

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
DEZEMBRO DE 1980

PROJETO E IMPLEMENTAÇÃO DE  
UMA REDE  
HIERÁRQUICA DE  
MICROPROCESSADORES

por : Takao Suguiy  
orientador : Prof. Dr. Márcio Luiz de Andrade Netto

40/80

Tese apresentada à Faculdade de Engenharia - FEC -  
UNICAMP como parte dos requisitos exigidos para  
obtenção do título de MESTRE EM CIÊNCIAS.

UNICAMP  
BIBLIOTECA CENTRAL

Este trabalho contou com o apoio financeiro do CONSELHO NACIONAL  
DE DESENVOLVIMENTO CIENTÍFICO E TECNOLÓGICO - CNPq.

## AGRADECIMENTOS

Desejo expressar minha gratidão àqueles que generosamente me ajudaram a colorir o mosaico deste trabalho, com os fragmentos de suas lembranças, conhecimento e técnicas. Em particular sou grato ao Prof. Dr. Márcio Luiz de Andrade Netto, meu orientador, pela sua assistência, estímulo e inúmeras sugestões em todas as fases deste trabalho, e a Sra. Míyano Nakatani pelo apoio.

## RESUMO

Descreve-se neste trabalho, o projeto e a implementação de uma rede hierárquica de microprocessadores desenvolvida no laboratório de microcomputadores do Departamento de Engenharia Elétrica FEC - UNICAMP. Tal projeto tem por finalidade a utilização didática e a realização de pesquisas na área de controle de processos industriais por computadores, tanto em seus aspectos "hardware" como no desenvolvimento da programação necessária para o efetivo aproveitamento dos recursos à disposição.

## INDICE

CAPÍTULO 1 - INTRODUÇÃO .....	1.1
1.1 - Tecnologia .....	1.1
1.2 - Características p/avaliação de Tecnologia ...	1.1
1.3 - Tecnologia MOS .....	1.2
1.4 - Tecnologia Bipolares.....	1.3
1.5 - Microcomputadores.....	1.4
CAPÍTULO 2 - REDE DE COMPUTADORES .....	2.1
2.1 - Definições e Conceitos.....	2.1
2.2 - Rede de Computadores .....	2.2
2.3 - Arquitetura .....	2.5
2.4 - Estrutura de Interconexão de Computadores ...	2.7
2.5 - Barramento .....	2.15
CAPÍTULO 3 - INTERFACE .....	3.1
3.1 - Introdução .....	3.1
3.2 - Funcionamento da PIC - 8259 .....	3.3
3.3 - Funcionamento da Interface .....	3.4
3.4 - Circuitos Implementados .....	3.5
3.5 - Barramentos Utilizados .....	3.8
CAPÍTULO 4 - TESTES E CONSIDERAÇÕES FINAIS .....	4.1
4.1 - Testes de Funcionamento .....	4.1
4.2 - Diagramas de Blocos .....	4.1
4.3 - Considerações Finais .....	4.12
ANEXO A .....	A.1
A.1 - Intel 8086 .....	A.1
A.2 - Zilog Z-80 .....	A.3
A.3 - PIC - 8259 .....	A.4
A.4 - Z-80 PIO .....	A.6



ANEXO B .....	B.1
B.1 - Barramento IEEE-488 .....	B.1
B.2 - Barramento 6800 .....	B.2
B.3 - Barramento Multibus .....	B.2
B.4 - Barramento S-100 .....	B.7
B.5 - Barramento Série RS-232 e RS-449 .....	B.11
ANEXO C .....	C.1
C.1 - Análise das Estruturas de Interconexão .....	C.1
BIBLIOGRAFIA .....	D.1

## CAPÍTULO 1

## 1. INTRODUÇÃO

O advento da tecnologia de integração em alta escala (LSI, VLSI) possibilitou reduzir significativamente o tamanho físico e o preço das máquinas computacionais. Em virtude desta redução tornaram-se viáveis diversas aplicações para estes dispositivos, bem como surgiram novas aplicações. Microcomputador foi a denominação recebida por estes dispositivos devido ao seu tamanho físico reduzido.

### 1.1 - Tecnologia

A influência da tecnologia de circuitos integrados na arquitetura dos computadores é bastante significativa. Levando-se em consideração a densidade de integração costuma-se usar a seguinte classificação:-

1.1.1 - SSI - INTEGRAÇÃO EM BAIXA ESCALA - "SMALL SCALE INTEGRATION" - usado para designar os circuitos integrados que possuem até 12 "gates" por "chip", esta tecnologia foi difundida entre 1.964 e meados de 1.970.

1.1.2 - MSI - INTEGRAÇÃO EM MÉDIA ESCALA - "MEDIUM SCALE INTEGRATION" - usada para designar os circuitos integrados que possuem de 12 a 100 "gates" por "chip", esta tecnologia é um aprimoramento da SSI e surgiu em 1.970.

1.1.3 - LSI - INTEGRAÇÃO EM LARGA ESCALA - "LARGE SCALE INTEGRATION" - usada para designar os circuitos integrados que possuem de 100 a 10.000 "gates" por "chip". Com esta tecnologia a fabricação de um processador central em um único "chip" com alta confiabilidade, baixa dissipação de potência, e uma redução substancial do tamanho físico, tornou-se realidade; a tecnologia LSI tornou-se viável comercialmente em 1.975.

1.1.4 - VLSI - INTEGRAÇÃO EM ESCALA MUITO ALTA - "VERY LARGE SCALE INTEGRATION" - usado para designar os integrados com mais de 10.000 "gates" por "chip".

### 1.2 - Características para Avaliação de Tecnologias

1.2.1 - ECONOMIA - medida em termos de custo por bit, custo por "gates", etc.

1.2.2 - DISSIPACÃO - deve ser baixa por "gate", visto que o circuito deve ser reduzido ao menor tamanho possível.

1.2.3 - VELOCIDADE - medida em termos de tempo de acesso a memória, tempo de processamento de funções, atraso de propagação, etc.

1.2.4 - CONCENTRAÇÃO - medida em termos de "gates", transistores, etc., por unidade de superfície ou área.

1.2.5 - CONFIABILIDADE - definida em termos de Tempo Médio entre Falhas - "*Mean Time between Failure*" - MTBF - dado o alto grau de concentração de "gates" pode ser um elemento crítico em VLSI.

1.2.6 - COMPATIBILIDADE - por exemplo, tensões altas de alimentação em MOS resultam em incompatibilidade com TTL.

### 1.3 - Tecnologia MOS

Foi a tecnologia utilizada nos primeiros microprocessadores monolíticos.

1.3.1 - PMOS - MOS CANAL P - entre os microprocessadores fabricados com esta tecnologia estão o INTEL 4004, INTEL 8008, PPS 4, PPS8, etc.

. vantagens:-

- .. simplicidade
- .. preço baixo
- .. produção com bom rendimento

. desvantagens:-

- .. tensão de limiar alta - "*threshold*"
- .. baixa velocidade
- .. tensões alta de alimentação

Hoje em dia, graças ao progresso desta tecnologia é possível conseguir uma baixa tensão de limiar, compatível com a tecnologia TTL.

1.3.2 - NMOS - MOS CANAL N - a vantagem do NMOS em relação do PMOS é o seu melhor desempenho, devido à maior mobilidade dos elétrons em relação às lacunas no silício.

Esta tecnologia é usada nos microprocessadores da segunda

geração como o INTEL 8080, INTEL 8085, F8, M6800, etc.

Como desvantagem podemos citar a alta sensibilidade do cristal à contaminação o que implica na necessidade de processos extremamente limpos, e com "yield" mais baixo.

#### 1.3.3 - CMOS - "COMPLEMENTARY SYMMETRY MOS"

. vantagens:-

- .. alto desempenho
- .. dissipação de energia extremamente baixa
- .. tensão única de alimentação, podendo ser de 3 a 18volts.
- .. alta imunidade a ruídos

. desvantagens:-

- .. menor integração
- .. produção difícil

1.3.4 - SOS - "SILICON ON SAPPHIRE OR SPINET" - tecnologia pouco divulgada.

1.3.5 - DMOS - "DOUBLE DIFFUSED MOS" - alto desempenho, ainda em estudo.

1.3.6 - VMOS - "V-GROOVE MOS" - permite uma integração muito alta.

1.3.7 - HMOS - "HIGH DENSITY MOS" - tecnologia empregada na fabricação do microprocessador INTEL 8086.

### 1.4 - Tecnologias Bipolares.

É a tecnologia tradicional no mercado de circuitos integrados.

1.4.1 - TTL - "TRANSISTOR TRANSISTOR LOGIC" - é a tecnologia mais divulgada nos últimos anos; assim sendo, frequentemente, impõe condições de compatibilidade às outras tecnologias.

Características:-

- . desempenho alto para a maioria das aplicações
- . custo relativamente baixo em virtude da alta produção .
- . grande familiaridade dos usuários.
- . grande número de produtores com experiência no campo.

A tecnologia TTL com relação à integração em larga escala apresenta desvantagens:-

- . baixo nível de integração.
- . dissipação de energia muito alta.
- . processo de fabricação relativamente complexa

A tecnologia TTL já demonstra grande melhoria de desempenho e densidade na integração em grande escala, nos microprocessadores o tipo "*bit-slice*", como o INTEL 3000, AMS 2900, etc.

1.4.2 - ECL - "*EMITTER COUPLED LOGIC*" - é uma tecnologia que ainda não se tornou popular no mercado, e os fabricantes dispõem de pouca experiência; mas apresenta a vantagem de ser a mais rápida de todas as tecnologias. O seu atraso de propagação típico é da ordem de 1 nanosegundo.

1.4.3 -  $I^2L$  - "*INTEGRATED INJECTION LOGIC*" - trata-se de uma das tecnologias bipolares mais recentes (1975), desenvolvida na Europa pela PHILIPS e IBM, hoje utilizada por várias companhias como por exemplo a TEXAS INSTRUMENTS.

. vantagens:-

- .. alta concentração, própria para LSI.
- .. baixa dissipação de energia semelhante à CMOS.
- .. desempenho relativamente alto.
- .. grande facilidade na combinação de circuitos analógicos e digitais.

. desvantagens:-

- .. pouca experiência dos fabricantes.
- .. a pouca popularidade no mercado.

## 1.5 - Microcomputadores

É uma ferramenta que tem a capacidade de manipular números binários (informações) através de uma sequência pré-determinada de passos (programa). Cada passo da sequência pré-determinada é denominada instrução.

Os microcomputadores são formados basicamente por:-

- . uma unidade central de processamento (CPU)
- . memória
- . portas de entrada e saída de dados
- . lógica auxiliar

De uma maneira mais simples podemos imaginar um microcomputador composto de 4 blocos básicos:-

- . dispositivos de entrada
- . memória
- . microrprocessador
- . dispositivos de saída.

Estes blocos são interconectados por um grupo de fios chamado barramento - "bus".

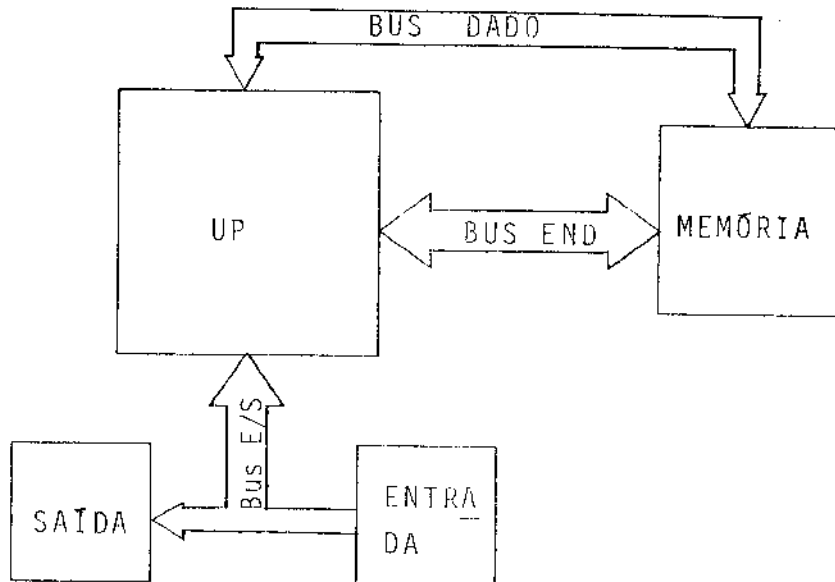


Figura 1.1

É comum microcomputadores em que a memória e os dispositivos de entrada/saída - E/S - dividam o mesmo barramento. Neste caso o microprocessador trata um dispositivo de E/S como um elemento de memória.

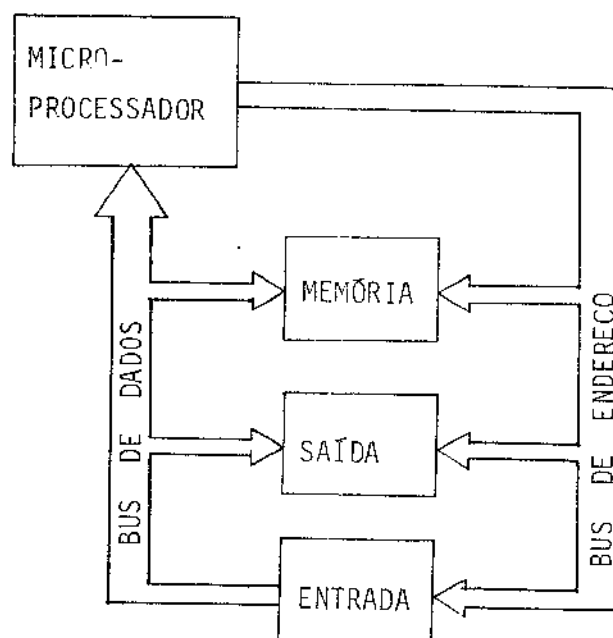


Figura 1.2

## CAPÍTULO 2



## 2. REDE DE COMPUTADORES

### 2.1 - Definições e Conceitos

2.1.1 - REDE DE COMPUTADORES - é o conjunto de um ou mais computadores, elos de comunicação e terminais interconectados para fornecer serviços a um conjunto de usuários.

2.1.2 - NÓ - é um sistema de computação que está unido a uma rede de computadores.

2.1.3 - REDE HOMOGÊNEA - é uma rede constituída de processadores idênticos, tanto no sentido físico como lógico.

2.1.4 - REDE HETEROGÊNEA - é uma rede que não é homogênea, mas pode conter subredes homogêneas.

2.1.5 - NÓ DE CHAVEAMENTO - é o nó no qual a função primária é o chaveamento de dados na rede.

2.1.6 - MODULARIDADE - é a habilidade de se realizar mudanças na estrutura visando um incremento da capacidade do sistema, com mínima implicação no custo - "*cost-modularity*" - e com flexibilidade quanto à localização - "*place-modularity*" - dos novos dispositivos.

2.1.7 - FALHA NOS DISPOSITIVOS - é a projeção estudada acerca dos efeitos - "*failure effect*" - causado pelo mau funcionamento de um elemento sobre toda a estrutura da rede. Tem-se também a discussão da configuração da rede perante uma falha - "*failure reconfiguration*" - na qual realizam-se modificações na estratégia de encaminhamento das mensagens - "*software*" - ou alterações das conexões, com a eliminação de processadores que apresentam problemas - "*hardware*".

2.1.8 - COMPLEXIDADE LÓGICA - diz respeito à lógica de interconexão usada, necessária a cada interface dos elementos da rede.

2.1.9 - GARGALOS - "*bottlenecks*" - onde são considerados os elementos que podem vir a degradar o desempenho do sistema como um todo.

2.1.10 - CAMINHO - "*path*" - é o meio pelo qual a mensagem é transferida entre os elementos do sistema.

2.1.11 - ELEMENTO COMUTADOR - "*switching element*" - é a entidade que pode ser vista como um intermediário inteligente entre o remetente e o destinatário da mensagem.

2.1.12 - CAMINHO COMPARTILHADO - é o caminho acessível por mais de dois pontos.

2.1.13 - TRANSMISSÃO DIRETA - quando a mensagem é enviada diretamente da fonte para o destinatário.

2.1.14 - TRANSMISSÃO INDIRETA - quando há a necessidade de um elemento comutador para enviar a mensagem.

2.1.15 - TRANSFERÊNCIA CENTRALIZADA - quando uma entidade simples comuta todas as mensagens.

2.1.16 - TRANSFERÊNCIA DESCENTRALIZADA - quando é utilizada mais de uma entidade comutadora de mensagem.

2.1.17 - TAMANHO DA REDE - geralmente refere-se ao número de nós, de computadores ou terminais conectados.

## 2.2 - Rede de Computadores

O termo "REDE DE COMPUTADORES" é correntemente usado para descrever uma ampla classe de sistemas de processamento de dados; no sentido mais amplo, pode ser escrito como um conjunto de computadores e terminais interconectados por meio de um sistema de comunicação. Entretanto, o uso de um simples termo para descrever este grau de variedade de sistemas esconde uma importante diferença que existe entre as redes de computadores. Esta distinção é baseada na maneira pela qual o usuário vê uma dada rede, ou seja, o grau de transparência que uma rede apresenta para o usuário. Por este critério podemos descrever duas classes diferentes de redes. Estas classes se distinguem na manipulação dos recursos dos computadores. No primeiro caso a manipulação dos recursos recai no usuário, ao passo que no segundo, o usuário pode se apoiar na ajuda do sistema operacional da rede para o tratamento dos recursos necessários.

As duas classes de redes são:-

- . rede de comunicação de computadores
- . rede de computador

2.2.1 - REDE DE COMUNICAÇÃO DE COMPUTADORES - a característica básica de uma rede de comunicação de computadores é que o usuário vê a rede como uma coleção de muitos sistemas de computadores, com serviços e capacidades variadas. Dos muitos sistemas de computadores disponíveis, o usuário pode explicitamente escolher o sistema no qual deseja executar o seu trabalho - "JOB" -; para isto o usuário deve estabelecer uma conexão através da rede para o sistema escolhido.

Recursos variados podem estar disponíveis para o usuário deste tipo de rede. Estes recursos podem ser banco de dados, subrotinas (programas de apoio ao usuário), hardware especializado.

Para usufruir destes recursos o usuário da rede deve especificar o sistema no qual os recursos residem, bem como familiarizar-se com os comandos necessários para sua manipulação.

A figura 2.1 visualiza a maneira pela qual o usuário vê uma rede de comunicação de computador.

