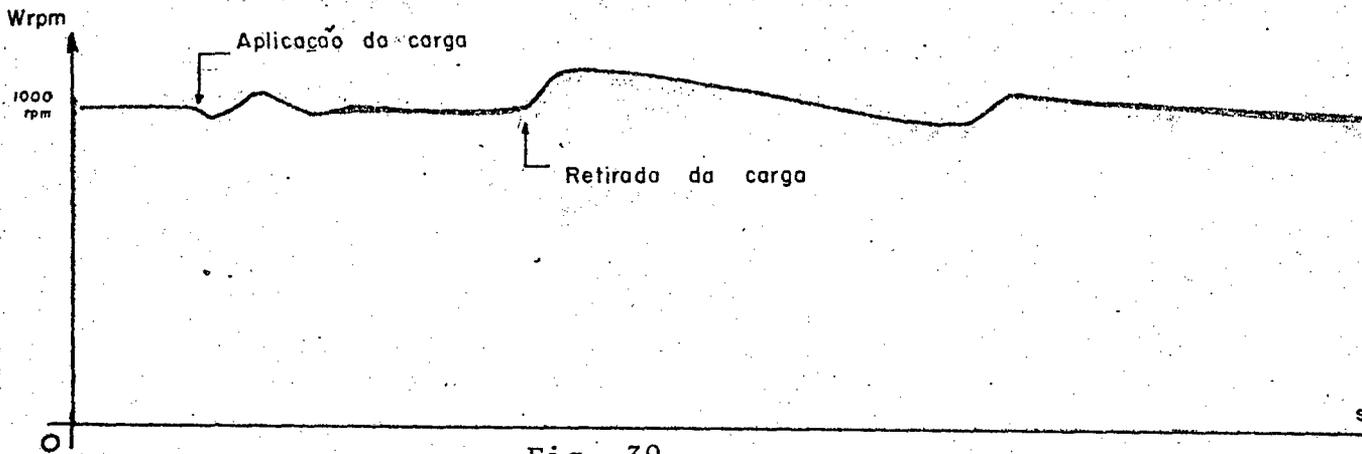


Fig. 39

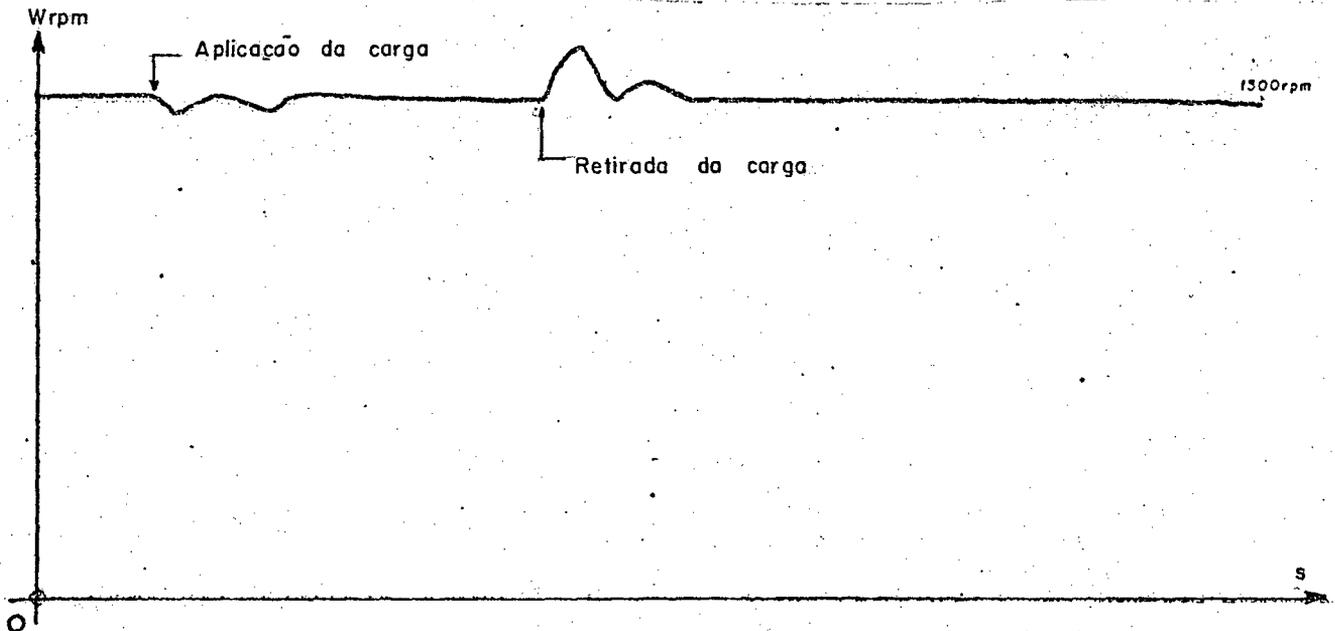


Fig. 40

## CAPÍTULO V

### CONCLUSÕES

Este sistema de controle da velocidade é praticamente todo digital. O uso do microcomputador dá uma grande flexibilidade ao sistema, o que compensa de certa maneira o custo para um sistema deste tipo.

