

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
SETEMBRO DE 1979

UM SISTEMA OPERACIONAL PARA
REDE DE MICRO-PROCESSADORES

por : Manoel da Nóbrega
orientador : Prof. Dr. Márcio L. de Andrade Netto

Tese apresentada à Faculdade de Engenharia
FEC-UNICAMP como parte dos requisitos exi-
gidos para obtenção do título de MESTRE EM
ENGENHARIA

AGRADECIMENTOS

A todos aqueles que direta ou indiretamente contribuíram na realização deste trabalho.

Este trabalho contou com o apoio financeiro das entidades:

CONSELHO NACIONAL DE DESENVOLVIMENTO CIENTÍFICO E TECNOLÓGICO - CNPq
COMPANHIA DO METROPOLITANO DE SÃO PAULO - METRÔ

RESUMO

Descreve-se neste trabalho, que é uma contribuição em termos de estudos realizados e programas desenvolvidos, um Sistema Operacional para uma Rede de Microcomputadores.

Tal projeto tem por finalidade utilizações didática e de pesquisa no Laboratório de Micro-computadores da Faculdade de Engenharia Elétrica da UNICAMP, podendo ser expandido e reestruturado de acordo com as necessidades.

Detalham-se a parte fixa do Sistema Operacional desenhado, assim como exemplificam-se os procedimentos necessários a uma outra configuração.

INDICE

INTRODUÇÃO	1
SEÇÃO 1- HISTÓRICO E CONSIDERAÇÕES GERAIS	3
1.1. - Histórico de Sistemas Operacionais e computadores eletrônicos	3
1.2. - Histórico de Redes de Computadores	5
1.3. - Estruturas de Interconexão de Processadores.....	6
SEÇÃO 2- RECURSOS COMPUTACIONAIS. REQUISITOS NECESSÁRIOS	14
2.1. - Apresentação dos recursos	14
2.1.1. - Interdata 80	14
2.1.2. - Signetics 2650	17
2.1.3. - Outros dispositivos	18
2.2. - Estrutura escolhida. Especificação	18
2.2.1. - Requisitos necessários	22
SEÇÃO 3- UM SISTEMA OPERACIONAL PARA REDE DE MICRO-PROCESSADORES	25
3.1. - Detalhes de implementação	25
3.1.1. - Interdata 80	26
3.1.2. - Signetics 2650	43
3.2. - Comentários sobre alocação de recursos dos processadores	49
3.3. - Considerações gerais sobre operação do Sistema	50

SEÇÃO 4- CONCLUSÕES	52
BIBLIOGRAFIA	53

INTRODUÇÃO

O advento dos computadores, a vasta gama de problemas e projetos de pesquisa resultante deste grande avanço tecnológico, permitiu, e ainda hoje propicia ao homem inúmeras reflexões e análises [7] sobre possíveis áreas de aplicações e de utilização de tão potentes ferramentas em seu trabalho. Assim sendo: "Computadores", "Inteligência Artificial", "Processamento Distribuído", "Controle de Processos", "Sistemas Operacionais", entre outras são atualmente algumas das áreas de conhecimento em grande desenvolvimento.

O trabalho ora apresentado, é uma contribuição ao estudo de programação visando a implementação de uma Rede de Microcomputadores na Faculdade de Engenharia - Laboratório de Microcomputadores. Tal pesquisa concorre para a assimilação de conhecimentos, na utilização de recursos disponíveis em projetos na área de "Processamento Distribuído" e "Controle de Processos".

Na seção 1 do trabalho elabora-se um histórico de Sistemas Operacionais e de Computadores Eletrônicos, analisando-se também estruturas de conexão de processadores, assim como as características e potencialidades de tais Redes.

Na seção 2 faz-se a análise de nosso próprio "meio ambiente", apresenta-se os recursos disponíveis e estabelece-se as metas principais a serem atingidas, delineando a estrutura que cumpre tais objetivos.