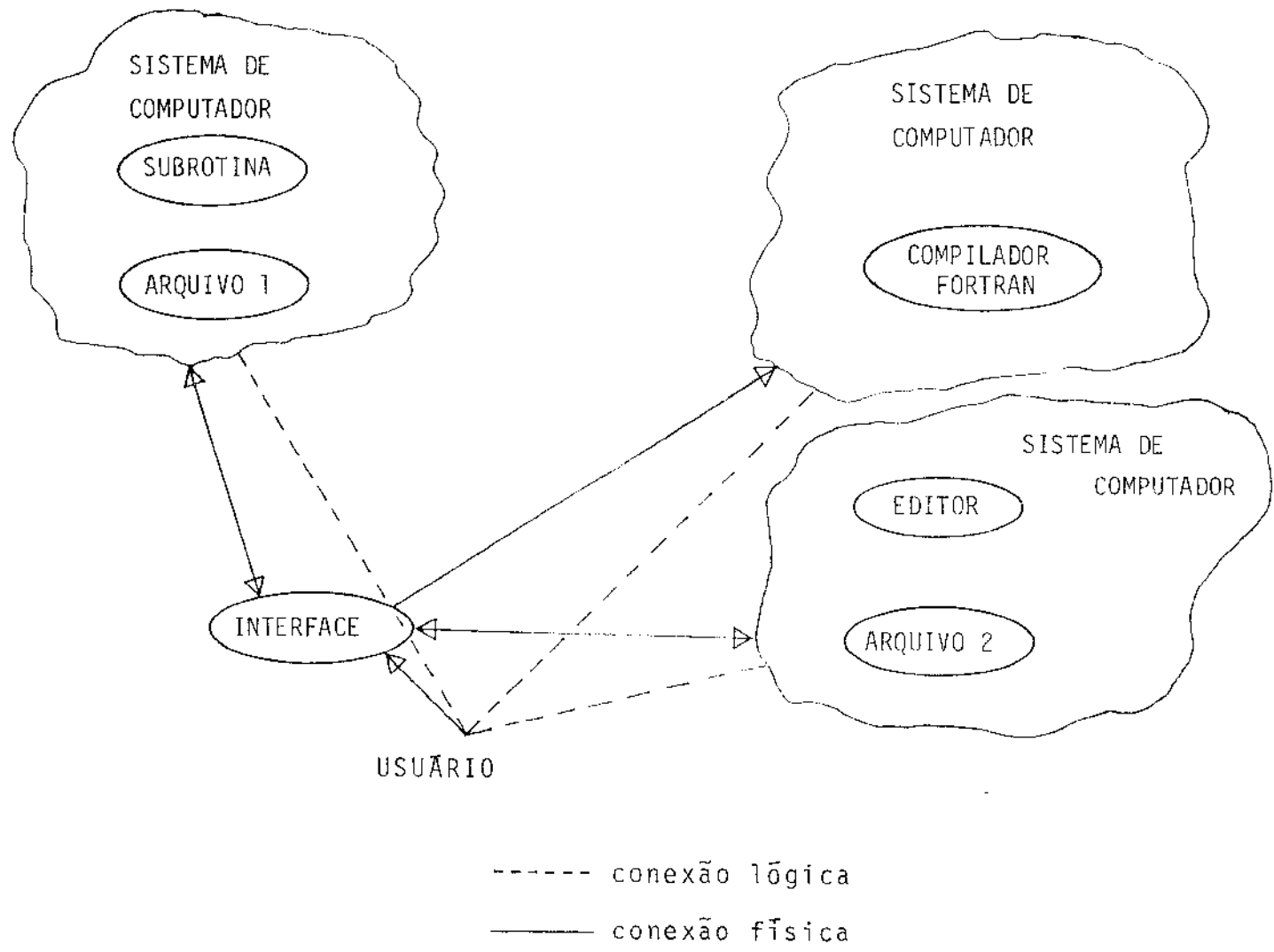


Figura 2.1 - REDE DE COMUNICAÇÃO DE COMPUTADORES

2.2.2 - REDE DE COMPUTADORES - também oferece uma variedade de recursos computacionais ao usuário. O usuário vê a rede de computadores como um sistema processador de grande porte.

A característica essencial de uma rede de computadores é que existe um sistema operacional da rede que toma as decisões para o usuário; como consequência não há necessidade de que o usuário saiba em qual sistema se localiza um determinado recurso.

A figura 2.2 mostra como o usuário vê uma rede de computadores.

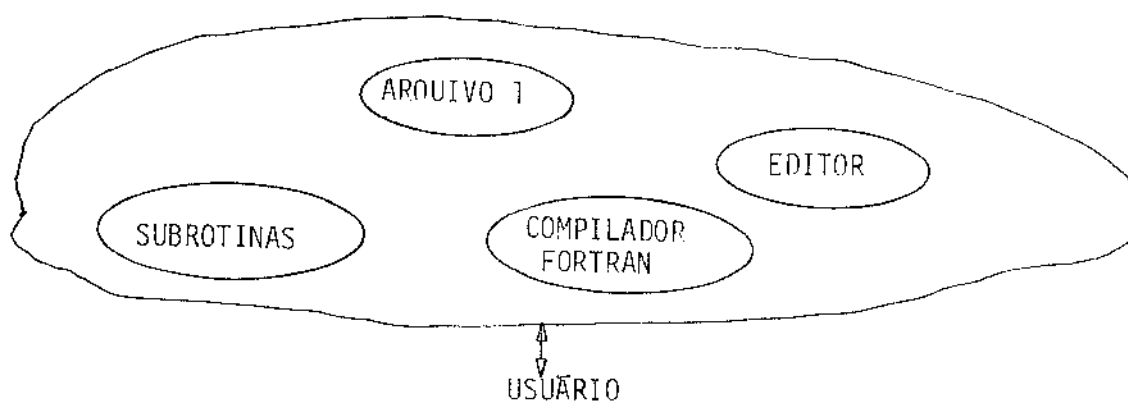


Figura 2.2 - REDE COMPUTADORES

Como exemplos podemos citar :

- . TYMNET, ARPANET(7) - redes de comunicação de computadores
- . RSEXEC (resource sharing executive), DCS (distributed computer system)(7) - redes de computadores.

## 2.3 - Arquitetura

Os atributos básicos de uma rede que distinguem sua arquitetura incluem sua topologia, composição, estratégia de utilização e mecanismo de controle.

### 2.3.1 - TOPOLOGIA

- . centralizado ou estrela;
- . distribuído, parcialmente ou totalmente conectado;
- . anel - é uma variante do sistema distribuído;
- . várias combinações possíveis.

2.3.1.1. - Centralizado - o nó central da rede despacha todas as mensagens.

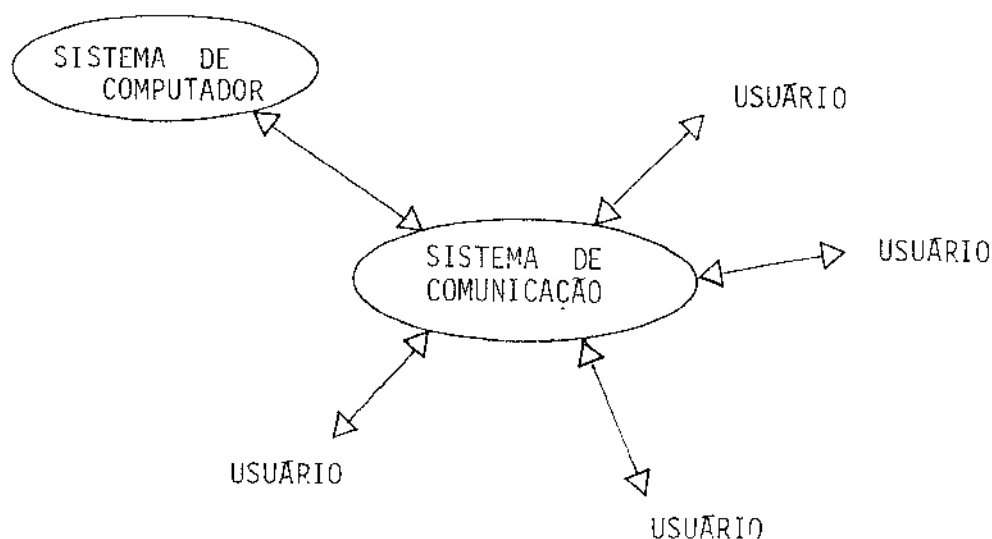
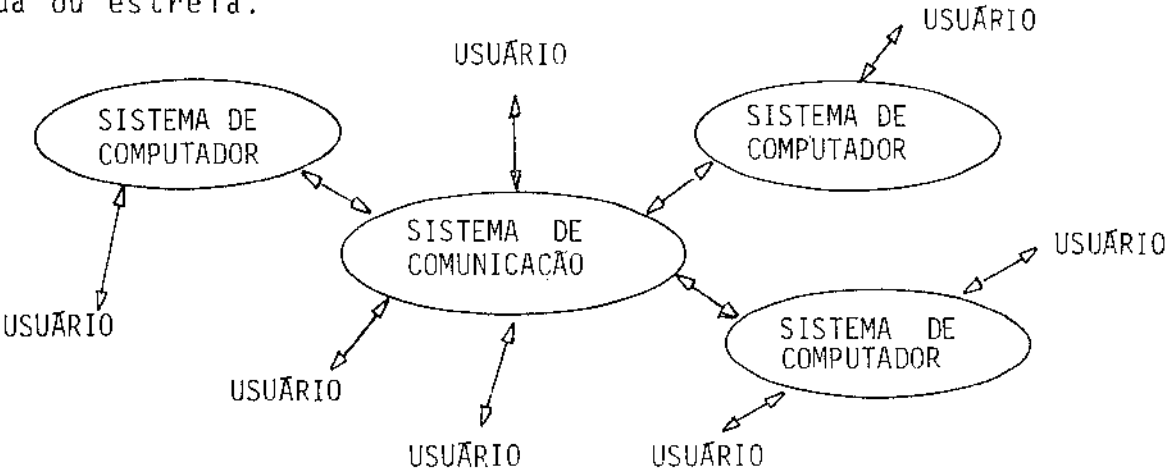


Figura 2.3 - CENTRALIZADO

2.3.1.2 - distribuido

. conexão completa - cada nó é conectado com todos outros nós.

. árvore - é uma extensão da rede centralizada ou estrela.



Distribuído

Figura 2.4

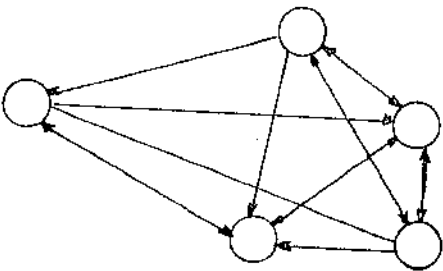


Figura 2.5

Conexão completa

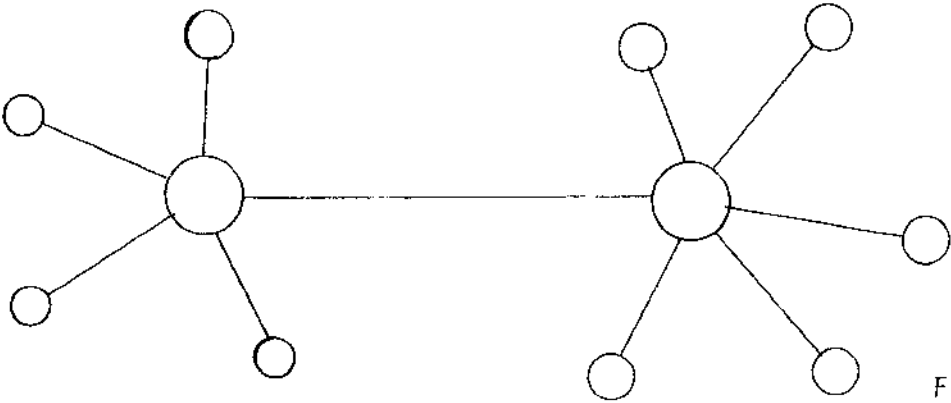


Figura 2.6

Árvore

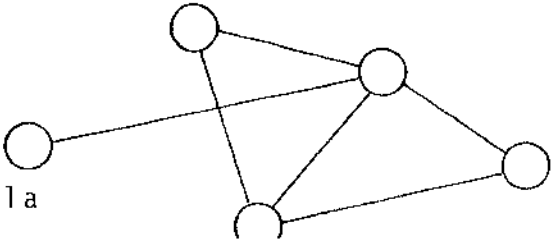
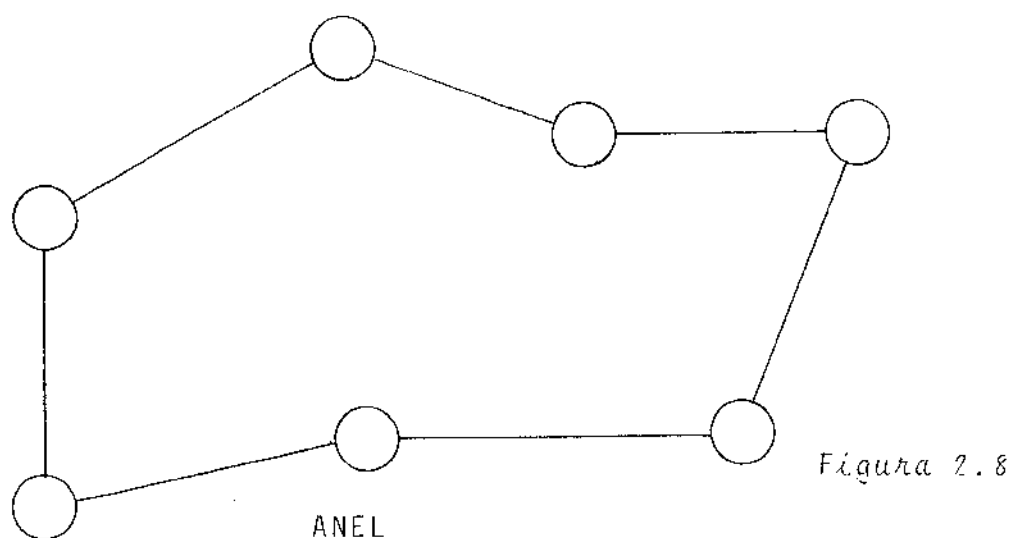


Figura 2.7

Conexão múltipla

### 2.3.1.3 - conexão múltipla

. anel - é um caso especial da rede distribuída, onde cada nó é conectado com exatamente dois nós vizinhos.



. combinações - podem ser homogêneas ou heterogêneas.

## 2.4 - Estruturas de Interconexão de Computadores.

A seguir descrevem-se e analisam-se as diferentes maneiras de se realizar as ligações entre processadores, baseando-se na manipulação de mensagens entre os dispositivos e a topologia de interconexão, conforme a árvore das estruturas de conexão de processadores, figura 2.9.

2.4.1 - DDL - ANEL DEDICADO - "*DIRECT DEDICATED LOOP*" - esta estrutura de conexão é também conhecido por estrutura anel, é caracterizado pelo fato de cada nó (elemento processador) ser conectado a somente dois nós vizinhos.

A mensagem na rede poderia, em princípio, fluir em ambas as direções; na prática a complexidade do tráfego bidirecional tem dirigido as redes para o tráfego unidirecional.

Neste tipo de estrutura, com comunicação unidirecional, um

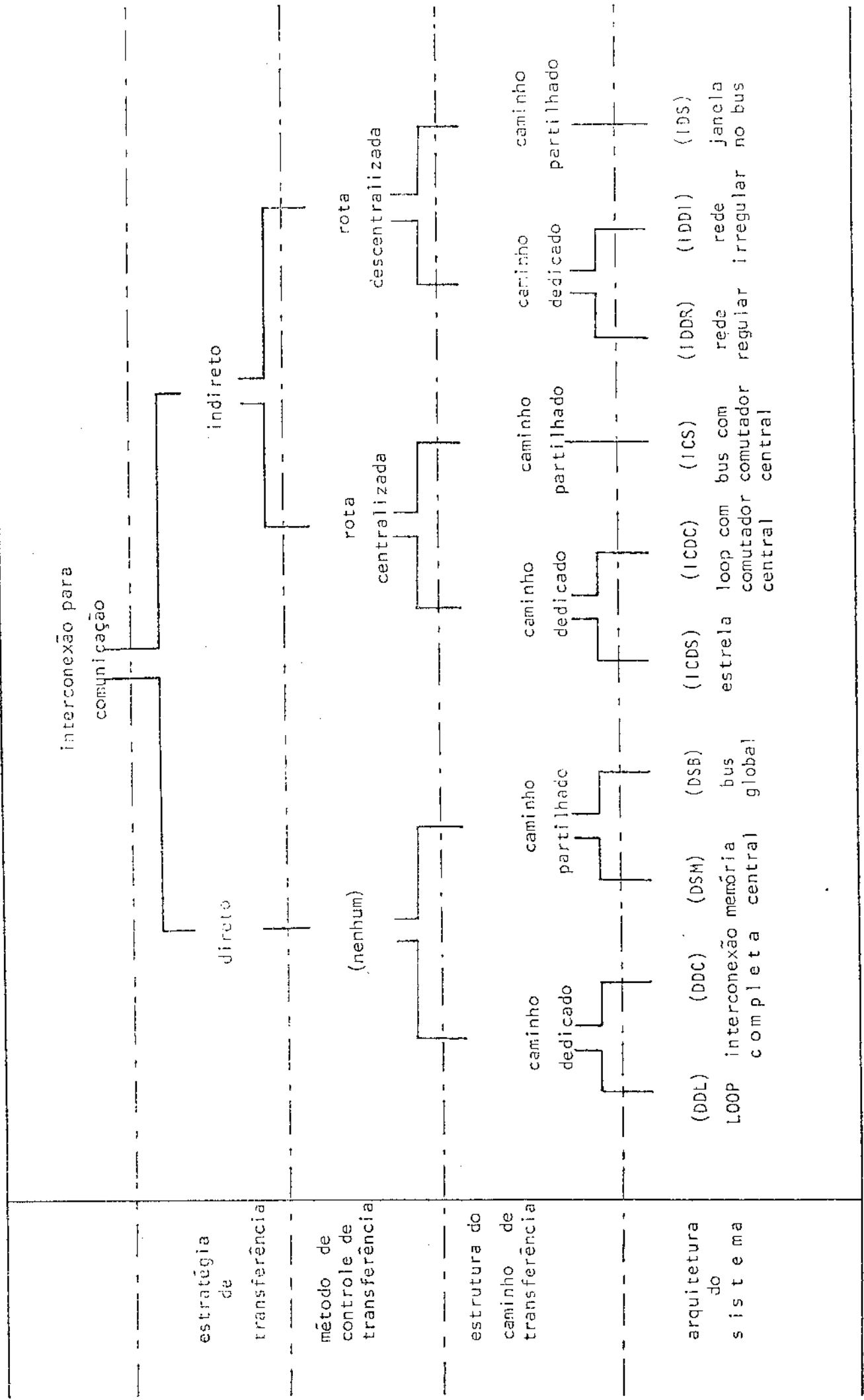


Figura 2.9 - ÁRVORE DAS ESTRUTURAS DE CONEXÕES DE PROCESSADORES



nó vizinho pode ser considerado como fonte e o outro como destino, ou seja, um nó recebe mensagem somente da fonte vizinha e envia mensagem somente para o vizinho destino. Mensagens circulam ao redor do anel da fonte para o destino com os nós intermediários atuando como re-transmissores ou retentores - "buffer".

Este tipo de conexão apresenta características muito boas quanto à modularidade, por permitir a inserção de um elemento processador em qualquer lugar da rede. É pobre quanto à falha nos dispositivos e reconfiguração da rede, devido as implementações serem geralmente com caminhos unidirecionais, podendo melhorar pela adoção de redundância nas ligações.

Sua complexidade lógica nas ligações é baixa e a capacidade de transmissão - "bandwidth" - do caminho de comunicação é um gargalo em potencial.

Em geral tais sistemas são projetados para aplicações onde os requisitos de confiabilidade e de desempenho não são demasiadamente grandes.

Como exemplo de um sistema DDL podemos citar o DCS (Distributed Computer System) da University of California, Irvine (7).

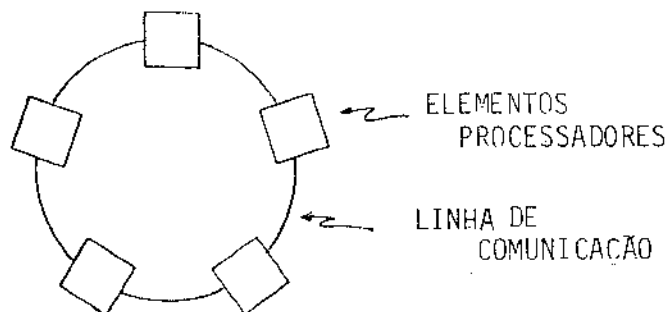


Figura 2.10 - DDL.

2.4.2 - DDC - DEDICADO COM CONEXÕES COMPLETAS - "DIRECT DEDICATED COMPLETE" - estrutura regular de conexão, também conhecida como conexão completa - "complete interconnection" -, na qual cada nó encontra-se simetricamente ligado a todos os outros nós, apresentando por isto uma baixa *cost-modularity*, e boa *place-modularity*; a mensagem entre os nós são transferidas somente por um único caminho entre os mesmos.

Sua flexibilidade nas conexões é alta e possui boas vantagens quanto aos efeitos de falha e boas possibilidades de reconfiguração. Não apresenta gargalos evidentes, e sua complexidade lógica de ligação é relativamente baixa.

Como exemplo de tal estrutura podemos citar o MERIT da Michigan University (7).

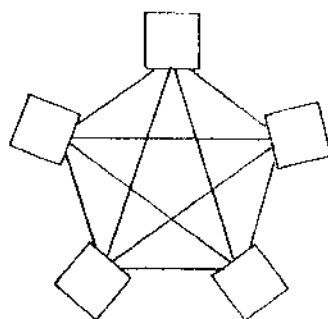


Figura 2.11 - DDC

2.4.3. - DSM - MEMÓRIA COMPARTILHADO - "DIRECT SHARED MEMORY" - apresenta como característica o fato dos nós se comunicarem por intermédio de uma memória de acesso comum.