Não foi um dos objetivos deste trabalho o desenvolvimento de um algoritmo de controle. Mas o sistema está preparado para receber algoritmos mais elaborados, e se necessário, com pequenas modificações, os períodos de amostragem dos dados e a ordem de prioridades no sistema de interrupções podem ser alterados.

Os trabalhos existentes utilizando regulação numérica em processos de controle são relativamente novos. Apesar de processos lentos apresentam muitas vantagens sobre sistemas analógicos, a começar pelo transdutor de velocidade onde não há nenhuma não linearidade e o sinal pode ser transmitido à distâncias bem maiores sem degradação (7) (10). O controle digital também não está sujeito a variações de temperaturas como num sistema analógico.

Com o programa de controle do processo gravado em uma memória PROM, caracterizando uma aplicação dedicada, o microcomputador para satisfazer as necessidades do sistema pode assumir uma configuração mínima, que é a constituída pelo microprocessador (8080), gerador de "clocks" (8224), controlador de barramentos (8228), duas interfaces programáveis (8255) e uma quantidade mínima de memórias RAM.

Um problema a ser enfrentado é a alta dose de ruídos existentes em oficinas ou laboratórios, onde funcionam es

tas máquinas a serem controladas. Além dos cuidados normalmente utilizados nestes casos, a implementação dos blocos com circuitos integrados CMOS por apresentarem alta imunidade contra ruídos, ajuda muito na minimização destes problemas.

Alguns melhoramentos podem ser introduzidos neste sistema; como o aumento do número de furos no disco acoplado ao eixo, o que possibilitaria menores bases de tempo no tacômetro digital e por consequência o processo se tornaria mais rápido. Na saída N do microcomputador, fazendo menor a variação de  $\theta$  para 1 LSB de N (resoluções menores), as exigências de tensão de armadura serão mais facilmente atendidas.

A utilização de um conversor reversível, constituído de duas pontes de tiristores colocados em anti-paralelo, em controle de velocidade resolve os problemas de frenagem do motor. Este será o objetivo para um trabalho que se segue a este. Nesta área de controle numérico, o uso de microprocessadores abre boas perspectivas, e projetos mais ambiciosos, como o controle de dois motores e controle de posição, são perfeitamente realizáveis.

A P Ê N D I C E

TABELA DOS ESTADOS EM UM CICLO DA MÁQUINA DO 8080

## ESTADO

$T_1$	Coloca endereço de memória ou dispositivo E/S na via de endereços, coloca informação de "Status" do ciclo de máquina na via de dados, liga o SYNC.
$T_2$	O processador testa as entradas READY e HOLD e instrução de HALT; desliga o sinal de SYNC.
$T_w$ (opcional)	O processador entra no estado de espera se "ready" é nível "0" ou se a instrução de "halt" foi executado.
$T_3$	A informação da via de dados é carregada no processador, quando DBIN fôr nível 1. Isto <u>o</u> corre em ciclos de máquina como "FETCH", "MEMORY READ", "STACK READ" e INTERRUPT". Se o processador está enviando dados à memória ou a dispositivo de E/S, o sinal $\overline{WR}$ do processador sinaliza com nível 0, o que ocorre nos ciclos de máquina como "MEMORY WRITE", "STACK WRITE" e "OUTPUT".
$T_4$ e $T_5$ (opcionais)	Nestes estados são executados as operações que faltam para completar a instrução.

SINAIS DOS PINOS DO MICROPROCESSADOR 8080

<u>SINAL</u>	<u>DESCRIÇÃO</u>
A <sub>0</sub> - A <sub>15</sub>	Via de endereços.
D <sub>0</sub> - D <sub>7</sub>	Via de dados (bi-direcional).
RESET	Entrada, limpa o contador de instruções reiniciando o processamento no endereço zero.
HOLD	Entrada para pedidos de acesso direto a memória, deixa o processador em estado de alta impedância.
HLDA	Saída, informa se está ou não em estado de HOLD.
INT	Entrada para pedidos de interrupções.
INTE	Saída, informa que o pedido de interrupção foi aceito.
$\phi_1$ e $\phi_2$	Entradas, sinais de relógios.
DBIN	Saída, sinaliza para a memória ou interface que o processador está pronto para receber dados.
$\overline{WR}$	Saída, sinaliza para memórias e interfaces o envio de dados pelo microprocessador.
SYNC	Saída, sinal gerado no início de um ciclo de máquina, sinaliza que a informação na via de dados é o "status" do microprocessador.
READY	Entrada que força o processador a entrar em estado de espera (T <sub>w</sub> ) aguardando memórias ou periféricos mais lentos.