Na seção 3 discute-se a estrutura em termos de módulos escritos e modificações procedidas para a concretização desse projeto. Exemplificam-se os detalhes de implementação de tal trabalho, apresentando-se algumas das sutilezas existentes em termos de Sistemas Operacionais para Redes de Microcomputadores.

Na seção 4 tecem-se comentários, considerações e sugestões sobre possíveis melhoramentos futuros, e discute-se a utilização de outras estratégias na abordagem do problema em questão.

Finalmente, é conveniente esclarecer que se optou pela inserção, em cada capítulo, de sua respectiva bibliografia, objetivando-se maior

modularidade e facilidade de consultas.

BIBLIOGRAFIA

- | 7 | Nievergelt, J. and Farrar, J.C.
What Machines Can and Cannot Do.
Computing Surveys, vol.4, nº 2. June 1972, pp.81-96.

SEÇÃO 1 - HISTÓRICO - CONSIDERAÇÕES GERAIS

1.1. Histórico de Sistemas Operacionais e Computadores Eletrônicos.

Anteriormente a 1956 realizava-se o processamento "job" a "job" no qual o usuário (programador) em contato direto com a máquina (painéis e chaves) depurava seu programa e preparava os cartões e "tapes" a ser usados, iniciando e monitorando a execução de sua tarefa.

No período 1956-1958, como decorrência dos esforços empreendidos para a criação do FORTRAN (1957), surgiram os primeiros sistemas "batch" que consistiam de um pequeno monitor para sequenciar as operações de cada "job" e chamar os vários componentes do Sistema: assemblers, compiladores, carregadores, biblioteca de rotinas, etc... Disso resultou: padronização de rotinas de Entrada/Saída, aprimoramentos no carregador e na biblioteca de rotinas, no encaimento das tarefas e também melhorias na documentação e relatórios.

De 1959-1961 do desenvolvimento de "hardware" (canais, interrupção) resultou uma grande mudança nos processos de entrada/saída e uso de processador, sendo criados Sistemas Operacionais Modulares consistindo de rotinas imprescindíveis à operação da máquina (núcleo do Sistema-Executivo) e outros acessórios, que acionados por uma linguagem de comando permitiam ao usuário a utilização de toda uma série de compiladores, "debugs" e recursos que facilitavam a execução de seus programas.

No período 1962-1964 já se tinha dado um grande passo na construção dos Sistemas Operacionais, trabalhava-se então em inovações e aperfeiçoamentos como a proteção de áreas por "hardware", o sistema de arquivo em disco, a multiprogramação visando otimização no uso da memória, o "spooling", o "swapping", técnicas essas aperfeiçoadas que culminaram na criação de Sistemas de Tempo Compartilhado "(Time Sharing Systems)", com a utilização de um mesmo processador por inúmeros usuários.

Novas e criativas soluções (1966- tempos atuais) foram propostas, gerando-se Sistemas de Tempo Real, Sistemas de Controle Distribuído que hoje são estudados, aprimorados e aplicados [1.1] nas interconexões de processado

res tornando-se os Sistemas Operacionais para Redes de Processadores.

Ressalta-se que os intervalos de tempos foram fixados com aproximações e que, atualmente, as pesquisas em "Sistemas Operacionais não Convencionais" para estruturas de processadores em hierarquias "piramidais" reconfiguráveis, e outras, acham-se em andamento, resultando novas definições e conceitos a serem assimilados.

Cabe reafirmar uma das características principais de Sistemas Operacionais, qual seja a de formarem "um conjunto de procedimentos automáticos e manuais que interagem proporcionando ao usuário a melhor utilização dos recursos disponíveis em um Sistema de Computação".

Paralelamente e também às vezes na dianteira do aperfeiçoamento do "software de Controle Operacional", o "hardware" contribuiu, e ainda hoje auxilia grandemente este esforço de otimização do uso dos recursos e dispositivos.

Assim é que, a partir de uma primeira geração onde válvulas eletrônicas eram universalmente usadas na implementação da lógica computacional como seus componentes ativos, passamos para uma segunda geração na qual o transistor veio modificar e simplificar os problemas existentes na criação e execução desses computadores eletrônicos.