Esta estrutura apresenta características muito boas quanto à modularidade; o custo de expansão dependendo da estrutura pela qual os processadores têm acesso ao sistema de memória pode ser alto, se para cada processador houver um caminho direto, e baixo se houver conexão por barramento. Apresenta baixa complexidade lógica; quanto às falhas apresenta uma boa qualificação, boa em relação aos processadores e baixa em termos da unidade central de memória, local de um possível gargalo.

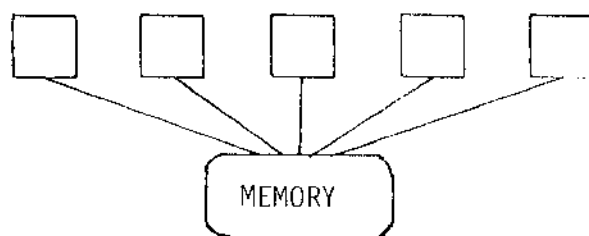


Figura 2.12 - DSM

2.4.4 - DSB - BARRAMENTO COMPARTILHADO - "DIRECT SHARED BUS" - arquitetura na qual os elementos processadores compartilham um barramento comum para a realização de suas comunicações. As mensagens são enviadas diretamente do nó fonte sobre o barramento, para serem reconhecidas e aceitas pelo destinatário. Apresentam baixa qualificação quanto ao caminho das mensagens, e boa em termos de adição de novos processadores. Similarmente, em relação a falhas nos dispositivos e reconfiguração da rede apresentam características muito boas, com relação ao processador e pobre com respeito ao barramento. Este tipo de estrutura é largamente utilizada em pesquisas aeroespaciais onde caminhos série e paralelos são utilizados.

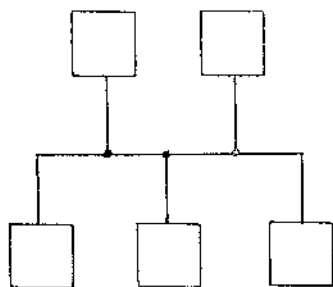


Figura 2.13 - DSB

2.4.5 - CONEXÃO ESTRELA - "INDIRECT CENTRALIZED DEDICATED STAR" - também conhecido como estrutura estrela, consiste de um dispositivo de comutação central ao qual cada nó está conectado por um caminho bidirecional. A troca de informações entre os nós é efetuado usando o comutador central como intermediário, que é o destino e fonte aparente para todas as mensagens. Apresenta características boas quanto a modularidade e falhas com relação aos processadores, tendo características ruins quanto à falhas no comutador central, um gargalo em potencial.

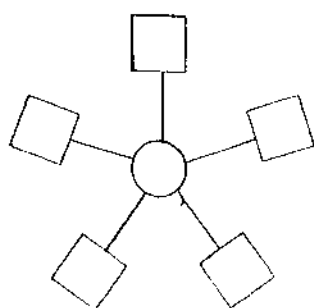


Figura 2.14 - ICDS

2.4.6 - IC DL - ANEL CENTRALIZADO - "INDIRECT CENTRALIZED DEDICATED LOOP" - consiste da implementação de conexões diretas inter dispositivos em uma estrutura ICDS. Nesta configuração as mensagens são colocadas no barramento pelos remetentes e removidas para uma operação de mapeamento de endereços por um dispositivo central de comutação sendo então recolocadas no barramento para seu destinatário. Sendo combinação das estruturas DDL e ICDS apresenta as mesmas características do DDL quanto as falhas em relação ao caminho de dados. Quanto ao comutador central, tratando-se de suas possíveis falhas apresenta características idênticas ao ICDS, ou seja ruins.

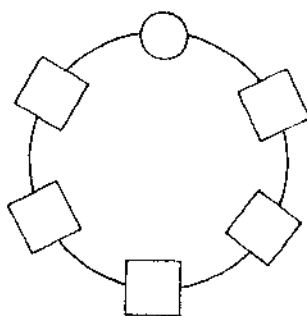


Figura 2.15 - IC DL

2.4.7 - ICS - COMPARTILHADO CENTRALIZADO - "INDIRECT CENTRALIZED SHARED" - esta configuração é funcionalmente equivalente a estrutura ICDS, com a diferença que os nós não estão individualmente conectados ao recurso comutador; em vez disto, compartilham um barramento de acesso ao comutador. Portanto, quando um elemento processador deseja transmitir uma mensagem, este, inicialmente deve requerer o barramento, e então transmitir a mensagem para o comutador; do comutador a mensagem é retransmitida sobre mesmo barramento para o destinatário. A retransmissão de mensagens pelo comutador central é a principal característica diferenciadora desta configuração em relação ao do tipo DSB e DSM. Apresenta boas vantagens no custo e na localização de novos nós na expansão, e características pobres em relação ao comutador. Similarmente apresenta boas características em relação às falhas de processador no sentido de uma reconfiguração e características pobres quanto ao efeito causados por falhas no comutador. Sua complexidade lógica é baixa e apresenta como gargalo em potencial a capacidade do comutador.

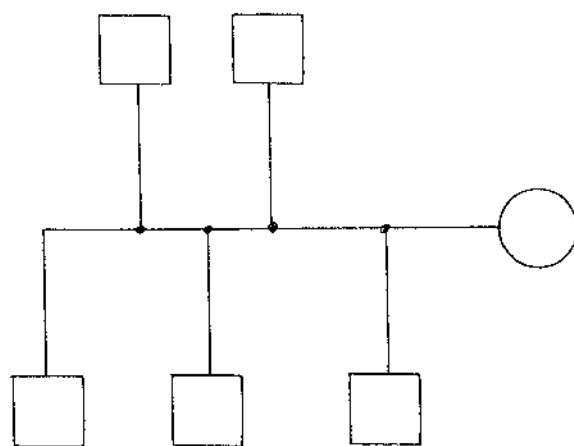


Figura 2.16 - ICS

2.4.8 - IDDR - DESCENTRALIZADO REGULAR - "INDIRECT DECENTRALIZED DEDICATED" - estrutura espacial regular de conexão, que apresenta como característica o fato de cada nó possuir vizinhanças à esquerda, à direita, em cima e em baixo. Podemos notar que a estrutura DDL é um caso particular desta, no qual cada elemento processador tem dois vizinhos. Apresenta características de modularidade extremamente pobre, advinda da rigidez da arquitetura, ou seja,

absoluta regularidade. Quanto à reconfiguração por falhas é muito pobre apresentando todavia uma possibilidade de superar os efeitos decorrentes do mau funcionamento de um nó pela implementação de diferentes caminhos de mensagens caracterizando assim a recuperação perante a falha de moderada a boa. A elegância desta estrutura tem causado interesse acadêmico, mas suas dificuldades práticas tem inviabilizado as implementações.

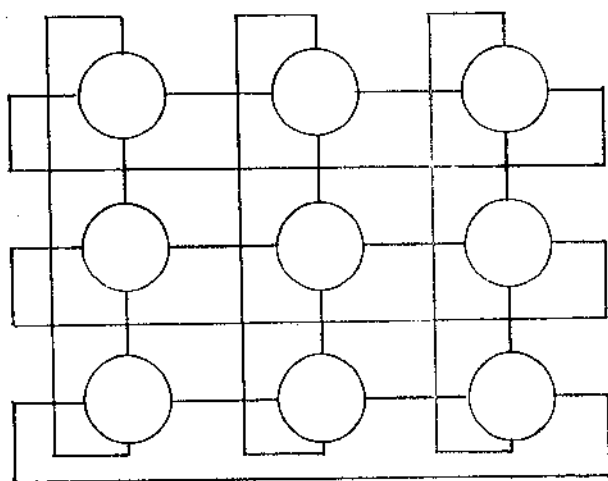


Figura 2.17 - IDDR

2.4.9 - IDDI - DESCENTRALIZADO IRREGULAR - "INDIRECT DECENTRALIZED DEDICATED" - é diferenciável do IDDR pela não regularidade das conexões. Os efeitos referentes às falhas são proporcionais e dependentes da estrutura. O gargalo não é um problema em potencial a menos de projetos específicos e usualmente a complexidade lógica das ligações tende a ser extremamente alta. A aplicação dominante deste tipo de estrutura é para rede de computadores geograficamente dispersos.

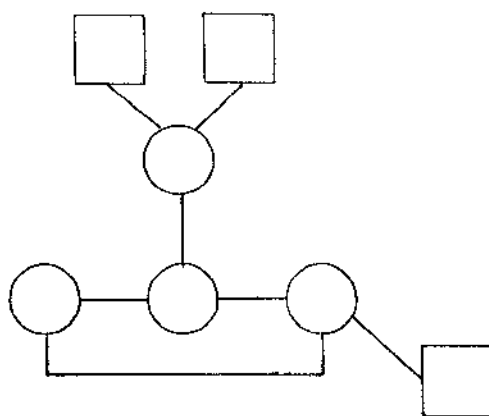


Figura 2.18 - IDDI

2.4.10 - IDS - COMPARTILHADO DESCENTRALIZADO - "INDIRECT DECENTRALIZED SHARED" - comutação de mensagens realizada por mais de um dispositivo, sendo a comunicação transmitida por barramento compartilhado. Quanto à modularidade, suas características tendem a ser extremamente boas, com características deficientes quanto à falhas, pois uma simples ligação pode afetar quase todo o sistema. A menos de estruturas específicas, não se detecta facilmente um gargalo e a sua complexidade lógica é usualmente alta.

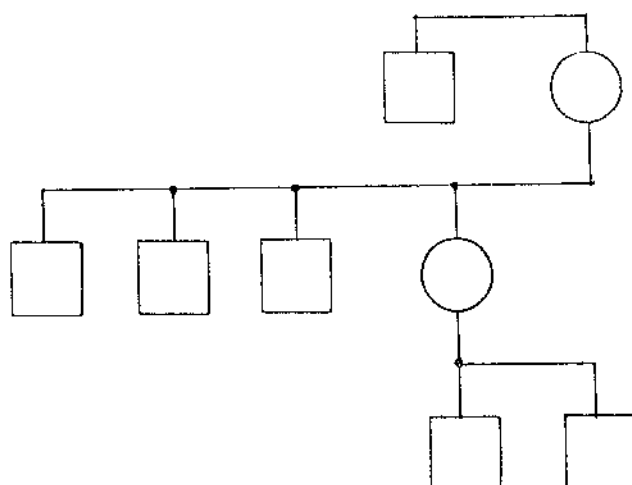


Figura 2.19 - IDS

## 2.5 - Barramento

Um sistema de computador é composto pela combinação de dispositivos de entrada/saída, de memórias, da unidade lógica e aritmética, e de circuitos auxiliares. Quando um dispositivo ou uma unidade é conectado com outro, há a necessidade de uma interface que contenha a lógica necessária.

A grande desvantagem do uso de um grande número de cabos individuais para interconectar as partes de um sistema são o alto custo e a complexidade.

Uma técnica largamente utilizada para obter uma interface modular, eficiente e com baixo custo, é a aplicação de um único barramento para interconectar todas as unidades. Barramentos são elementos vitais num sistema digital, ele interliga registros, subsistemas e sistemas.

Em qualquer sistema de computador há muitos barramentos ,

tais como:-

- . barramento interno, conectando registros e lógica aritmética com o processador central.
- . barramento de entrada/saída, conectando o processador com as memórias e dispositivos periféricos.
- . barramento de comunicação externa interligando o sistema de computação ao telefone ou a outro meio de comunicação.

Dois tipos de barramentos podem ser distinguidos:-

- . barramento série
- . barramento paralelo.

2.5.1 - BARRAMENTO SÉRIE - é comum o uso do sistema serial para a transmissão de dados com uma taxa de 0 a 500 caracteres por segundo, que requer somente um ou dois fios para transmitir todos os sinais necessários entre módulos.

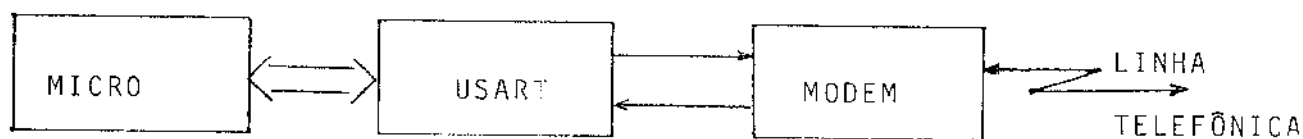


Figura 2.20

Cabe realçar que para este tipo de transmissão há a necessidade de uma interface que serialize a informação enviado em paralelo pelo microcomputador e tenha a capacidade de receber em série e enviar em paralelo para o microcomputador.

Em virtude do uso bastante comum da interface série os fabricantes de semicondutores fornecem circuitos USART "Universal Synchronous/Asynchronous Receiver/Transmitter" que é um "CHIP" LSI na qual estão implementados os conversores paralelo-série e série-paralelo.

Cada caracter série consiste basicamente de 4 partes:-

- . 1 bit de inicialização
- . 5 a 8 bits de dados
- . opcionalmente podemos ter um bit de paridade



. 1 a 2 bits de parada - "stop bit".

2.5.1.1 - baud-rate - é a taxa de transmissão em bits por segundo; por exemplo - sendo o caracter composto por 11 bits, o qual é transmitido a uma razão de 10 caracter por segundo, isto corresponde a 110 "baud". O período de um bit é o inverso do "baud"; as taxas de comunicações mais comum comercialmente são:- 75, 110, 300, 600, 1200, 2400, 4800, 9600 e 19200 "baud".

2.5.1.2 - sinais de controle - os sinais de controle e estados habitualmente presentes em uma USART são:-

- . DTR - Terminal de dados pronto - "Data terminal ready".
- . RTS - Solicitação de envio de dados - "Request to send".
- . DSR - Conjunto de dados prontos - "Data set ready".
- . CTS - Limpo para envio - "Clear to send".
- . TE - Transmissor Vazio - "Transmitter buffer empty".
- . TRDY - Transmissor pronto para receber dados - "Transmitter buffer ready".
- . RDY - Receptor pronto - "Receiver Buffer ready".
- . SINC - sincronização externa.

As interfaces seriais podem operar no modo síncrono ou assíncrono.

2.5.2 - BARRAMENTO PARALELO - é empregado quando se faz necessário um sistema de comunicação em alta velocidade entre os vários módulos do sistema.

Este tipo de barramento transfere simultaneamente todos os bits de informação através de fios separados e deve dispor de linhas para endereços, dados e controle.

O barramento de dados é usado para transferência de informações entre o processador e qualquer outro dispositivo a ele conectado.

O barramento de endereços transporta o endereço na memória

ou da porta de E/S para o qual os dados devem ser transferidos.

As linhas de controle podem ser usadas para indicar ciclo de leitura ou escrita, endereço válido no barramento de endereço, pedido de interrupção, pedido para acesso direto à memória (DMA - "*Direct Memory Access*"), sincronismo, etc.

## CAPÍTULO 3

### 3. INTERFACE

#### 3.1 - Introdução

Antes de entrarmos em detalhes sobre a interface teceremos algumas considerações sobre os requisitos que devem ser satisfeitos.

Para se projetar uma interface há a necessidade de conhecer a estrutura do barramento, a arquitetura do computador e as características de E/S dos dispositivos que pretendemos interconectar.

Uma interface para providenciar um elo de comunicação efetivo deve oferecer flexibilidade, custo razoável e compatibilidade com outros sistemas.

A interface pode ser projetada utilizando-se componentes discretos, circuitos integrados de baixa, média ou alta integração - SSI, MSI, LSI.

Optamos por LSI, pois estes componentes apresentam confiabilidade muito boa, apresenta um custo relativamente baixo, além de o número de "chíps" necessários ser bastante reduzido, o que implica em menor espaço ocupado, bem como maior facilidade na realização e manutenção.

O projeto foi desenvolvido em torno da utilização do integrado 8259 da INTEL - PIC-Controlador de Interrupção Programável. Como o PIC tem a capacidade de aceitar até oito níveis de interrupções, isto significa que com mais alguns integrados para satisfazer os circuitos auxiliares poderemos ter uma rede com 1 micro comandando até oito micros em um nível inferior.

Como o nosso laboratório dispõe do microcomputador SDK-86 que é um sistema baseado no microprocessador 8086 da INTEL, de 16 bits, utilizamos este sistema como "master", ou seja, em um nível superior comandando as ações dos "slaves", que são os micros do tipo 8080, Z-80, SIGNETICS2650.

Elegemos o sistema 8086 como "master" porque este sistema é comparável a um minicomputador de porte médio em seu desempenho, permitindo códigos re-entrantes, programas relocáveis dinamicamente e capacidade de endereçamento até 1 megabyte, diretamente.

A estrutura básica de nossa rede de microcomputadores pode ser analisada na figura 3.1 e 3.2.