WAIT

Saída que sinaliza que o processador está em estado de espera ( $T_w$ ).

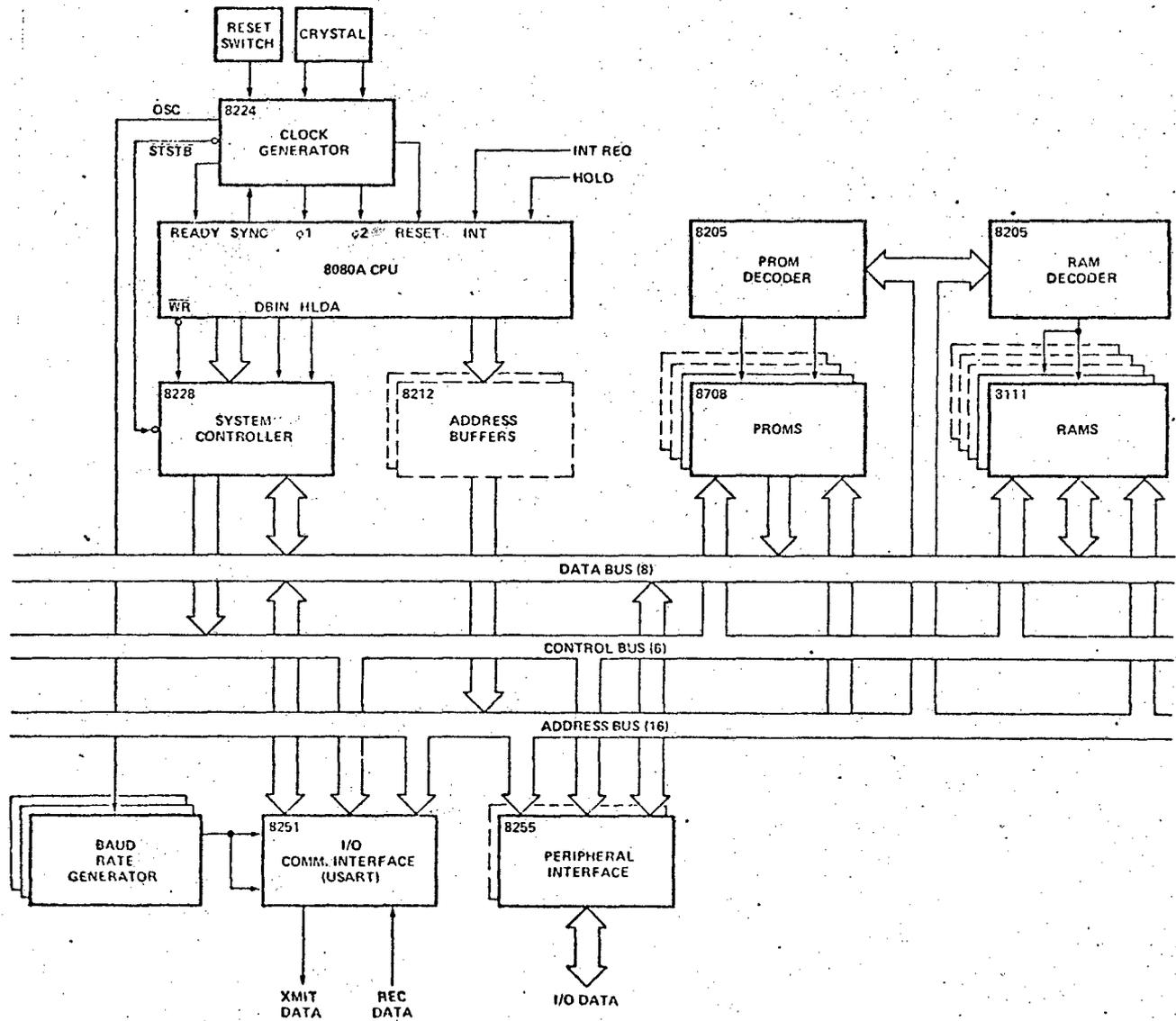


DIAGRAMA DO MICROCOMPUTADOR MCS-80 (SDK-80)