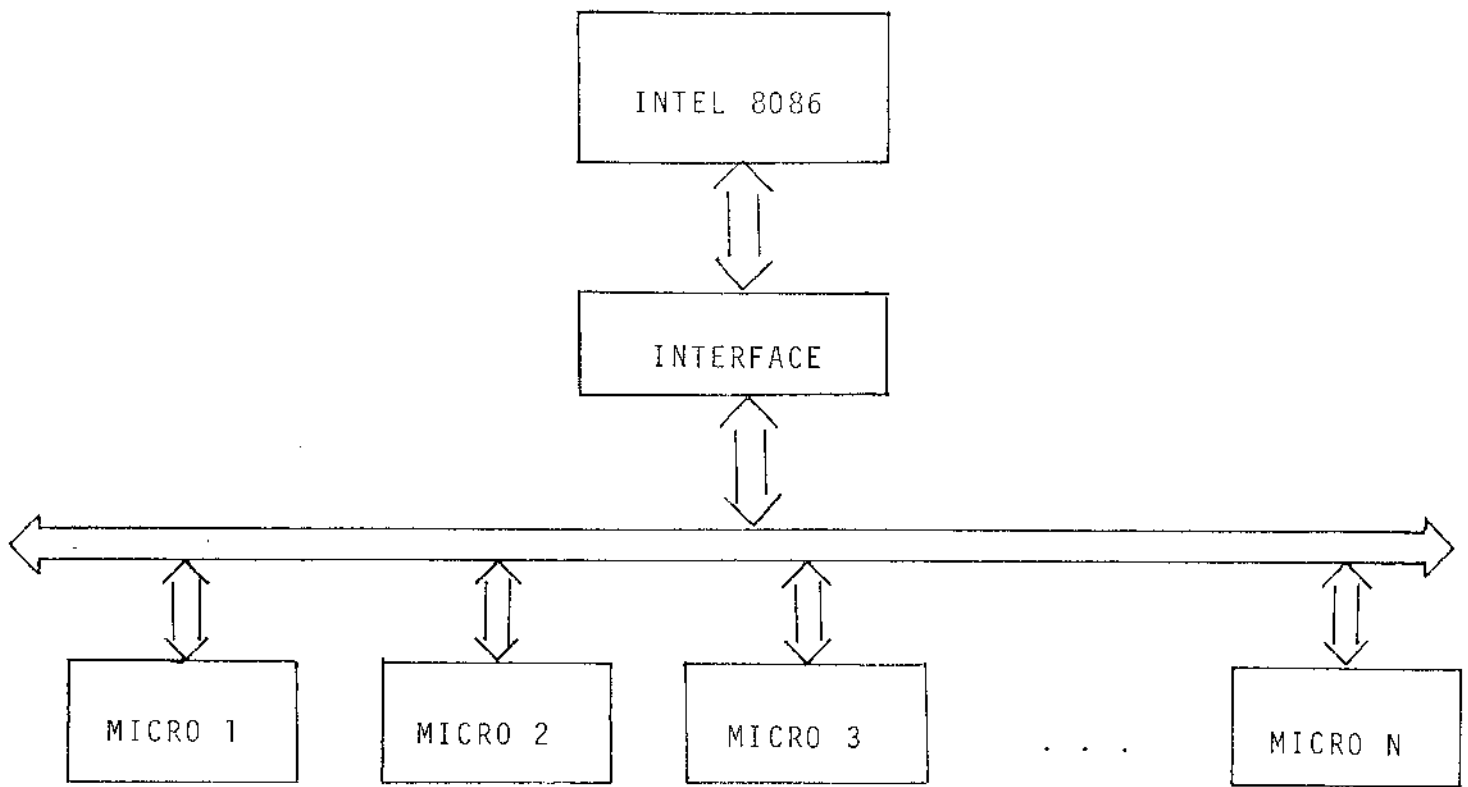


Figura 3.1

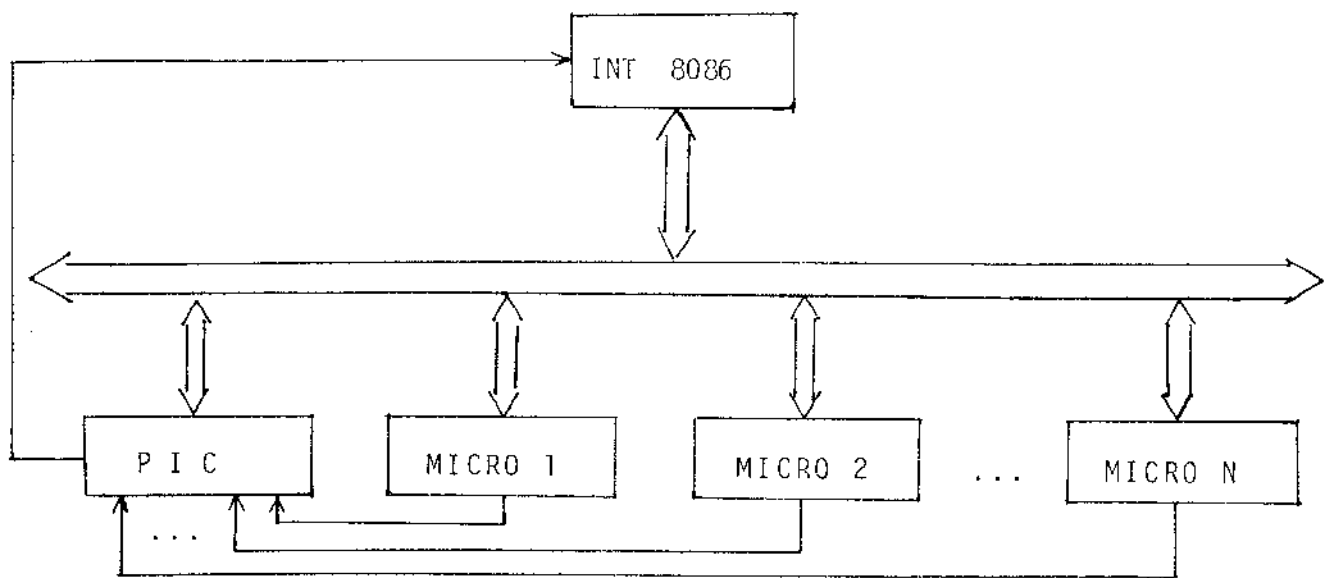


Figura 3.2

Os microcomputadores de nível inferior estão todos conectados no barramento do mestre, como se tratassem de dispositivos de E/S do 8086; esta estrutura foi realizada com os objetivos de se liberar as portas paralelas do 8086 para outros dispositivos periféricos, permitir DMA e tornar as transações mais rápidas.

A permissão para o uso do barramento por parte dos "slaves", só será concedida pelo CPU 8086 após a PIC verificar prioridades decidir qual dispositivo tem preferência, e então informar a CPU o endereço do dispositivo agraciado.

A estrutura de rede proposta se enquadra na configuração DSB - Barramento Compartilhado Direto, que apresenta características boas quanto à modularidade em relação aos processadores, bem como características boas quanto às falhas e reconfiguração dos processadores, além de apresentar uma complexidade lógica baixa. A estrutura tem como maior problema em potencial a possibilidade de congestionamento de tráfego de mensagens no barramento.

Como o objetivo principal deste trabalho é desenvolver uma estrutura de microcomputadores que realize o processamento distribuído e descentralizado para controle de processos, a configuração DSB foi escolhida pelo fato de apresentar um custo relativo baixo para a adição de novos processadores quando isto se fizer necessário. Os sistemas distribuídos de controle não apresentam alta taxa e envio de mensagens, o que torna plausível o uso da configuração DSB. Entretanto, o entrave do congestionamento em potencial pode ser solucionado adicionando-se um barramento série para efetuar transferências que não requer alta taxa de transmissão, amenizando assim o barramento paralelo.

### 3.2 - Funcionamento da 8259 - PIC

A sequência de eventos numa interrupção, passo a passo, é descrita a seguir:-

- . uma ou mais das linhas de requisição de interrupção tornam-se altas, ajustando os bits correspondentes no registro IRR;
- . a 8259 avalia as prioridades destas requisições e envia um sinal de pedido de interrupção, INT, à CPU, caso seja apropriado;
- . a CPU reconhece a interrupção enviando um pulso na linha

INTA;

- quando o "chip" 8259 recebe um pulso de INTA, o bit no registro ISR correspondente à requisição de mais alta prioridade é ajustado e o bit correspondente no registro IRR é zerado. Em outras palavras, quando a requisição de mais alta prioridade é atendida e colocada em serviço o seu pedido é cancelado no registro IRR, e no registro ISR.

No caso de usarmos a CPU-8086, esta providencia um segundo pulso de INTA. Durante a ocorrência deste pulso a PIC libera um apontador de 8 bits sobre o barramento de dados, o qual será lido pela CPU-8086. Normalmente o bits do registro ISR é zerado ao fim da execução da subrotina de interrupção para onde o processamento da CPU foi deslocado, valendo-se do endereço indicado pelo apontador fornecido pela PIC. Isto completa o ciclo de interrupção.

### 3.3 - Funcionamento da Interface

Deve-se ter em mente que esta rede estará operando com um "master", 8086, e vários dispositivos "slaves". Sendo assim a CPU-8086 é que estará controlando as comunicações entre ela e os demais micros da rede.

Para controle de processos, qualquer anormalidade que ocorra no processo, o micro que está controlando esta parte deve solucionar o mesmo por si só ou com a intervenção do "master". Neste caso há a necessidade de se estabelecer um elo de comunicação que será efetuado através da porta paralela, com o envio de um pedido de interrupção, a ser acolhido pela PIC. Este, por sua vez resolverá os conflitos de prioridade, caso ocorram e remeterá um pedido de interrupção, INT à CPU-8086, seguindo-se então toda a sequência vista anteriormente.

Se a comunicação se processar no sentido do "master" para o "slave", o "master" deve endereçar o processador eleito. Este endereçamento irá interromper o micro correspondente, para que a comunicação seja completada.

A CPU-8086, "master", deverá dispor de diversas rotinas de tratamento de interrupções, mais especificamente deverá ter uma para cada micro da rede. Assim, ao ser interrompido pela PIC, o mesmo enviará um apontador diferente para cada micro a ela conectado. A

CPU-8086 saberá qual micro está sendo tratado conseqüentemente qual a comunicação a ser feita bem como que transferência de dados realizar. Ao endereçar então o micro que está sendo tratado, a CPU- 8086 o interromperá, através de um dos "strokes", STBR ou STRW, gerados apartir da decodificação do endereço em conjunto com os sinais READ ou WRITE; em caso de se desejar efetuar uma operação de leitura, de verá ser ativada a linha STBR.

Fica claro que há a necessidade de se implementar um sistema operacional que providencie a geração destes sinais de controle, bem como análise a validade ou não de um dado enviado por qualquer elemento da rede.

Cada micro da rede deverá ter o seu sistema operacional com a capacidade de efetuar um elo de comunicação com o micro "master", bem como reconhecer a requisição.

### 3.4 - Circuitos Implementados

Foi implementada uma ligação interconectando o micro SDK-86 com o Z-80.

O micro Z-80 foi selecionado aqui porque dentre os micros disponíveis nos laboratórios da FEC-UNICAMP era o que exigia o menor número de modificações em termos de "hardware" para efetivar a ligação proposta, além de ter um sistema operacional mais versátil para a confecção e depuração de programas.

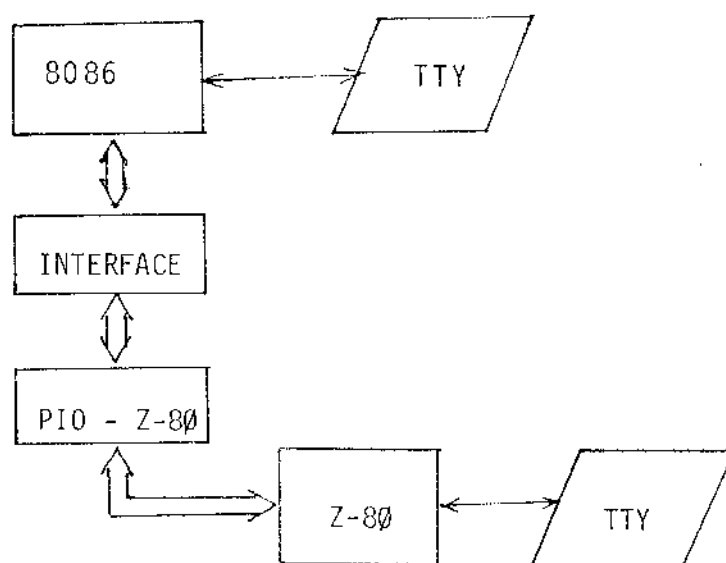


Figura 3.3 - ESTRUTURA DA REDE IMPLEMENTADA



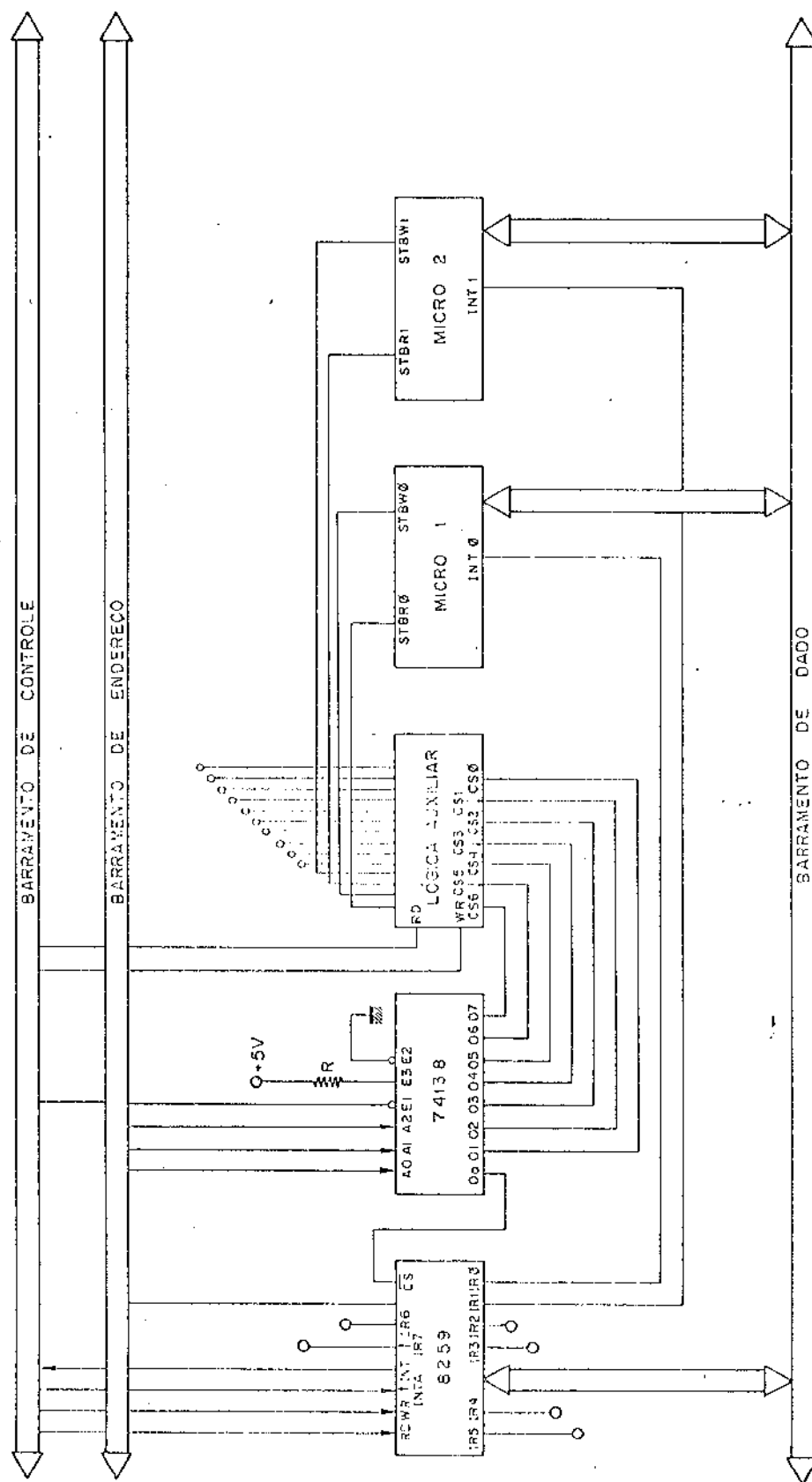


FIG. 3.4 - ESTRUTURA IMPLEMENTADA

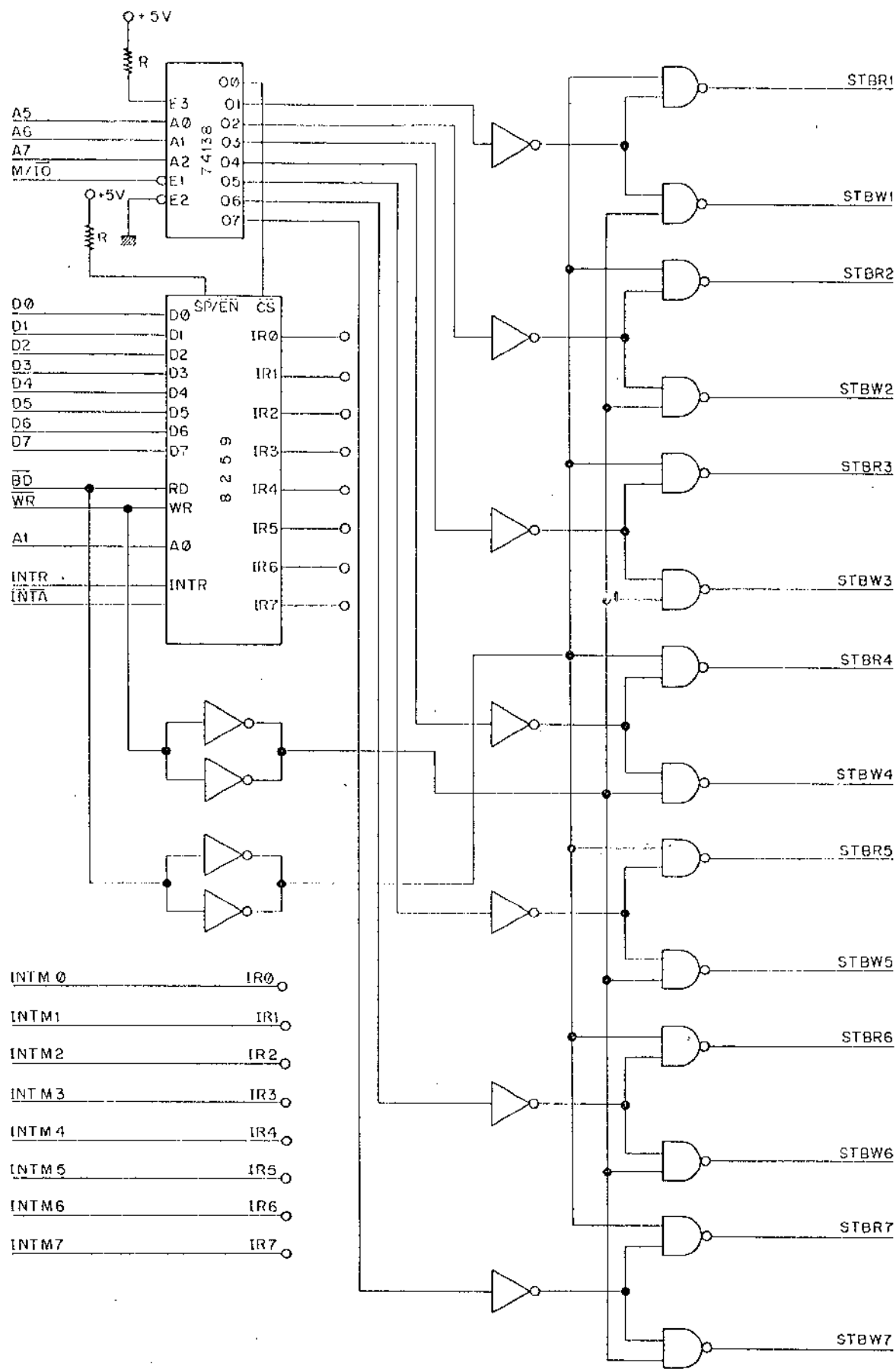


FIG. 3.5 - ESQUEMA ELÉTRICO DA INTERFACE

### 3.5 - Barramentos Utilizados

O sistema implementado consiste basicamente de dois barramentos interligando os dispositivos. A conexão entre o 8086 e a interface foi efetuada utilizando-se o mesmo barramento do micro 8086, por este motivo o 8086 trata os outros micros como periféricos. Os vários elementos do sistema SDK-86 são interligados por um sistema de barramento compatível com o padrão Multibus da Intel.

A utilização deste padrão é justificável porque o mercado oferece uma gama bastante ampla de periféricos compatíveis.

Para o segundo barramento; conexão entre as interfaces ; não houve a preocupação em seguir um padrão específico, visto que este barramento terá um índice de ocupação baixo. Quando se tratar de um sistema geograficamente distribuído podemos utilizar um barramento de comunicação série por razões de custo. Em nosso caso foi implementado um barramento paralelo; com 8 linhas bidirecionais de dados duas linhas de controle.

O terceiro barramento é o próprio barramento do micro conectado no nível inferior.

## CAPÍTULO 4

## 4. TESTES E CONSIDERAÇÕES FINAIS

### 4.1 - Testes de Funcionamento

Para caracterizar e testar o funcionamento da interface para a rede proposta foram feitos programas residentes nos dois microcomputadores.

Realizamos a comunicação nos dois sentidos, ou seja, o micro 8086 enviando dados para o micro Z-80 e o Z-80 enviando dados para o 8086. As duas unidades TTY são utilizadas para o acompanhamento dos processos de interrupção e como meio de comunicação com os processadores. Os programas elaborados para este fim são simples pois não tínhamos o objetivo de realizar um sistema operacional para tal finalidade, mas simplesmente testar o seu funcionamento. Neste sistema implementado, os eventos ocorrem como descritos a seguir:- os dois processadores são colocados a executar os seus respectivos programas, neste caso implementamos um programa que imprime continuamente uma mensagem no terminal a ele acoplado; se uma tecla do terminal acoplado ao Z-80 é acionado, o processador é interrompido, passando-se a executar uma outra rotina que envia o código correspondente, ASCII, ao caracter acionado no TTY, pela porta paralela ao micro 8086. O envio deste caracter ocasiona a interrupção do 8086, que passa a executar rotina específica. Para tornar visível o efeitos destas ações, o programa teste envia uma mensagem diferente ao terminal destacando o caracter recebido. Após este processo a execução volta ao programa principal. O processo inverso é bastante semelhante.

### 4.2 - Diagramas de Blocos

#### 4.2.1 - ROTINA DE INTERRUPCAO - 8086

- . preserva os registros - os principais registros são colocados em pilha de dados.
- . reinicialização da PIC - após o atendimento de uma interrupção a PIC necessita de uma nova palavra de controle.
- . inicialização da USART - garantir o funcionamento da USART de acordo com os propositos estabelecidos.

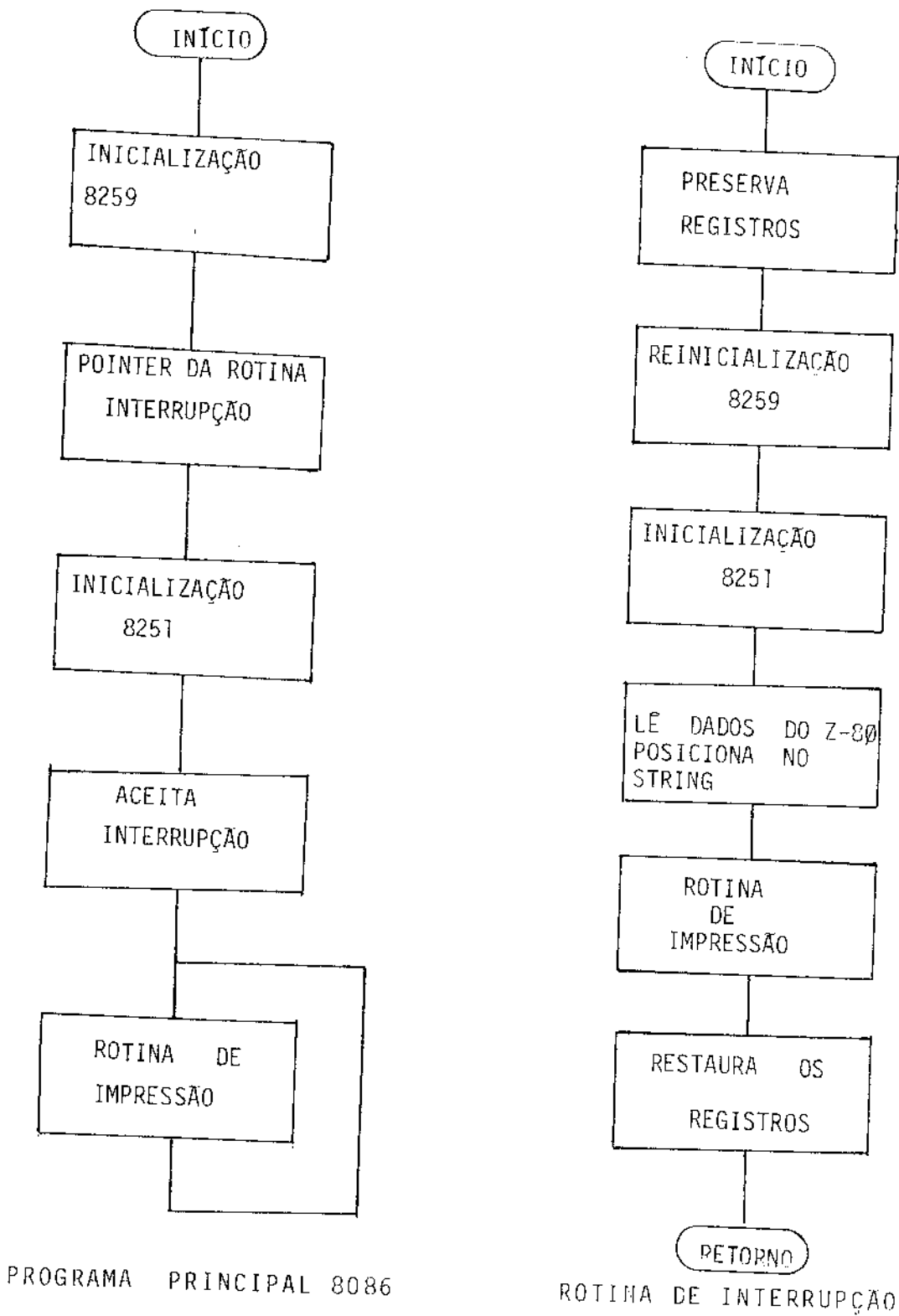


Figura 4.1

```

;*****
;

```

```

; PROGRAMA PRINCIPAL PARA TESTE DE COMUNICACAO - 9086
;

```

```

;*****

```

```

ORG 110

```

```

ICAPAC: EQU 433

```

```

; EQU 65

```

```

IEM EQU 419

```

```

INTI: DB 4036C

```

```

INTL: EQU 4036C

```

```

;*****
;

```

```

; ROTINA DE ATRASO
;

```

```

;*****

```

```

ATRASO: PUSH BP

```

```

MOV BP,SP

```

```

JMPA : CMP IBPI,C,00

```

```

JBE FIDDA

```

```

SUB IBPI,C,01

```

```

JMP COMDA

```

```

FILDA : POP BP

```

```

RET 02

```

```

PUSH BP ; INICIALIZA A 8250

```

```

MOV BP,SP

```

```

MOV AL,037

```

```

OUT 00

```

```

MOV AL,01

```

```

PUSH AX

```

```

CALL ATRASO

```

```

MOV AL,0FFH

```

```

OUT 02

```

```

MOV AL,01

```

```

PUSH AX

```

```

CALL ATRASO

```

```

MOV AL,01

```

```

OUT 02

```

```

POP BP

```

```

MOV IBPI,IBPI ; CONFERENCIA DA ROTINA DE INTERUPCAO

```

```

MOV IBPI,IBPI

```

```

MOV OA,0FFH

```

```

MOV AL,0CF

```

```

OUT 0A

```

```

MOV AL,02

```

```

PUSH AX

```

```

CALL ATRASO

```

```

MOV AL,027

```

```

OUT 0A

```

```

SII : ACERTAR A CONTINUAÇÃO

```

```

MOV CA,0C00H

```

```

OUT 0: MOV OA,0FFH

```

```

MOV OA,CA

```

```

AC: IN OA

```

```

TEST AL,01

```

```

JZ INC

```

```

MOV OA,0FFH

```

```

MOV AL,0C00H

```

```

OUT 0A

```

```

JMP OUT

```

```

JMP ABVD
ESTE: DC #00
      DC #0A
      DC #2A
      DC #2A
      DC #23
      DC #2A
      DC #2A
      DC #20
      DC #20
      DC #5A
      DC #45
      DC #53
      DC #54
      DC #45
      DC #20
      DC #20
      DC #43
      DC #4E
      DC #40
      DC #55
      DC #4E
      DC #69
      DC #43
      DC #41
      DC #43
      DC #41
      DC #4F
      DC #20
      DC #20
      DC #49
      DC #4C
      DC #4E
      DC #47
      DC #20
      DC #2F
      DC #20
      DC #49
      DC #4E
      DC #45
      DC #4C
      DC #20
      DC #20
      DC #2A
      DC #2A
      DC #23
      DC #23
      DC #2A
      DC #00
      DC #0A

```



ORG 500

```

;*****
;
;  ROTINA DE INTERUPCAO 8086
;
;*****

```

NLMCAR: EQU #23

TAB: DA #530

PUSH ES : PRESERVA REGISTROS

PUSH DS

PUSH AX

PUSH CX

PUSH DX

PUSH BX

PUSH SI

PUSH DI

PUSHF

MOV AL,#37 : REINICIALIZACAO DA #259

OUT 00

MOV DX, #FFFF2, INICIALIZACAO DA #251

MOV AL,0CF

OUT 0A

MOV AL,02

PUSH AX

CALL ATRADO

MOV AL,#27

OUT 0A

IN #30 : DE DADO DA 2-80

MOV (IAB),AL

MOV CX,00000000

OUT00: MOV DX,0FFFF2

MOVY DX,CX

INCPB: IN DX

TEST AL,01

JZ INCPB

MOV DX,0FFFF2

MOV AL,CABAC

OUT 0A

LOOP OUT00

MOV (IAB),20 : RESTAURA O CONTEUDO DE (IAB)

POPF : RESTAURA REGISTROS

POP SI

POP DI

POP BX

POP DX

POP CX

POP AX

POP DS

POP ES

IRET

HOUVE: DC #00

DC #0A

DC #2A

DC #30

DC #7A

DC #30

DC #7A

DC #30

DC #20

DC #20

TAB:

DC #48  
DC #4F  
DC #55  
DC #56  
DC #45  
DC #20  
DC #20  
DC #56  
DC #20  
DC #20  
DC #20  
DC #50  
DC #20  
DC #20  
DC #2A  
DC #30  
DC #2A  
DC #30  
DC #7A  
DC #30  
DC #2A  
DC #00  
DC #0A

- . leitura de dado enviado pelo Z-80 e o armazenamento em uma posição pré-determinada.
- . impressão da mensagem com o caracter recebido no terminal.
- . restauração dos registros a partir da pilha de dados.
- . retorno ao programa principal.

#### 4.2.2 - PROGRAMA PRINCIPAL - 8086

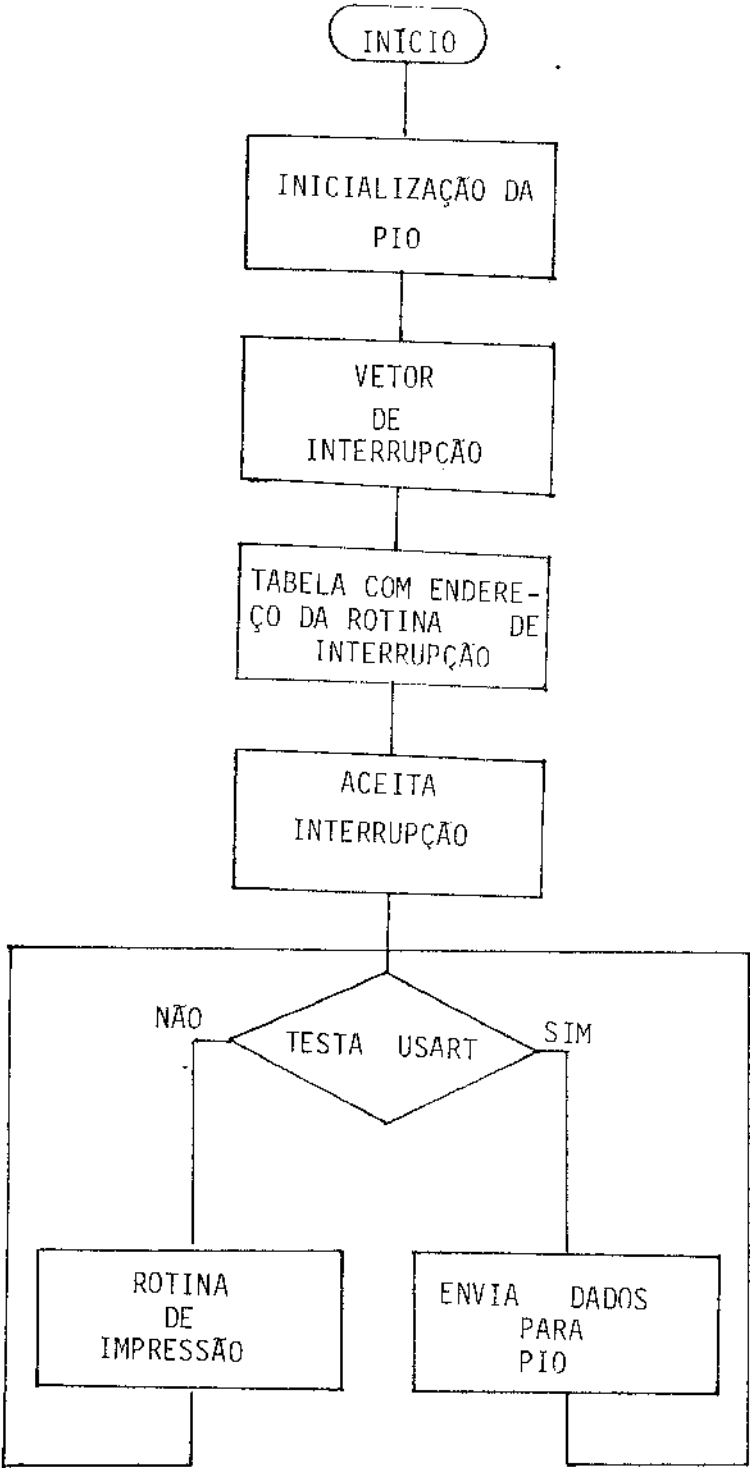
- . inicialização da PIC - há a necessidade de programá-la de acordo com as funções desejadas. Entre o envio de duas palavras de controle é necessário um atraso, pois este processo é bem mais lento que a execução de instruções pela CPU.
- . armazenamento dos apontadores da rotina de interrupção.
- . inicialização da USART.
- . ajusta bit de interrupções - as interrupções são liberadas.
- . programa de impressão de mensagem - programa do usuário.

#### 4.2.3 - ROTINA DE INTERRUPÇÃO - Z-80

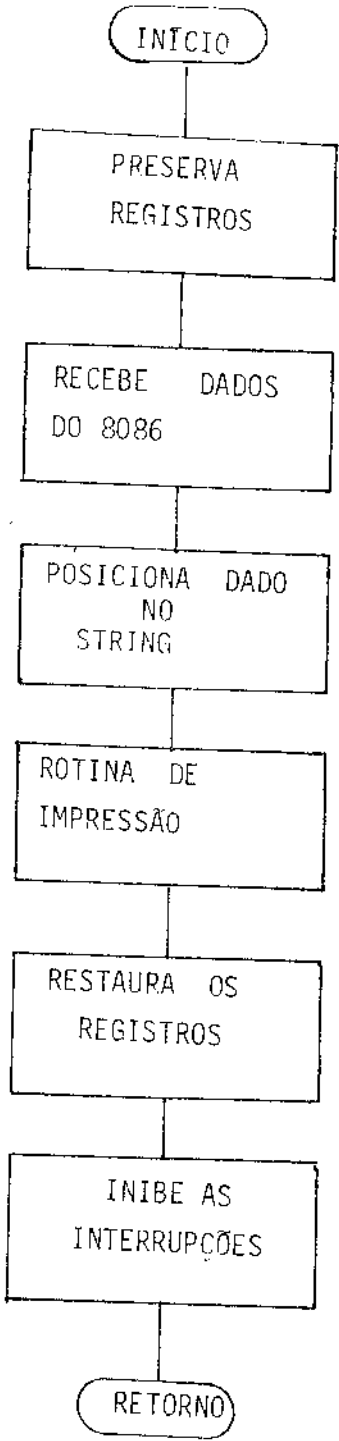
- . preserva os registros.
- . leitura de dado enviado pelo 8086, armazenamento em uma posição pré-determinada.
- . impressão da mensagem - imprime mensagem com o caracter recebido no terminal.
- . restaura registros.
- . retorna ao programa principal.

#### 4.2.4 - PROGRAMA PRINCIPAL - Z-80

- . inicialização da PIO - a PIO deve ser programada. Para a sua operação foi escolhida o modo bidirecional.
- . vetor de interrupção - o apontador para o endereço da rotina de interrupção é obtido compondo-se o conteúdo do registro de interrupção com um byte enviado pelo dispositivo que está inter-



PROGRAMA PRINCIPAL Z-80



ROTINA DE INTERRUPÇÃO

Figura 4.2

ORG 000

```

;*****
;
; PROGRAMA DE COMUNICACAO Z-80
;
;*****

```

PTXT: EQU #E3C7

```

LD A,#0F ; INICIALIZA A PIO
OUT 01,A
LD A,00
OUT 01,A
LD A,#87
OUT 01,A
OUT A,01 ; VETOR DE INTERRUPTAO
LD I,A
LD HL,#100 ; ENDERECO INICIAL DA ROTINA DE INTERRUPTAO
LD A,00
LD (HL),A
INC HL
LD A,02
LD (HL),A
IM 2
EI ; ACEITA INTERRUPTAO
STATUS: IN A,#00 ; TESTA O STATUS DA USART
AND A,01
JPZ PRESS
IN A,#0C
OUT 00,A
JP STATUS
PRESS: LD HL,TESTE
LD E,00
CALL PTAT
JP STATUS

```

ORG 200

```

;*****
;
; ROTINA DE INTERRUPTAO Z-80
;
;*****

```

```

PUSH PSW ; PRESERVA OS REGISTROS
PUSH S
PUSH D
PUSH H
IN A,#09 ; LE DADO DO 8086
LD (TAB),A
LD HL,NOUVE
LD C,00
CALL PTAT
LD (TAB),20
POP H ; RESTAURA REGISTROS
POP D
POP H
POP PSW
EI

```

ESTR: DC #0D  
DC #0A  
DC #2A  
DC #2A  
DC #23  
DC #2A  
DC #2A  
DC #20  
DC #20  
DC #54  
DC #45  
DC #53  
DC #51  
DC #45  
DC #23  
DC #20  
DC #43  
DC #4F  
DC #40  
DC #55  
DC #4E  
DC #49  
DC #43  
DC #41  
DC #43  
DC #41  
DC #4F  
DC #20  
DC #20  
DC #49  
DC #4C  
DC #4E  
DC #47  
DC #20  
DC #2E  
DC #20  
DC #49  
DC #4E  
DC #51  
DC #45  
DC #4C  
DC #20  
DC #20  
DC #2E  
DC #2A  
DC #23  
DC #2A  
DC #2A  
DC #0D  
DC #0A

HOUVE: DC #0D  
DC #0A  
DC #2A  
DC #30  
DC #2A  
DC #30  
DC #2A  
DC #30  
DC #20  
DC #20  
DC #48  
DC #4F  
DC #55  
DC #56  
DC #45  
DC #29  
DC #29  
DC #56  
DC #26  
AB: DC #29  
DC #26  
DC #50  
DC #20  
DC #20  
DC #2A  
DC #30  
DC #2A  
DC #30  
DC #2A  
DC #40  
DC #2A  
DC #00  
DC #0A

- rompendo o Z-80. Esta informação aponta para uma posição de memória a qual contém o endereço da rotina de interrupção:
- . ajusta bit de interrupção - passa a permitir interrupções.
- . testa o estado da USART.
- . se há caracter, envia o mesmo para a PIO, que o remeterá para o 8086.
- . se não há, salta para o programa do usuário - impressão de mensagem.

### 4.3 - Considerações Finais

Este trabalho contribui para a realização do projeto e implementação de uma rede de microprocessadores no Laboratório de Microcomputadores da Faculdade de Engenharia de Campinas.

A estrutura geral proposta é uma rede composta de 3 níveis, figura 4.1, tendo no 1º nível o sistema DEC-10; no 2º nível a proposta inicial coloca o minicomputador Interdata-80, este por ser um sistema dedicado e fazer parte do projeto SAIC em convênio com o IBC(Instituto Brasileiro do Café), mostrou-se pouco conveniente para uso devido às modificações necessárias. Por estes motivos optou-se por um microcomputador SDK-86 que apresenta capacidade comparável ao de um minicomputador; o 3º nível composto por vários micros de diversos tipos controlando os processos físicos e trocando informações com os demais níveis.

Foi implementada a interligação entre o 2º e o 3º níveis visando detectar e solucionar os possíveis problemas de "hardware".

Cabe salientar que há a necessidade de se desenvolver um "software" adequado para minimizar os possíveis gargalos, fazer um estudo visando a obtenção de um protocolo de comunicação mais refinado, bem como a melhor utilização dos sistemas operacionais já existentes nos micros.