BLOCK DIAGRAM

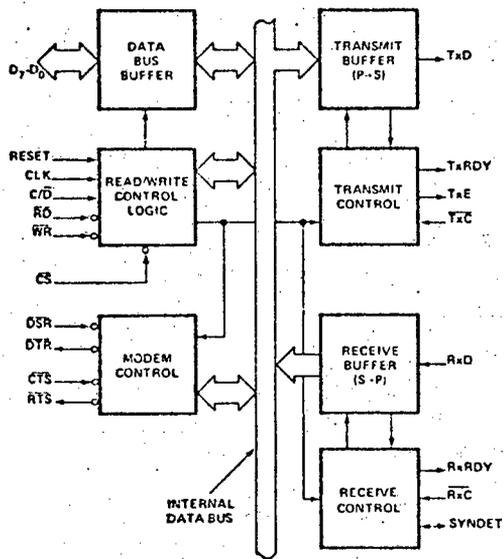


DIAGRAMA FUNCIONAL DO 8251  
(USART)

8255 BLOCK DIAGRAM

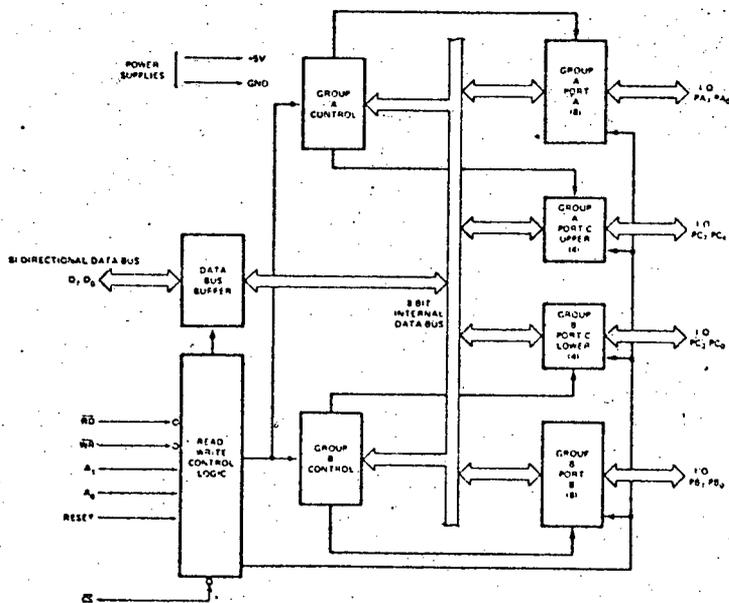


DIAGRAMA FUNCIONAL DO 8255  
(INTERFACE PROGRAMÁVEIS DE PERIFÉRICOS)

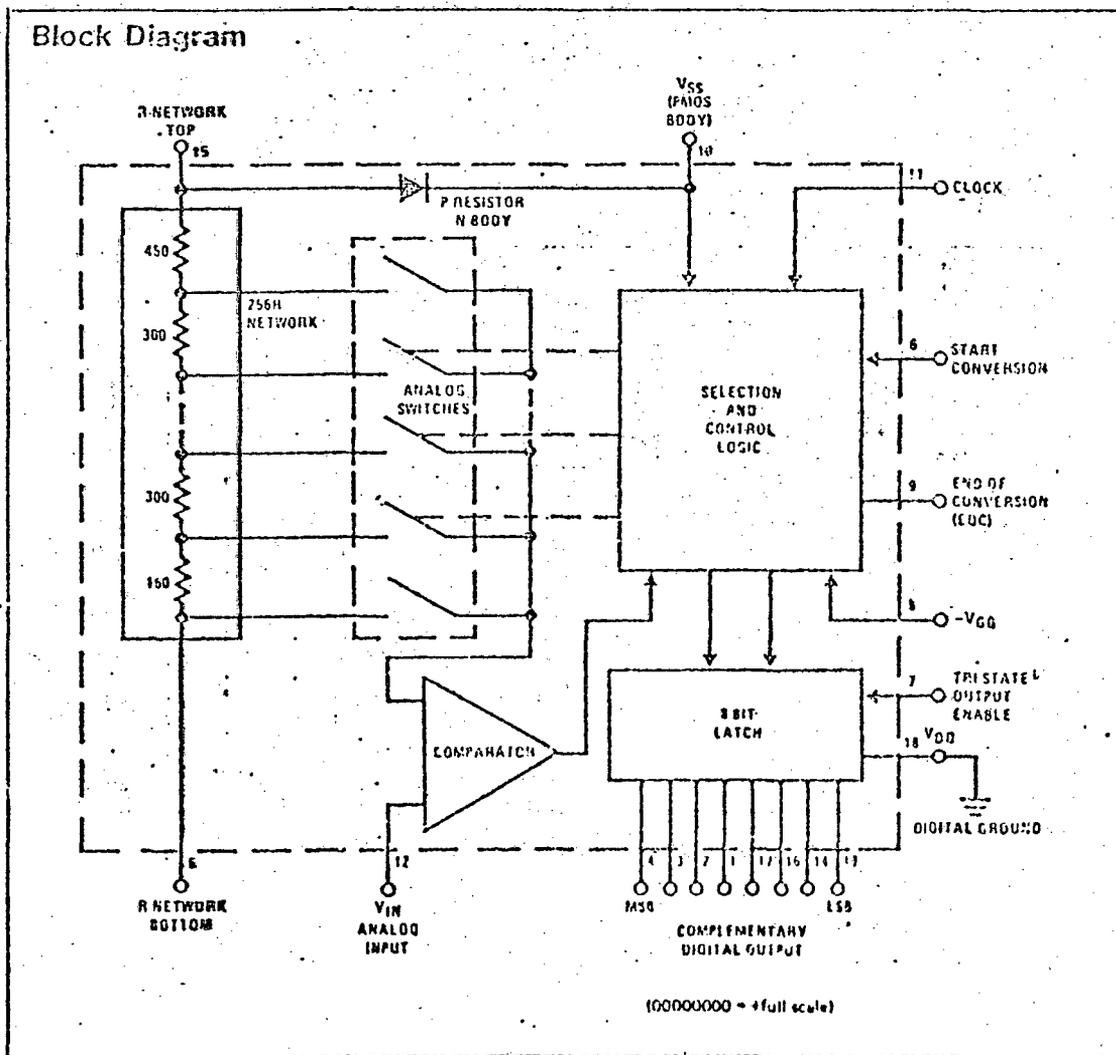