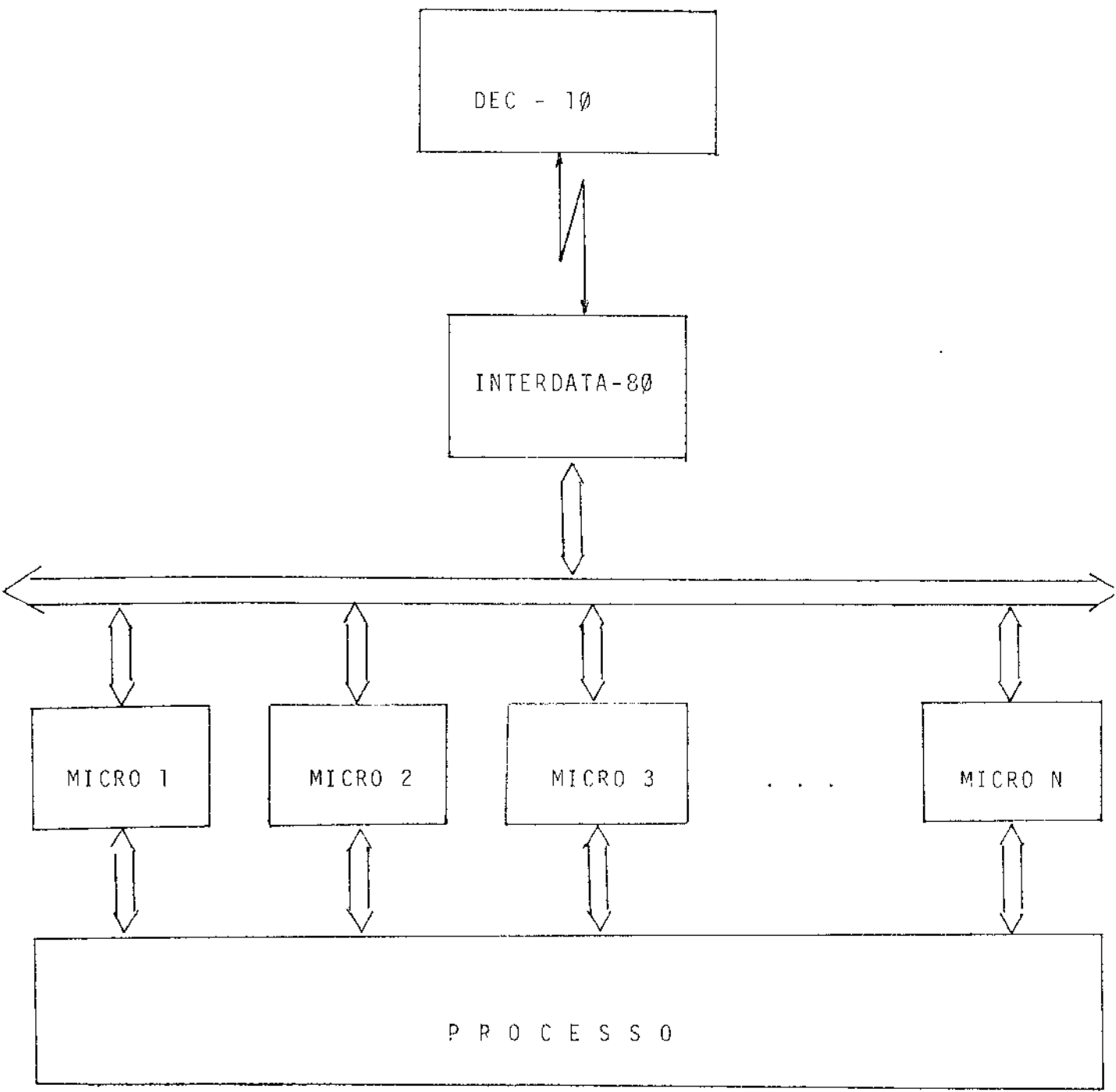


Figura 4.3 - ESTRUTURA GERAL PROPOSTA

## ANEXO A

Descreve-se a seguir os dispositivos diretamente envolvidos no projeto.

#### A.1 - Intel 8086

Foi desenvolvido de forma a estender a capacidade da família 8080 para o processamento de 16 bits em paralelo. Este processador possui atributos de processadores de 8 e 16 bits, através da execução do jogo de instruções do 8080/8085 adicionadas a um conjunto de instruções de 16 bits.

Como melhoramento pode-se citar:-

- . aritmética de 16 bits ;
- . aritmética com sinal de 8 e 16 bits , incluindo multiplicação e divisão;
- . código re-entrante;
- . relocação dinâmica de programas;
- . capacidade de endereçamento direto até 1 megabyte.

A figura A.1 mostra o diagrama de blocos funcional do 8086.

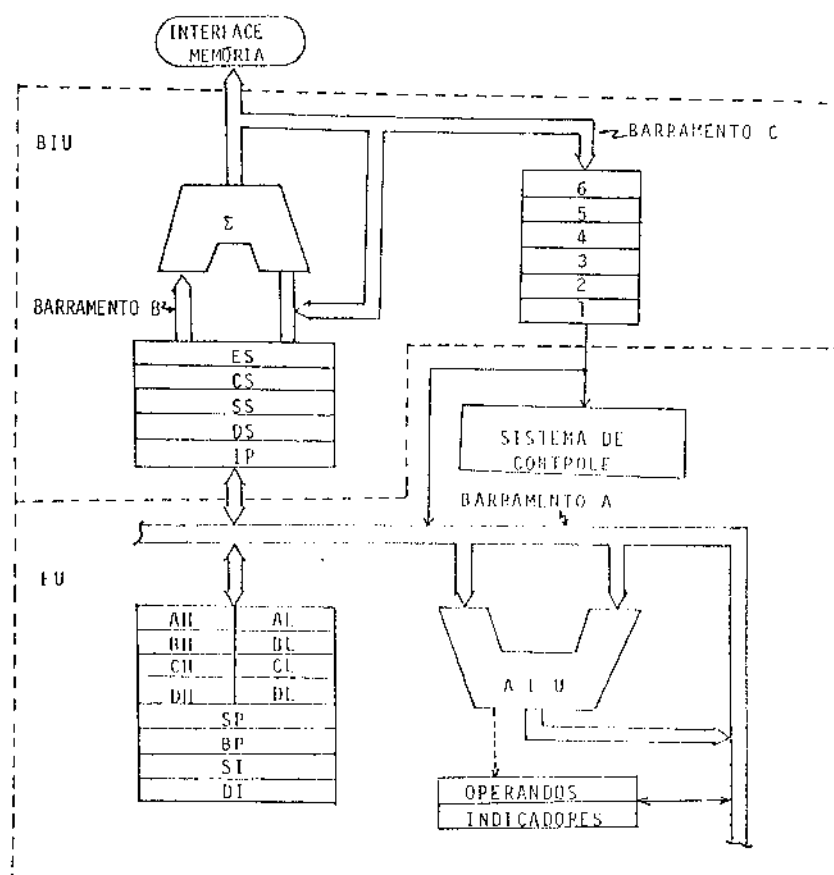


Figura A.1 - DIAGRAMA FUNCIONAL DO 8086

As funções internas são realizadas em duas unidades independentes.

A.1.1 - UNIDADE DE EXECUÇÃO (EU) - esta unidade realiza as funções básicas de processamento pois contém os registros de dados e a Unidade Lógica e Aritmética - "ALU". Desta maneira a EU recebe os operandos da memória através da BIU, processa-os e passa os resultados à BIU para armazenamento.

A.1.2 - UNIDADE DE INTERFACE DO BARRAMENTO (BIU) - o propósito desta unidade é maximizar a utilização do barramento, pois este é o principal fator de limitação de velocidade do processador. Esta maximização é procurada de duas formas:-

- a) - realizando uma busca prévia - "*prefetch*" - de instruções antes delas serem solicitadas pela EU. Estas instruções são colocadas numa fila que pode conter até 6 bytes aguardando a sua decodificação e execução; desta maneira a EU não necessita na maior parte das vezes esperar o final do ciclo do barramento para executar a próxima instrução.
- b) - realizando funções relacionadas à busca - "*fetch*" - e armazenamento dos operandos, relocação de endereços e controle do barramento, paralelamente ao processamento das operações na EU.

A.1.3 - REGISTROS - o 8086 contém 3 conjuntos de registros, contendo cada um 4 registros de 16 bits; um registro de estado - "*status*", contendo 9 indicadores - "*flags*" - de um bit cada e um registro apontador de instruções de 16 bits.

A.1.3.1 - registros gerais - AX, BX, CX, DX - estes registros participam das operações lógicas e aritméticas sem restrições.

A.1.3.2 - registro index e apontadores - SP, BP, DI, SI - os registros deste grupo são similares entre si, na medida em que, eles possuem um endereço de deslocamento - "*offset*" - utilizado para endereçamento dentro de segmentos de 64 kbytes; estes registros podem participar de operações lógicas e aritméticas.

A.1.3.3 - registro de segmentos - CS, DS, SS, ES - estes registros são utilizados na determinação de qualquer endereço da memória.

- . CS - segmento de código
- . DS - segmento de dado
- . SS - segmento de pilha - "stack"
- . ES - segmento extra.

A.1.4 - ENDEREÇAMENTO DE MEMÓRIA - os endereços da memória estão divididos em segmentos de 64 kbytes, que podem conter indistintamente instruções, dados e a pilha. O registro de segmento normalmente utilizado para a geração do endereço dos dados é o DS o que pode ser mudado através da utilização de um prefixo. Esta característica também ocorre com o registro SS, conforme mostrado na tabela A.1

Default	Prefixo
IP + CS = endereço do código	não
SP + SS = endereço da pilha	não
BP + SS = endereço da pilha	BP + DS ou ES, ou CS
EA + DS = endereço do dado	EA + ES ou SS, ou CS

Tabela A.1

A.2 - Zilog 80 - Z-80

É um processador de propósitos gerais, projetado como um melhoramento do INTEL 8080, seu código de máquina é compatível com o do 8080. O conjunto de instruções do Z-80 possui as instruções do 8080 como um subconjunto, entretanto a CPU Z-80 e os seus dispositivos de suporte não apresentam compatibilidade com os equivalentes do 8080.

Como melhoramento pode-se citar:-

- . a CPU é equivalente à CPU 8080 mais o controlador de "clock" (8224) e o controlador do sistema (8228).
- . utiliza somente uma fonte de alimentação -(5 volts).
- . os registros gerais e os indicadores - "flags" - foram duplicados.
- . instruções foram adicionadas de modo a se poder testar ou

alterar a condições de bit individuais do registro e memória.

- 2 registros de indexação foram adicionados, isto significa que algumas instruções do Z-80 podem utilizar endereçamento indexado.
- uma única instrução de movimento de bloco permite que os conteúdos de qualquer número de bytes adjacentes da memória sejam movidos de uma área da memória para outra, ou entre área de memória e uma porta de E/S. É possível pesquisar um bloco de memória com um determinado valor através da execução de uma instrução de comparação de blocos.

Um aspecto negativo do Z-80 é que os mnemônicos utilizados para o programa fonte são diferentes dos equivalentes encontrados no INTEL 8080.

### A.3 - Controlador de Interrupção Programável - PIC

Estrutura e pinagem do Controlador de Interrupção Programável - PIC 8259.

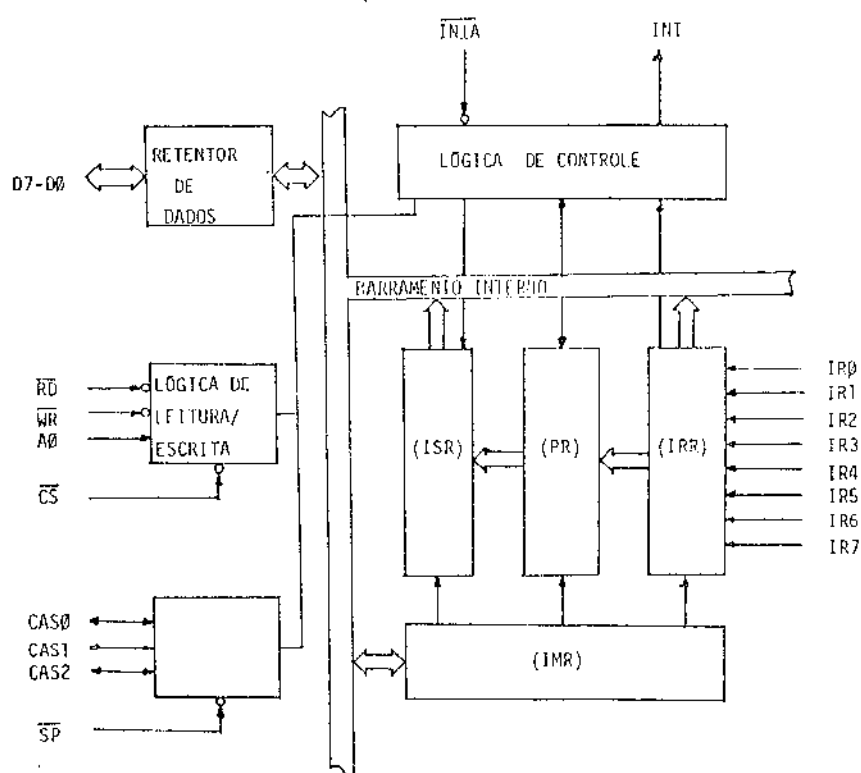


Figura A.2 - ESTRUTURA INTERNA DA PIC

- IR - "Interrupt Request Inputs" - 8 linhas de interrupção que podem ser controladas pela PIC.
- INT - "Interrupt Output" - sinal para a linha de interrupção da CPU.
- $\overline{\text{INTA}}$  - "Interrupt Acknowledge Input" - sinal de reconhecimento de interrupção, enviado pela CPU.
- $\overline{\text{CS}}$  - "Chip Select" - capacita a PIC realizar leituras e escritas através do barramento de dados.
- $\overline{\text{WR}}$  - "Write" - permite que as palavras de controle enviados pela CPU sejam introduzidas na PIC.
- $\overline{\text{RD}}$  - "Read" - permite que a PIC envie o estado dos diversos registros e níveis de interrupções ao barramento de dados.
- AO - "Command Select Address" - está conectado diretamente a uma das linhas de endereço. Age, juntamente com RD e WR, para que se tenha acesso ao registro da máscara de interrupção ou aos registros das palavras de comando.
- CAS - "Cascade Lines" - utilizado quando temos um mestre e até 8 escravos, 3 linhas.
- IRR - "Interrupt Request Register" - registro de requisição de interrupção, é usado para armazenar todos os níveis que requerem serviço.
- ISR - "In-Service Register" - registro de serviço
- IMR - "Interrupt Mask Register" - armazena os bits correspondente as linhas de interrupções a serem mascarados. O IMR opera sobre o registro ISR.
- PR - "Priority Resolver" - bloco lógico que determina a prioridade das interrupções.
- R/WC - "Read/Write Control Logic" - aceita comandos da CPU. Este bloco dispõe de registros que mantêm a palavra de comando de inicialização - ICW - e a palavra de comando de operações, que vão ditar as várias formas de operação do dispositivo. Permite também que o estado da PIC seja transferido para o barramento de dados.

Básicamente as interrupções são tratadas por dois registros em cascata; IRR e o ISR. IRR armazena as linhas de interrupções que estão requisitando serviço, e o ISR armazena as que estão sendo aten-

didás.

A.4 - Z-80 PIO

Pinagem do Z-80 PIO

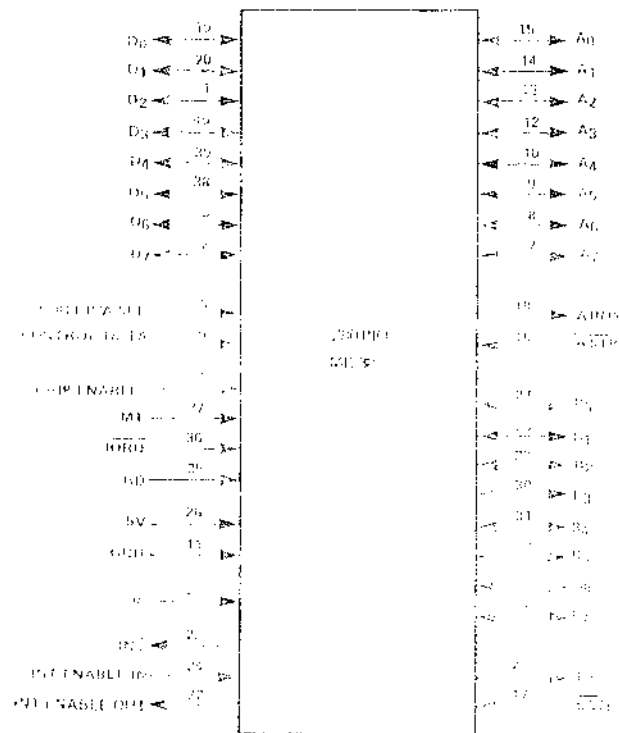


Figura A.3 - PINAGEM DO Z-80 PIO

- D7-D0 = "Z80-CPU Data Bus" - bidirecional - é usado para transferir todos os dados e comandos entre a CPU e a PIO.
- B/ASel = "Port B on A Select" - seleciona a porta A ou B, este sinal define qual a porta a que se deve ter acesso durante a transferência de dados entre a CPU e a PIO.



- C/DSel - "Control or Data Select" - seleciona dado ou controle; este sinal define o tipo de transferência a ser executada entre a CPU e a PIO. Um nível alto sobre este pino durante uma operação de escrita da CPU para a PIO faz com que o barramento de dados do Z-80 seja interpretado como um comando para a porta selecionada pela linha B/A; um nível baixo sobre esta linha significa que o barramento de dados do Z-80 está sendo usado para a transferência de dados entre a CPU e a PIO.
- $\overline{CE}$  - "Chip Enable" - um nível baixo sobre este pino permite que a PIO aceite comando ou dado vindo da CPU durante um ciclo de escrita, ou transmita dados para a CPU durante um ciclo de leitura.
- $\Phi$  - "System Clock" - relógio do sistema; a PIO usa a sinal de relógio padrão Z-80, para sincronizar alguns sinais internamente; este é um relógio de uma única fase.
- $\overline{M1}$  - "Machine Cycle One Signal" - sinal do primeiro ciclo de uma instrução na CPU; é usado como pulso de sincronismo para muitas operações interna da PIO. Quando  $\overline{M1}$  está ativo e o sinal  $\overline{RD}$  está ativo a CPU está buscando uma instrução na memória; quando  $\overline{M1}$  está ativo e o  $\overline{IORQ}$  está ativo, a CPU está reconhecendo uma interrupção. Além disso o sinal  $\overline{M1}$  tem outras duas funções dentro da PIO:-
- a) - sincronizar a lógica de interrupção da PIO;
  - b) - quando  $\overline{M1}$  ocorre na falta do sinal  $\overline{RD}$  ou  $\overline{IORQ}$  a lógica da PIO é re-inicializada (RESET).
- $\overline{IORQ}$  - "Input/Output Request" - pedido de E/S da CPU; este sinal é usado em conjunto com os sinais B/A, C/D,  $\overline{CE}$  e  $\overline{RD}$  para transferir comandos e dados entre a CPU e a PIO. Quando  $\overline{CE}$ ,  $\overline{RD}$  e  $\overline{IORQ}$  estão ativos, a porta endereçada por B/A deverá transferir dados para a CPU. Quando  $\overline{CE}$  e  $\overline{IORQ}$  estão ativos, mas  $\overline{RD}$  não está ativo, então na porta endereçada por B/A deverá ser escrita, pela CPU, um dado ou uma informação de controle. Se  $\overline{IORQ}$  e  $\overline{M1}$  são ativos simultaneamente, a CPU está reconhecendo uma interrupção e a porta interruptora deverá automaticamen-

te colocar um vetor de interrupção sobre o barramento de dados da CPU.

- $\overline{RD}$  - "*Read Cycle Status*" - ciclo de leitura de estado da CPU; se RD está ativo, uma operação de leitura de memória ou de E/S está em andamento. O sinal  $\overline{RD}$  é usado com os sinais B/A, C/D,  $\overline{CE}$  e o  $\overline{IO-RQ}$  para transferir dados da PIO para a CPU.
- IEI - "*Interrupt Enable In*" - interrupção de entrada permitida; este sinal é usado para formar um vetor de prioridade de interrupção; quando mais que um dispositivo com interrupção está sendo usado.
- IEO - "*Interrupt Enable Out*" - interrupção de saída permitida; é o sinal requerido para formar o esquema de prioridade.
- $\overline{INT}$  - "*Interrupt Request*" - requisição de interrupção; quando  $\overline{INT}$  está ativo a PIO está requerendo uma interrupção para a CPU.
- A7-A0 - "*Port A Bus*" - barramento da porta A; este barramento de 8 "*bits*" é usado para transferir dados, estados ou comandos entre a porta A e um dispositivo periférico.
- $\overline{ASTB}$  - "*Port A Strobe Pulse from Peripheral*" - pulso de "*strobe*" para a porta A do dispositivo periférico; o significado deste sinal depende do modo de operação selecionado para a porta A.
  - 1) - modo de saída - um sinal lógico positivo é enviado pelo periférico para reconhecer o recebimento de dados tornado disponível pela PIO.
  - 2) - modo de entrada - o "*strobe*" é enviado pelo periférico para carregar dados do periférico no registro de entrada da porta A.
  - 3) - modo bidirecional - quando este sinal é ativado, dados do registro de saída da porta A são liberados para dentro do barramento de dados da porta A.
  - 4) - modo de controle - o "*strobe*" é inibido internamente.
- ARDY - "*Register A Ready*" - registro A pronto; o signifi

cado deste sinal depende do modo de operação selecionado para a porta A.

1) - modo de saída - este sinal fica ativo para indicar que o registro de saída da porta A foi carregado e que o barramento de dados do periférico está estável e pronto para a transferência ao dispositivo periférico.