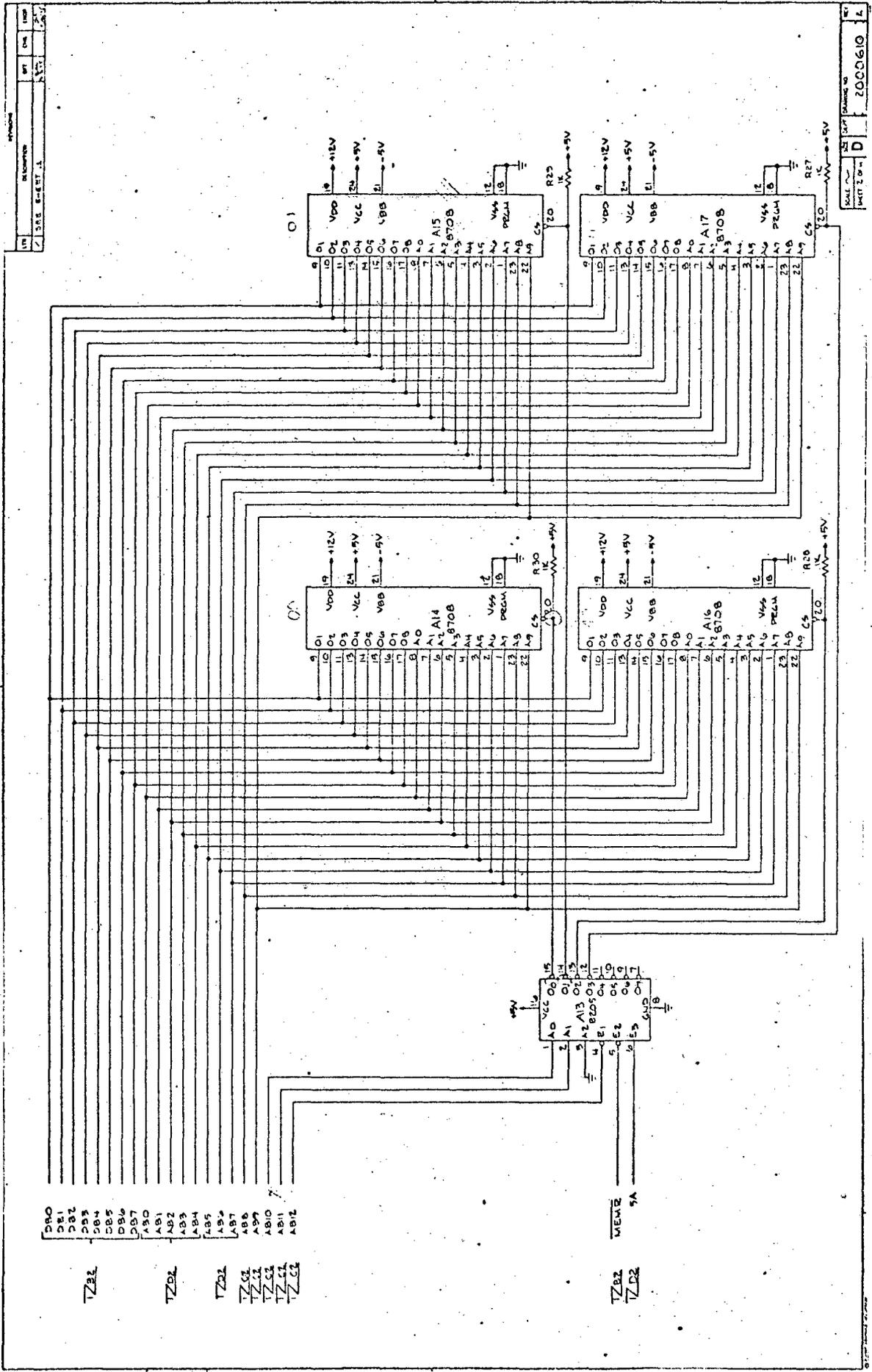


DIAGRAMA FUNCIONAL DO CONVERSOR ADC 800

Levantamento dos valores analógicos na saída ( $V_o$ ) do amplificador diferencial e valores convertidos na saída do conversor A/D, em resposta a valores ( $V_i$ ) na entrada do amplificador.

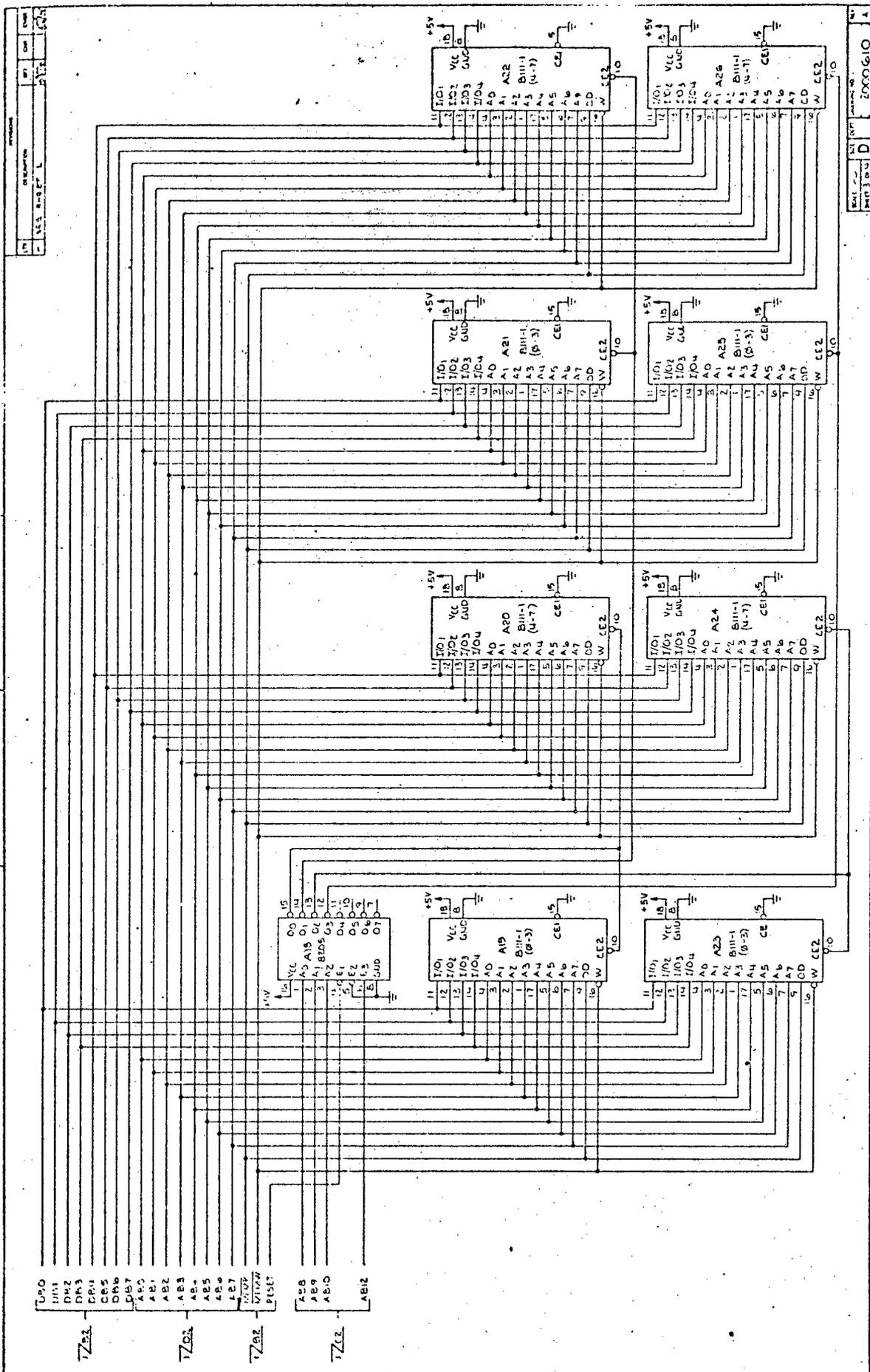
$V_i$	$V_o$	$V_o$ (Valores Digitais)
0,060 V	-5,00 V	FF
0,055 V	-4,65 V	F9
0,050 V	-4,45 V	F3
0,045 V	-4,26 V	ED
0,040 V	-3,92 V	E5
0,035 V	-3,60 V	DD
0,030 V	-3,05 V	CF
0,025 V	-2,65 V	C5
0,020 V	-2,15 V	B9
0,015 V	-1,75 V	AE
0,010 V	-1,11 V	9D
0,005 V	-0,63 V	91
0,00 V	-0,01 V	7F
-0,005 V	0,58 V	70
-0,010 V	1,03 V	64
-0,015 V	1,52 V	53
-0,020 V	2,10 V	47
-0,025 V	2,53 V	3C
-0,030 V	3,07 V	2C
-0,035 V	3,54 V	20
-0,040 V	3,9 V	15
-0,045 V	4,35 V	0D
-0,049 V	4,41 V	08
-0,055 V	4,67 V	02
-0,060 V	5,0 V	00