2) - modo de entrada - este sinal está ativo, quando o registro de entrada da porta A está livre e pronto para aceitar dados do dispositivo periférico.

3) - modo bidirecional - este sinal é ativado quando o dado está disponível no registro de saída da porta A para ser transferido do dispositivo periférico.

4) - modo de controle - este sinal é desativado e é forçado para o nível baixo.

B7-B0 - "Port B Bus" - barramento da porta B; este barramento é usado para transferir dados, estados ou informação de controle entre a porta B da PIO e o dispositivo periférico.

$\overline{\text{BSTB}}$  - "Port B Strobe Pulse from Peripheral Device" - pulso de "strobe" da porta B; o significado deste sinal é similar ao  $\overline{\text{ASTB}}$  com a seguinte diferença:-  
- no modo bidirecional da porta A este sinal controla dados do dispositivo periférico.

BRDY - "Register B Ready" - registro B pronto; o significado deste sinal é similar ao ARDY com a seguinte diferença :- no modo bidirecional da porta A este sinal está alto quando o registro de entrada da porta A está livre e apto a aceitar dados do dispositivo periférico.

## ANEXO B

## B. BARRAMENTOS

### B.1 - Barramento IEEE-488 (6)

Este barramento surgiu como resultado de 3 anos de discussão no IEC - "*International Electrotechnical Commission*". Em 1.974 o IEEE aprovou as restrições do IEC, surgindo como fruto o IEEE-488.

Trata-se de um barramento constituído de 16 linhas, sendo :-

- . 5 linhas de controle para supervisão da interface; 4 destas linhas são dirigidas pelo controlador - "*Controller*" - com a finalidade de emitir ordem para outros instrumentos sobre o barramento, como os dispositivos receptores e transmissores. A quinta linha é o SRQ, usado pelo dispositivo para informar ao controlador que o mesmo deseja o uso do barramento.
- . 3 sinais de controle de sincronismo usados para assegurar uma transferência de dados válidos entre o transmissor e qualquer número de receptores. Estes também asseguram a transferência de informações de controle entre o controlador e todos os outros dispositivos conectados ao barramento.
- . 8 linhas de dados usadas para transferir dados entre o transmissor e o receptor durante o modo de transmissão de dados. Este modo é sinalizado pelo controlador tornando a linha ATN=0. Estas linhas são utilizadas para transmitir informações de controle do controlador para todos os outros dispositivos durante o modo de transmissão de comandos, sinalizado pelo controlador tornando ATN=1.

#### B.1.1 - LINHAS DE CONTROLE

- ATN - "*Attention*" - quando falsa, indica que as linhas de dados contêm dados, quando verdadeiro, indica que as linhas de dados contêm um comando ou um endereço ocupando 7 linhas.
- IFC - "*Interface Clear*" - coloca o sistema em um estado conhecido.
- SRQ - "*Service Request*" - quando verdadeira, os sinalizadores da unidade de controle indicam que um dispositivo necessita de serviço.

REN - "*Remote Enable*" - ajusta o modo de cada dispositivo, em conjunto com outros códigos, para operar localmente ou remotamente.

EOI - "*End-or-Identify*" - é usada para assinalamento da unidade de controle, como fim de uma transferência de dados.

#### B.1.2 - LINHAS DE CONTROLE E SINCRONISMO

DAV - "*Data Valid*" - dado válido sobre o barramento de dados.

NDAC - "*Not Data Accepted*" - dado não aceito, quando verdadeira indica que o módulo do sistema está pronto para aceitar dados.

NFRD - "*Not Ready for Data*" - quando verdadeira significa que a informação não foi aceita pelo dispositivo receptor.

#### B.1.3 - LINHAS DE DADOS

D101/D108 - 8 linhas de dados bidirecionais.

### B.2 - Barramento 6800 (2)

Este barramento é composto por:-

- . 8 linhas bidirecionais de dados;
- . 16 linhas unidirecionais de endereços;
- . 9 linhas de controle

### B.3 - Barramento Multibus (16)

Os módulos que utilizam o Multibus tem uma relação mestre-escravo - "*master-slave*". O módulo mestre pode controlar o barramento acionando os comandos e endereços, o módulo escravo não pode controlar o barramento.

Este barramento permite processadores de 8 e 16 bits além de suportar vários dispositivos mestre.

Os sinais do Multibus podem ser agrupados em várias classes baseadas nas funções atribuídas. Estas classes são:-

- . linhas de controles
- . linhas de endereços
- . linhas de inibição
- . linhas de dados

- . linhas de interrupções
- . linhas de permutação de barramento
- . linha de falha no sistema de alimentação

### B.3.1 - LINHAS DE CONTROLE

- BCLK - "*Bus Clock*" - a borda de descida do BCLK é usado para sincronizar o circuito de resolução de prioridades do barramento. Se há mais de um mestre no sistema, somente um deverá ser utilizado para gerar o BCLK.
- CCLK - "*Constant Clock*" - é um sinal de relógio de frequência constante que pode ser usado pelos dispositivos conectados ao barramento como relógio mestre.
- MWTC - "*Memory Write Command*" - comanda a escrita na memória, indica que os dados devem ser escritos na localização de memória endereçada pelas linhas de endereço.
- MRDC - "*Memory Read Command*" - comando de leitura da memória, indica que o endereço foi colocado nas linhas de endereço e que o conteúdo da posição de memória endereçada deve ser colocado no barramento de dados.
- IOWC - "*I/O Write Command*" - comando de escrita no dispositivo de E/S, indica que uma porta de E/S foi ativada, devendo o conteúdo do barramento de dado ser escrito na mesma.
- IORC - "*I/O Read Command*" - comando de leitura do dispositivo de E/S, indica que uma porta de E/S foi ativada, devendo o conteúdo da porta ser colocado no barramento de dados.
- XACK - "*Transfer Acknowledge Signal*" - linha de reconhecimento do escravo a um pedido do mestre, indica para o mestre que a ação requerida foi completada, e que os dados foram aceitos ou colocados nas linhas de dados.
- INIT - "*Initialization Signal*" - é gerado para colocar todo o sistema num estado conhecido. Este sinal é gerado usualmente para começar

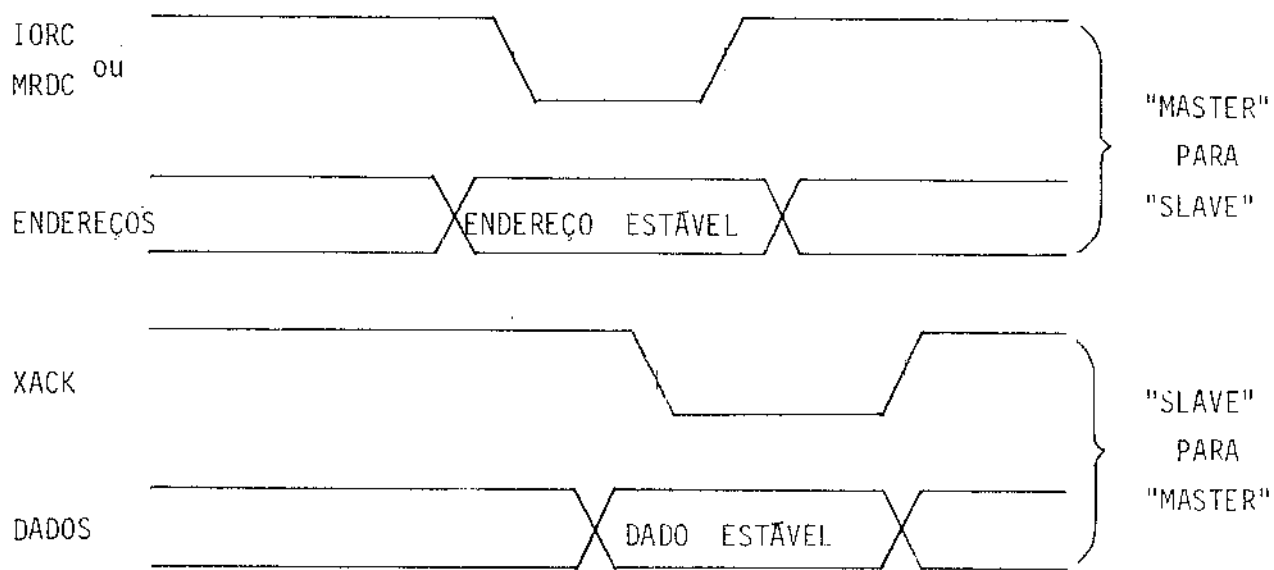


Figura B.1 - CICLO DE LEITURA

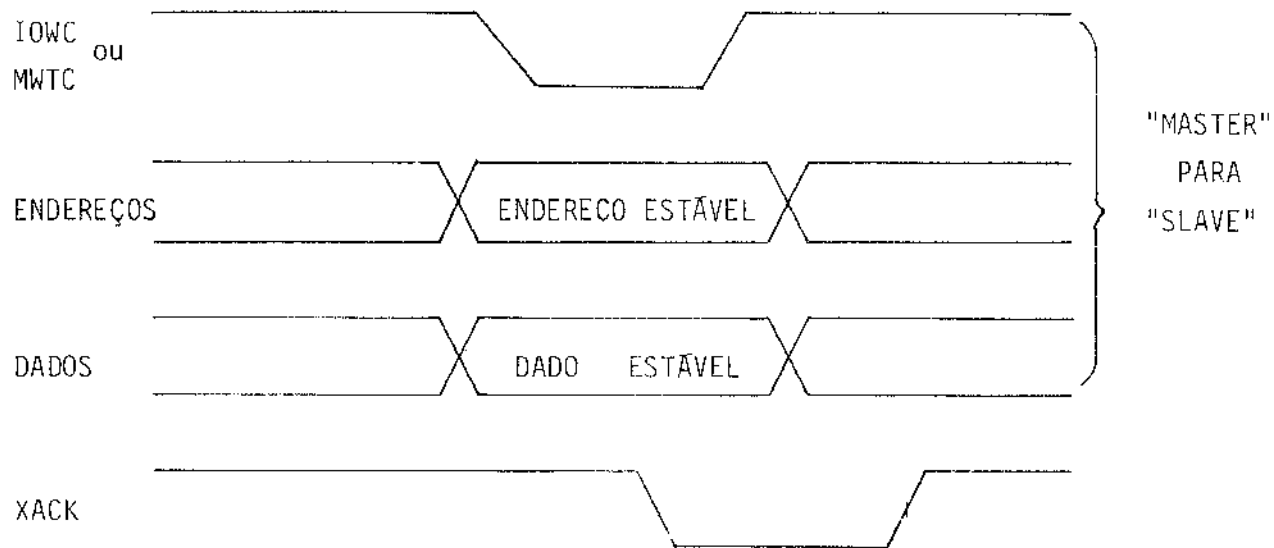


Figura B.2 - CICLO DE ESCRITA



qualquer operação com o sistema. INIT pode ser gerado por qualquer mestre ou por uma fonte externa, como uma chave "reset" de painel.

### B.3.2 - LINHAS DE ENDEREÇOS

ADR0/ADR13 - transmite o endereço de uma posição de memória ou de um dispositivo de E/S a ser referenciado. As 20 linhas de endereços permitem endereçar até 1 megabyte de memória. Para endereçar dispositivos de E/S são utilizadas 12 linhas (ADR0-ADRB) permitindo-se endereçar até 4.096 dispositivos.

### B.3.3 - LINHAS DE INIBIÇÃO

INH1 - "*Inhibit Ram Signal*" - evita que as memórias RAM respondam ao endereço de memória ativo no barramento. Este sinal permite que memórias ROM ou dispositivos de E/S mapeados na memória ocupem os mesmos endereços da RAM.

INH2 - "*Inhibit Rom Signal*" - evita que o módulo ROM responda ao endereço de memória ativo no barramento.

BHEN - "*Byte Control Line*" - é usado para selecionar o byte superior (DAT8-DATF). Este sinal é usado somente por sistemas que incorporam memórias e/ou módulos de E/S de 16 bits.

### B.3.4 - LINHAS DE DADOS

DAT0/DATF - 16 linhas de dados bidirecionais; são usados para transmitir ou receber de/para uma localização de memória ou dispositivos de E/S. Em sistemas de 8 bits são usadas as linhas DAT0-DAT7.

### B.3.5 - LINHAS DE INTERRUPÇÕES

INT0/INT7-interrupções são requisitadas ativando-

se uma das oito linhas. INT0 é o de mais alta prioridade e INT7 de mais baixa.

INTA - "Interrupt Acknowledge" - linha acionada pelo mestre em resposta a um pedido de interrupção, bem como para requerer a informação sobre a prioridade e o vetor de interrupção.

#### B.3.6 - LINHAS DE PERMUTAÇÃO DE BARRAMENTOS

BREQ - "Bus Request Signal" - usado por um dispositivo mestre no circuito de resolução de prioridades paralela para requerer controle do barramento com o finalidade de transferir um ou mais dados.

BPRN - "Bus Priority In Signal" - indica para um mestre que não há outro de prioridade mais alta requerendo o controle do barramento.

BPRO - "Bus Priority Out Signal" - usado no esquema de resolução série de prioridades. Este sinal é passado para a entrada BPRN do módulo mestre com a próxima prioridade.

BUSY - "Bus Busy Signal" - linha acionada pelo mestre indicando que o barramento está em uso. Este sinal evita que outros mestres obtenha o controle do barramento.

CBRQ - "Common Bus Request" - linha em coletor aberto que é acionada por todos os mestres em potencial, é usada para informar ao mestre ativo que um outro mestre deseja o uso do barramento. Se CBRQ está verdadeira, indica ao mestre ativo que não há outro mestre requisitando o barramento.

#### B.3.7 - LINHAS DE FALHAS NO SISTEMA DE ALIMENTAÇÃO

ACLO - "AC Low" - este sinal gerado pela fonte fica verdadeiro quando a voltagem da linha AC cai abaixo de uma certa voltagem, indicando que a fonte DC sairá de operação em 3 ms. ACLO torna-se falso quando todas as voltagens DC atingirem 95% do valor pré-fixado.

UNICAMP  
BIBLIOTECA CENTRAL

- PFIN - "*Power Fail Interrupt*" - este sinal interrompe o processador quando ocorre uma falha no sistema de alimentação; é acionado por um circuito externo.
- PFSN - "*Power Fail Sense*" - esta linha é um sinal de saída "*latch*" que indica a ocorrência de uma falta de energia. É cancelado pelo PFSR. Este "*latch*" deve ser alimentado por uma fonte externa.
- PFSR - "*Power Fail Sense Reset*" - esta linha é usada para reajustar o "*latch*" do sensor que indica falta de energia.
- MPRO - "*Memory Protect*" - evita operações com a memória durante o período incerto da fonte DC, inibindo a requisição de memória. MPRO deve ser acionado por uma fonte externa.
- ALC - "*Address Latch Enable*" - gerado pela CPU para providenciar um "*latch*" auxiliar de endereço.
- HALT - "*halt*" - indica que a CPU mestre está no estado "*halt*".
- AUX RESET - "*Auxiliary Reset*" - este sinal gerado externamente inicia a sequência de ligação das fontes de potência.
- WAIT - "*Bus Master Wait State*" - este sinal indica que o processador está no estado "*wait*".

#### B.4 - Barramento S-100 (17)

Originalmente este barramento foi definido tendo como padrão os sinais e as características do microprocessador INTEL 8080. A produção de microprocessadores com características diferentes; como as versões de alta velocidade e os micros de 16 bits com grande capacidade de endereçamento restringiu a aplicação deste padrão.

O barramento original apresenta, separadamente, 8 linhas para dados de saída e 8 linhas de dados de entrada. Para permitir o emprego com microprocessadores de 16 bits foi necessária a adição de dois sinais de reconhecimento. O barramento de endereço foi ampliado para 24 linhas.

O S-100 é organizado dentro de 8 conjuntos de linhas de si

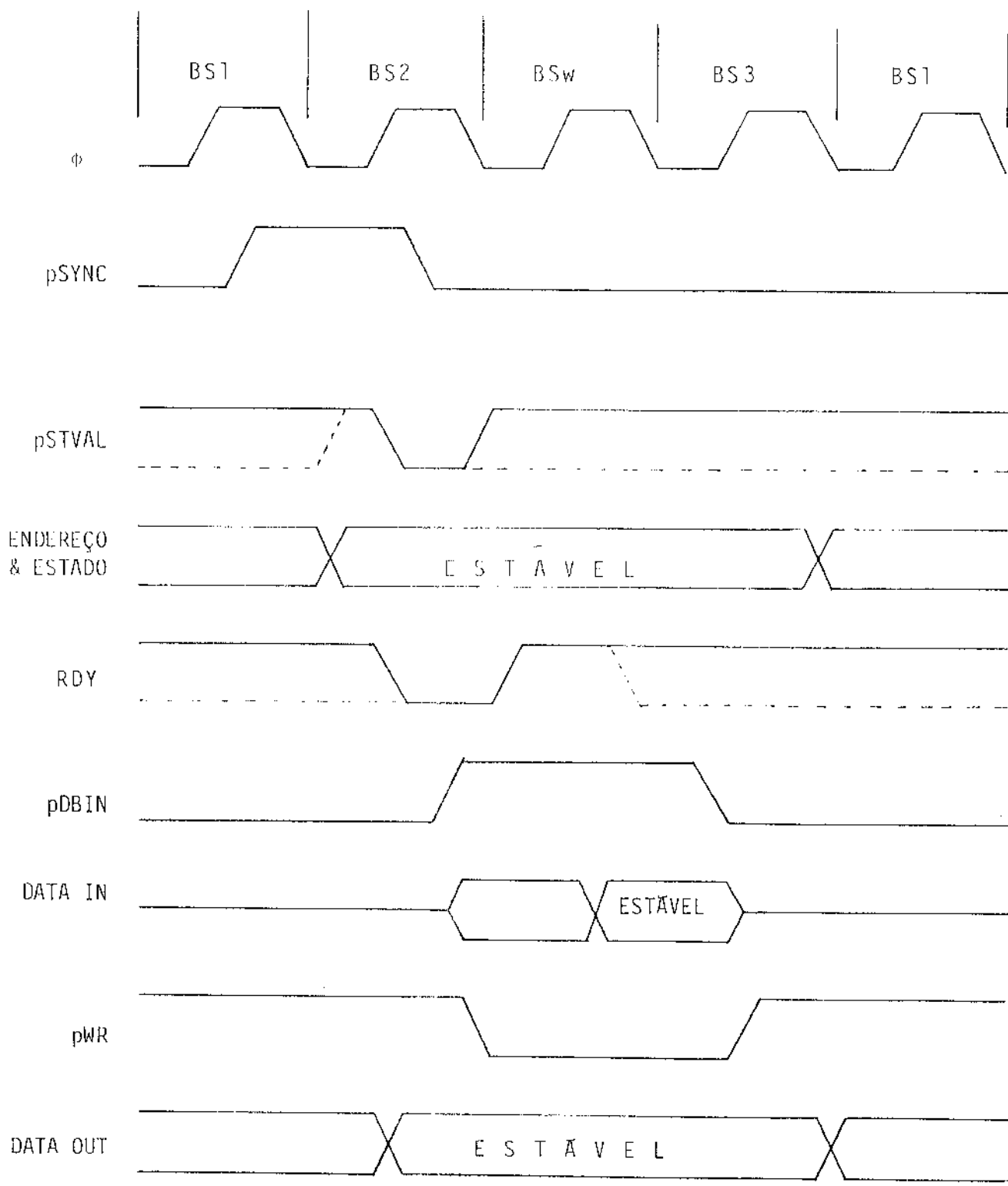


Figura B.3 - CICLO DE LEITURA E ESCRITA

PIRO Nº	SINAL/TIPO	NÍVEL	DESCRIÇÃO
01	+ 8 volts		
02	+ 16 volts		
03	XRDY(S)	H	
04	VI0(S)	L(OC)	Linha 0 do vetor de interrupção.
05	VI1(S)	L(OC)	Linha 1 do vetor de interrupção.
06	VI2(S)	L(OC)	Linha 2 do vetor de interrupção.
07	VI3(S)	L(OC)	Linha 3 do vetor de interrupção.
08	VI4(S)	L(OC)	Linha 4 do vetor de interrupção.
09	VI5(S)	L(OC)	Linha 5 do vetor de interrupção.
10	VI6(S)	L(OC)	Linha 6 do vetor de interrupção.
11	VI7(S)	L(OC)	Linha 7 do vetor de interrupção.
12	NMI(S)	L(OC)	Interrupção não mascarável.
13	PWRFAIL(B)	L	
14	DMA3(M)	L(OC)	
15	A18(M)	H	Bit 18 do endereço expandido.
16	A16(M)	H	Bit 16 do endereço expandido.
17	A17(M)	H	Bit 17 do endereço expandido.
18	SOSB(M)	L(OC)	Sinal de controle para inibir os sinais de "status".
19	CDSN(M)	L(OC)	Sinal de controle para inibir os sinais de controle de saída.
20	GND(B)		Terra.
21	NDEF		Não definido.
22	AUSB(M)	L(OC)	Sinal de controle para inibir as 16 linhas de endereço.
23	DOSB(M)	L(OC)	Sinal de controle para inibir as 8 linhas de dados.
24	Ø	H	
25	pSTVAL(M)	L	
26	pHLDA(M)	H	Sinal de controle usado em conjunto com o hold para coordenar as operações de transferência.
27	RFU		Reservado para uso futuro.
28	RFU		Reservado para uso futuro.
29	A5(M)	H	Bit 5 de endereço.
30	A4(M)	H	Bit 4 de endereço.
31	A3(M)	H	Bit 3 de endereço.
32	A15(M)	H	Bit 15 de endereço.
33	A12(M)	H	Bit 12 de endereço.
34	A9(M)	H	Bit 9 de endereço.
35	DO1(M)/DATA 1(M/S)	H	Bit 1 do dado de saída/bit 1 do dado bidirecional.
36	DO0(M)/DATA 0(M/S)	H	
37	A10(M)	H	Bit 10 de endereço.
38	DOA(M)/DATA4(M/S)	H	
39	DO5(M)/DATA5(M/S)	H	
40	DO6(M)/DATA6(M/S)	H	
41	DI2(S)/DATA 10(M/S)	H	Bit 2 do dado de entrada/bit 10 bidirecional
42	DI3(S)/DATA 11(M/S)	H	
43	DI7(S)/DATA 15(M/S)	H	
44	SMI(M)	H	Sinal que indica que o ciclo presente é uma busca de operando.
45	SOUT(M)	H	Sinal de status identificando a transferência de dados para um dispositivo de saída.
46	SINP(M)	H	Sinal de status identificando a transferência de dados de um dispositivo de entrada.
47	SMEMR(M)	H	Sinal de status identificando a transferência de dados da memória para o mestre.
48	SHLTA(M)	H	Sinal de status de reconhecimento que uma instrução de HALT está sendo executada.
49	CLOCK(B)		
50	GND(B)		

PINO Nº	SINAL/TIPO	NÍVEL	DESCRIÇÃO
51	+ 8 volts (B)		
52	- 16 volts (B)		
53	GND(B)		
54	SLAVE CLR(B)	L(OC)	Sinal para "reset" do escravo. Deve ser ativo com o POC.
55	DMA0(M)	L(OC)	Mestre temporário com prioridade 0.
56	DMA1(M)	L(OC)	Mestre temporário com prioridade 1.
57	DMA2(M)	L(OC)	Mestre temporário com prioridade 2.
58	SXTRQ(M)	L	
59	A19(M)	H	Bit 19 do endereço estendido.
60	SIXTN(S)	L(OC)	
61	A20(M)	H	Bit 20 do endereço estendido.
62	A21(M)	H	Bit 21 do endereço estendido.
63	A22(M)	H	Bit 22 do endereço estendido.
64	A23(M)	H	Bit 23 do endereço estendido.
65	NDEF		Não definido.
66	NDEF		Não definido.
67	PHANTOM(M/S)	L(OC)	
68	MNRT(H)	H	
69	RFU		Reservado para uso futuro.
70	GND(B)		
71	RFU		Reservado para uso futuro.
72	RDY(S)	H(OC)	
73	INT(S)	L(OC)	Sinal de requisição de interrupção.
74	HOLD(M)	L(OC)	Sinal de controle usado em conjunto com o HOLDA para coordenar as operações de transferência.
75	RESET(B)	L(OC)	Sinal de "reset" para o mestre. Deve ser ativo com o POC
76	pSYNC(M)	H	Sinal de controle identificando BSI.
77	pWR(M)	L	Sinal de controle indicando a presença de dado válido sobre o barramento de dado.
78	pDB1N(M)	H	Sinal de controle que requisita dados.
79	A0(M)	H	Bit 0 de endereço.
80	A1(M)	H	Bit 1 de endereço.
81	A2(M)	H	Bit 2 de endereço.
82	A6(M)	H	Bit 6 de endereço.
83	A7(M)	H	Bit 7 de endereço.
84	A8(M)	H	Bit 8 de endereço.
85	A13(M)	H	Bit 13 de endereço.
86	A14(M)	H	Bit 14 de endereço.
87	A11(M)	H	Bit 11 de endereço.
88	DO2(M)/DATA 2(M/S)	H	Bit 2 do dado de saída/bit 2 do dado bidirecional.
89	DO3(M)/DATA3(M/S)	H	Bit 3 do dado de saída/bit 3 do dado bidirecional.
90	DO7(M)/DATA7(M/S)	H	Bit 7 do dado de saída/bit 7 do dado bidirecional.
91	DI4(S)/DATA 12(M/S)	H	Bit 4 do dado de entrada/bit 12 do dado bidirecional.
92	DI5(S)/DATA 13 (M/S)	H	Bit 5 do dado de entrada/bit 13 do dado bidirecional.
93	DI6(S)/DATA 14 (M/S)	H	Bit 6 do dado de entrada/bit 14 do dado bidirecional.
94	DI1(S)/DATA 9(M/S)	H	Bit 1 do dado de entrada/bit 9 do dado bidirecional.
95	DI0(S)/DATA 8(M/S)	H	Bit 0 do dado de entrada/bit 8 do dado bidirecional.
96	sINTA(M)	H	Sinal de status em resposta ao sinal INT.
97	sWO(M)	L	Sinal de status indicando que o presente ciclo é uma escrita na memória ou E/S.
98	ERROR(S)	L (OC)	Sinal de status significando erro durante o ciclo presente.
99	POC(B)	L	POWER ON CLEAR
100	GND		

nais a saber:-

- . 16 linhas de dados
- . 16/24 linhas de endereços
- . 8 linhas de estados
- . 5 linhas de controle de saída
- . 6 linhas de controle de entrada
- . 8 linhas para controle de DMA
- . 8 linhas para o vetor de interrupção
- . 20 linhas utilitárias

A descrição dos sinais do S-100 é fornecida na tabela B.1

#### B.5 - Barramento Série EIA-RS-232, RS-449

O padrão RS-232 surgiu quando a EIA - "*Electronic Industries Association*" - resolveu padronizar os níveis de tensão e impedância para a transmissão de dados digitais em série.

O RS-232 é definida como uma comunicação entre um modem-modulador/demodulador - e algum outro dispositivo.

Existem dois tipos de RS-232:-

- . padrão RS-232 B
- . padrão RS-232 C

B.5.1 - PADRÃO RS-232 B - é o mais antigo e foi desenvolvido para equipamentos que apresentavam variações mais amplas de sinal, em relação as máquinas atuais.

A faixa permissível, para "1" lógico neste padrão cobre de -5 volts a -25 volts, enquanto que o "0" lógico vai de +5 volts a +25 volts.

B.5.2 - PADRÃO RS-232 C - este padrão apresenta as faixas mais restritas, "1" lógico vai de -5 a -15 volts e de +5 a +15 volts para o "0" lógico.

A figura B.4 representa a diferença principal entre estes padrões

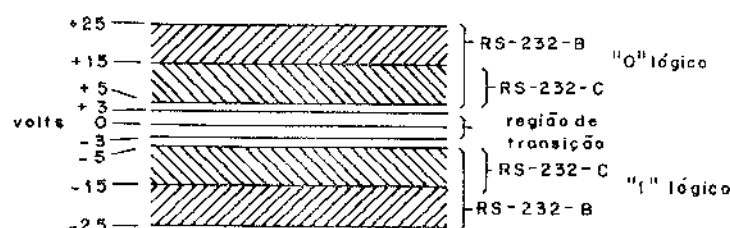


Figura B.4 - RS-232 B, RS-232 C

Nos receptores RS-232, qualquer sinal localizado dentro da faixa reconhecida como um nível lógico válido. Não é necessário que os níveis lógicos "1" e "0" sejam simétricos em relação ao zero, desde que caiam nas faixas especificadas. É interessante notar que os níveis lógicos têm polaridade invertida.

Na tabela B.2 estão especificadas as características elétricas do padrão RS-232 C, que é o mais utilizado atualmente.

Níveis de saída do excitador c/carga de 3 a 7 kΩ	"0" lógico: +5 a +15 V "1" lógico: -5 a -15 V
Tensão de saída do excitador sem carga	- 25 a +25 V
Impedância de saída do excitador sem alimentação	maior que 300 Ω
Corrente de curto na saída	menos de 0,5 A
Slew Rate do excitador	menos de 30 V/μs
Impedância de entrada do receptor	entre 3 e 7 kΩ
Faixa permissível de tensões na entrada do receptor	de -25 a +25 V
Saída do receptor com entrada em aberto	"1" lógico
Saída do receptor com 300 Ω na entrada	"1" lógico
Saída do receptor com entrada de +3 V	"0" lógico
Saída do receptor com entrada de -3 V	"1" lógico

Tabela B.2 - Especificações Elétricas para o RS-232C

A limitação da máxima variação do sinal RS-232 - "*slew rate*" - está relacionada com o problema de interferência entre os condutores de um mesmo cabo - "*cross talk*", quanto mais rápida a transição, maior será o acoplamento entre dois condutores próximos. Em muitos casos isto não representa um problema, pelo fato das extensões de cabo serem reduzidas.



B.5.3 - CONECTOR PADRÃO RS-232 - os dispositivos que empregam a interface RS-232 também utilizam o conector padrão de 25 terminais. O padrão no caso é para a interligação entre o modem e algum outro dispositivo, tal como um terminal de vídeo, ou um computador. Descreveremos a seguir as linhas interconectadas ao conector padrão de 25 terminais:-

- pino 1 - AA - é uma linha protetora de terra, interligando os chassis de todos os dispositivos do sistema.
- pino 2 - BA - é responsável pelos dados transmitidos do terminal para o modem.
- pino 3 - BB - é usado para o envio das informações ao terminal de dados, pelo modem.
- pino 4 - CA - requisição de envio.
- pino 5 - CB - permissão de envio.
- pino 6 - CC - representa o indicador de dados prontos.
- pino 7 - AB - terra do sinal.
- pino 8 - CF - detector de portadora - indica quando o modem recebe um sinal de portadora do outro modem na outra extremidade da linha.
- pino 9 - não definido.
- pino 10- não definido.
- pino 11- não definido.
- pino 12- não definido.
- pino 13- não definido.
- pino 14- não definido.
- pino 15- DB - relógio do bit transmitido; interno.
- pino 16- não definido.
- pino 17- DD - relógio do bit recebido.
- pino 18- não definido.
- pino 19- não definido.
- pino 20- CD - terminal de dados pronto.
- pino 21- não definido.
- pino 22- CE - indicador de linha telefônica; indica quando o modem recebe um sinal da linha telefônica.
- pino 23- não definido.
- pino 24- DA - relógio do bit transmitido; externo.
- pino 25- não definido.

Os terminais 1 e 7 costumam ser utilizados em comum.

Os terminais 15, 17 e 24 são úteis aos modem que transmitem a taxas relativamente altas, 1200 a 2400 "baud", para a sincronização

de dados.

De todas essas linhas as mais utilizadas são:-

- pino 1 - terra de proteção.
- pino 2 - dados transmitidos.
- pino 3 - dados recebidos.
- pino 4 - requisição de envio.
- pino 5 - permissão de envio.
- pino 7 - terra do sinal.

B.5.4 - PADRÃO RS-449 - a rápida evolução dos sistemas de teleprocessamento, com a introdução de interfaces mais rápidas e a necessidade de sinalização mais sofisticada, tornou necessário o desenvolvimento de um padrão série, para substituir paulatinamente o padrão RS-232.

As diferenças entre as normas RS-232 e RS-449 prendem-se às características elétricas dos sinais e à definição de circuitos de controle adicionais:-

- . para permitir operações balanceadas os sinais de transmissão e recepção de dados passaram a ter os seus próprios sinais de terra; tornando possível definir maiores distâncias entre equipamentos através da redução de interferência entre os circuitos de intercâmbio de dados e elevar as taxas de sinalização a  $2 \times 10^6$  bits por segundo (2 "megabaud").

- . foram definidos dez novos circuitos de sinalização que não fazem parte da norma RS-232.

- . algumas mudanças foram feitas na definição do circuito.

- . foi estabelecido um novo conjunto de interfaces padronizadas para a configuração do sistema de comunicação.

- . criou-se um novo conjunto de nomes e mnemônicos.

- . foi especificado um conector de interface diferente em tamanho, e o sistema de travamento deste conector.

Desta maneira torna-se possível tirar proveito dos recentes avanços na tecnologia de circuitos integrados e acomodar as elevadas taxas de transmissão necessárias na comunicação entre unidades em redes de processamento distribuído.

Damos a seguir uma breve explicação de cada sinal.

- .. SG - Terra do Sinal - "Signal Ground" - conecta diretamente o terra dos equipamentos para estabelecer um retorno comum dos sinais.

- .. SC - Retorno da Transmissão - "*Send Common*" - é usada como referência de potencial para transmissão.
- .. RC - Retorno da Recepção - "*Receive Common*" - é usada como referência de potencial para recepção.
- .. IS - Terminal Operando - "*Terminal In Service*" - indica quando o equipamento de dado está disponível para operação.
- .. IC - Chamada - "*In Coming Call*" - indica quando um sinal de chamada está sendo recebido pelo equipamento de comunicação.
- .. TR - Terminal Pronto - "*Terminal Ready*" - usado para controlar o chaveamento do equipamento de comunicações de/para o canal de comunicação.
- .. DM - Estado do Equipamento de Comunicação - "*Data Mode*" - indica o estado - "*status*" - do equipamento de comunicação .
- .. SD - Transmite Dados - "*Send Data*".
- .. RD - Recebe Dados - "*Receive Data*".
- .. TT - Temporização do Terminal - "*Terminal Timing*" - fornece ao equipamento de comunicação informações sobre temporização do sinal de transmissão, ou seja, sua transição indica o centro do bit a serem transmitidos.
- .. ST - Temporização da Transmissão - "*Send Timing*" - sua transição indica o centro dos bits transmitido.
- .. RT - Temporização da Recepção - "*Receive Timing*" - sua transição indica o centro dos bits recebidos.
- .. RS - Requisição para Transmitir - "*Request to Send*" - controla a função de transmissão no canal de dados do equipamento de comunicação local e controla o sentido de transmissão de dados no canal "*Half-Duplex*" do equipamento de comunicação local.
- .. CS - Pronto para Transmitir - "*Clear to Send*" - indica quando o equipamento de comunicação está pronto para transmitir dados.
- .. RR - Receptor Pronto - "*Receive Ready*" - indica quando o equipamento de comunicação está pronto para receber dados.
- .. SQ - Qualidade do Sinal - "*Signal Quality*" - indica

quando existe uma alta probabilidade de erro nos sinais recebidos.

- .. NS - Novo Sinal - "*New Signal*" - é usado na estação de controle de sistema multiponto - "*polling*", para desencadear rápidas trocas de mensagens a sucessivas estações remotas.
- .. SF - Seleção de Frequência - "*Select Frequency*" - usado para selecionar faixas de frequências de transmissão e recepção do equipamento de comunicação.
- .. SR - Seletor da Taxa de Sinalização - "*Signaling Rate Selector*" - usado para selecionar uma das duas taxas de sinalização de dados do equipamento de comunicação.
- .. SI - Indicador da Taxa de Sinalização - "*Signaling Rate Indicator*" - usado para indicar uma das duas taxas de sinalização de dados do equipamento de comunicação.
- .. SSD- Transmite Dados Secundários - "*Secondary Send Data*" - é equivalente ao SD, exceto que pode ser usado para transmitir dados via canal secundário.
- .. SRD- Recebe Dados Secundários - "*Secondary Receive Data*" - é equivalente ao RD, exceto que pode ser usado para receber dados via canal secundário.
- .. SRS- Requisição para Transmitir Secundário - "*Secondary Request to Send*" - é equivalente do RS, exceto por ser usado para controlar a função de transmissão no canal de dados secundário.
- .. SCS- Pronto para Transmitir Secundário - "*Secondary Clear to Send*" - é equivalente ao CS, indicando quando o equipamento de comunicação está pronto para transmitir dados no canal secundário.
- .. SRR- Receptor Secundário Pronto - "*Secondary Receive Ready*" - é equivalente ao RR, indicando quando o receptor no canal secundário do equipamento de comunicação está recebendo um sinal apropriado.
- .. LL - Loopback Local - "*Local Loopback*" - usado para controlar o teste de "*loopback*" local.
- .. RL - Loopback Remoto - "*Remote Loopback*" - usado para controlar o teste de "*loopback*" remoto.

- .. TM - Modo Teste - "*Test Mode*" - indica que o equipamento de dado local está sendo testado.
- .. SS - Selecciona Repouso - "*Select Standby*" - indica quando o equipamento de comunicação está operando em modo normal ou repouso.

## ANEXO C



## BIBLIOGRAFIA



- /1/ PEATMAN, J.B. - "Microcomputer Based Design" - McGraw Hill Kogakusha - Tokyo - 1977.
- /2/ LESEA; A. e ZAKS, R. - "Microprocessor Interfacing Techniques" - Sybex, Berkeley 1977.
- /3/ ROSEN, S. - "Electronic Computer - A Historical Survey" - Computing Survey, vol 1, n° 1, march 1969 - pp 7 - 36.
- /4/ NICOUD, J.DANIEL - "Peripheral Interface Standards for Microprocessor" - Proceedings of the IEEE, vol 64, n° 6, june 1976 - pp-896 - 904.
- /5/ LOUGHRY, DONALD C. e ALLEN, MARKS - "IEEE Standard 488 and Microprocessor Synergism" - Proceedings of the IEEE, vol 66, n° 2, february 1978 - pp 162 - 172.
- /6/ MOFFA, ROY - "Interfacing Peripheral in Mixed Systems" - Computer Design - april 1975 - pp 77 - 84.
- /7/ ANDERSON, GEORG A. e JENSEN, E.DOUGLAS - "Computer Interconnection Structures: Taxonomy Characteristics and Examples" - Computing Survey, vol 7, n° 4, december 1975 - pp 197 - 213.
- /8/ ADAMS, G.E. e RALANDER, THOMAS - "Design Motivations for Multipleprocessor Microcomputer Systems" - Computer Design - march 1978 , pp 81 - 89.
- /9/ MORGAN, D.E. , TAYLOR, D.J. e CUSTEAU, G. - "A Survey of Methods for Improving" - Computer Network Reliability and Availability Computer - november 1977, pp 42-50.
- /10/ LOUGHRY, D.D. - "What makes a good Interface ?" - IEEE Spectrum november 1974 - pp 52 - 57.
- /11/ KNOBLOCK, D.E.; LOUGHRY, D.C. e VISSERS, C.A. - "Insight into Interfacing" - IEEE Spectrum - may 1975 - pp 50 - 57.

- /12/ MOSTEK 1979 MICROCOMPUTER DATA BOOK - Mostek 1979.
- /13/ ZUFFO, J.A. "Fundamentos da Arquitetura e Organização dos Microprocessador" - Editora Edgard Blucher Ltda.
- /14/ MCS-86 USER'S MANUAL - Intel february 1979.
- /15/ KLINGMAN, E.E. - "Microprocessor Systems Design" - Prentice-Hall Inc - 1977.
- /16/ INTEL MULTIBUS INTERFACING - Intel january 1979 - AP-28A
- /17/ ELMQUIST, K.A.; FULMERH.; GUSTAVSON, D.B. e MORROW, GEORGE - "Standard Specification for S-100 Bus Interface Devices" Computer - july 1979 - pp 28 - 52.