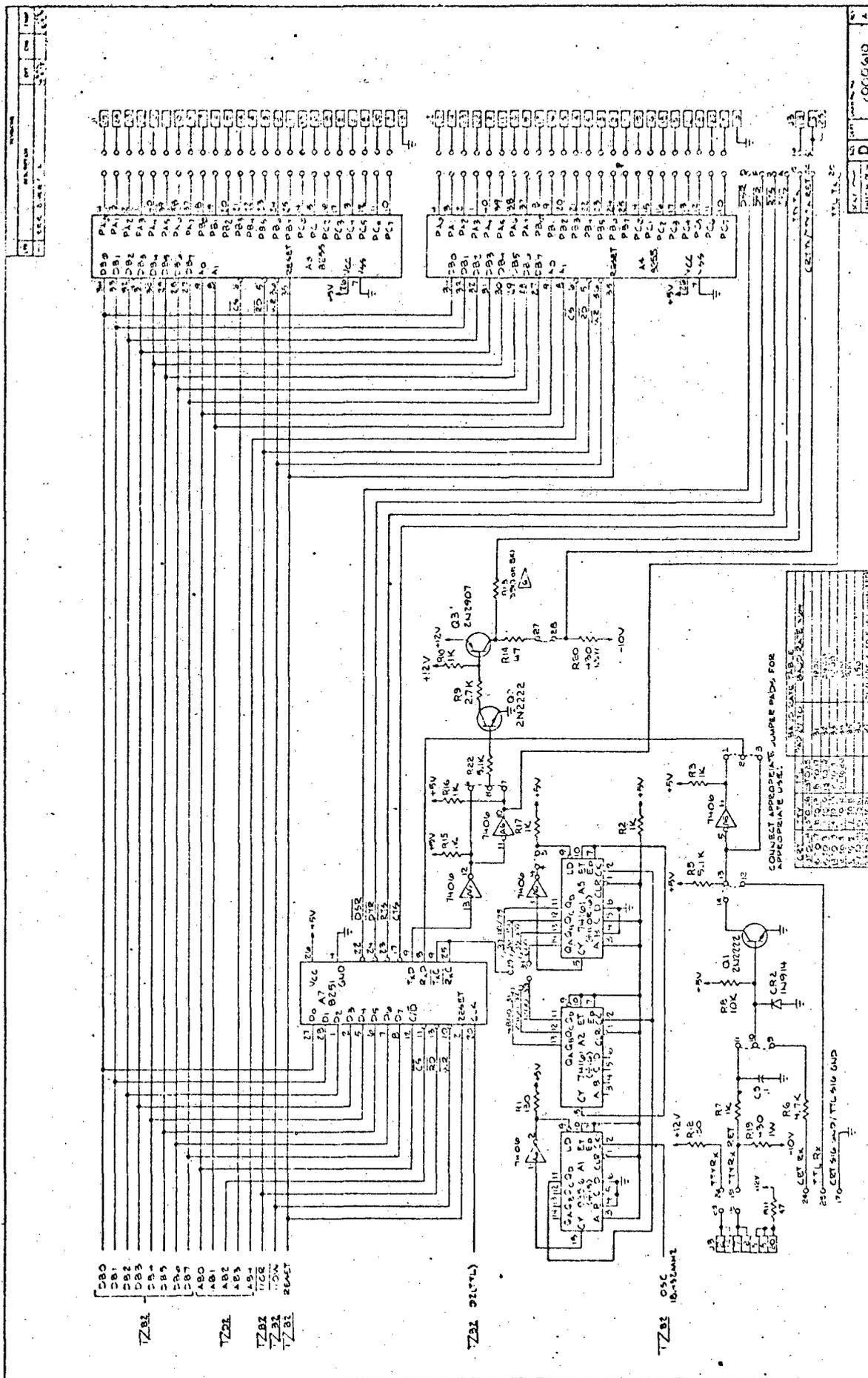
ESQUEMAS DO SISTEMA MCS-80



REV: 1.0  
 DATE: 10/10/00  
 DRAWN BY: [Signature]  
 CHECKED BY: [Signature]

SCALE: 1:1  
 SHEET: 1 OF 1  
 PROJECT: 20000610





REV. 1.0  
 DATE: 10/15/81  
 DRAWN: J. J. J.  
 CHECKED: J. J. J.  
 APPROVED: J. J. J.

1000010

D

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010

1000010



REFERÊNCIAS BIBLIOGRÁFICAS

1. LIN, A.K.; KOEPEL, W.W. - "A Microprocessor Speed Control System", IEEE, Trans. Ind. Electron. Control Instrum. Vol. IECI 24, nº 3 - agosto 1977.
2. LANGDON JR., G.G.; FREGNI, E. - "Projeto de Computadores Digitais". Editora Edgard Blücher LTDA. São Paulo - Brasil - 1974.
3. WYLAND, D.C. - "How to design your own microprocessor". NMI-Monolithic Memories, Inc. Sunnyvale, California, U.S.A.
4. ZUFFO, J.A. - "Sistemas Eletrônicos Digitais" - Volume I Editora Universidade de São Paulo - Editora Edgard Blücher LTDA. São Paulo - Brasil, 1976.
5. TEXAS INSTRUMENTS, ELECTRONICS SÉRIES - "Electronic Power Control and Digital Techniques" - Editora McGraw-Hill. New York - U.S.A., 1976.
6. NATIONAL SEMICONDUCTOR - "Memory Applications Handbook", 1978.
7. MALONEY, T. J.; ALVARADO, F.L. - "A Digital Method for DC Motor Speed Control", IEEE Trans. Ind. Electron. Constr. Instrum., Vol. IECI - 23, pg 44-46 Fev. 1976.
8. INTEL CORPORATION - "Microprocessor devices System SAB 8080, 1975.
9. INTEL CORPORATION - "MCS-80 System design Kit User's Guide", 1975.
10. DEMERLE, M.; FROMONT, J. - "Régulation Numérique directe de vitesse, première partie; Capteur numérique de vitesse, Revue Jeumont-Schneider, Nº 26. Agosto 1978.
11. SIFFER;EN, T.P.; VARTARIAN, V. - "Digital Electronics with Engineering Application" - Electrical Engineering Series, Prentice-Hall, Inc. Englewood Cliffs, N. J., U.S.A., 1970.

12. WAIT, J.V.; HUELSMAN, L.P.; KORN, G.A. - "Introduction to Operational Amplifier Theory and Application". Editora McGraw Hill - Kogakusha, Tokyo, 1975.
13. JACKSON, R.D.; WEATHERBY, R.D. - "Direct Digital Control of Thyristor Convertors".  
IFAC Symposium on Control in Power Electronics and Electrical Drives, Duesseldorf, October, 7-9, 1974 - pg 431 - 441.
14. KLINGMAN, E. E. - "Microprocessor Systems Design", Editora Prentice-Hall, New Jersey, 1977.
15. TEXAS INSTRUMENTS INCORPORATED - "Design with TTL Integrated Circuits". Editora McGraw-Hill Kogakusha, Tokyo, Japão, 1971.
16. WAKERLY, J.F. - "Logic Design Projects Using Standard Circuits", Editora John Wiley e Sons, New York, 1976.
17. NATIONAL CORPORATION - "CMOS databook", 1978.
18. TEXAS INSTRUMENTS - "The TTL databook", 1976
19. FAIRCHILD SEMICONDUCTOR - "MOS/CCD databook", 1977.