Contribuíram identicamente para o aperfeiçoamento dos processadores os sucessos que foram sendo conquistados através da evolução dos equipamentos periféricos (leitoras de cartões, fitas, discos e tambores magnéticos, cassetes, terminais, etc.)

Houve muitas discussões para que se estabelecesse uma distinção entre segunda e terceira geração; aqueles que ainda usavam componentes discretos em suas máquinas reclamavam a transição em termos de melhor performance do sistema; outros que tinham baseado seus novos produtos em circuitos monolíticos integrados pretendiam tal diferenciação sobre esse fator de maior integração obtido nos componentes.

O observado e que está progressivamente se repetindo, é que a maior integração dos componentes é a responsável direta pelo avanço na performance dos dispositivos, assim é que dispositivos L.S.I. e V.L.S.I. constituem agora a quarta geração de produtos: microcomputadores, memórias, etc. Para o desenvolvimento do "hardware" não houve preocupação com a cronologia pois a mesma é re

lativamente coincidente com aquela dos Sistemas Operacionais.

Atualmente e em futuro breve acredita-se que os progressos advirão de uma interrelação entre esses dois campos com perspectivas de maiores contribuições por parte do "software".

Um histórico mais completo pode ser visto em [8] e [9].

1.2 - Considera-se agora o tópico referente a Redes de Processadores e suas características fazendo-se um breve histórico do desenvolvimento das mesmas.

Pode-se dizer que a primeira rede de computadores constituía-se de um processador com inúmeros cabos de pequeno comprimento conectados a dispositivos periféricos. Naturalmente tal configuração evoluiu para a adição de terminais remotos anexados aos Sistema, possibilitando assim uma comunicação à maior distância do Processador Central.

Com o crescimento das redes ocorreu incremento dos custos, particularmente daqueles referentes a comunicações, decorrendo de tal situação o surgimento de linhas "multidrop" que permitem a um certo número de diferentes terminais, o compartilhamento de uma mesma linha comum.

A próxima dificuldade e conseqüente aperfeiçoamento referia-se ao problema de modificações que precisavam ser realizadas na estrutura do "software", acertos estes excepcionalmente difíceis e custosos.

Desta forma para reduzir o tempo e as despesas com o aprimoramento do Sistema, foram criados dispositivos chamados "front-end" os quais se paravam as funções de comunicação das de processamento.

Tais elementos evoluíram muito rapidamente com início na década de 60, e foram acompanhados pela introdução dos minicomputadores de baixo custo.

Hoje, "front end processors" tornaram-se uma parte importante na comunicação de uma rede.

Outro aperfeiçoamento natural referiu-se à anexação junto ao computador central de um dispositivo "front end" que completava a isolação dos elementos da rede, possibilitando maior flexibilidade na readaptação dos dispo

sitiyos a ele conectados.

Paralelamente a esse desenvolvimento de redes orientadas para terminais, pode-se assinalar a evolução de redes com ligações computador-computador. Inicialmente surgidas por conexões simples entre os processadores, e modificando-se com a introdução de processadores "front end" também chamados "network interface processor", evoluíram para estruturas mais complexas, estudadas a seguir.

Ressalta-se igualmente a evolução decorrente de pesquisas e resultados obtidos em estudos de teoria de filas, simulações de modelos de estruturas e surgimento de novas e melhores técnicas de comunicação dinâmica "(packet switching)", etc.

Recomenda-se o estudo de [2] e [4] para aprofundamento no tema.

1.3 - Quando se visa um estudo acerca de Conexões Interprocessadores, deve-se inicialmente detalhar as características desejáveis que devem estar presentes tais como:

- Modularidade - que é a habilidade de se realizar mudanças na estrutura visando um incremento da capacidade do Sistema, com mínima implicação nos custos "(cost modularity)" e com flexibilidade quanto à localização ("place modularity") dos novos dispositivos.

- Falhas nos dispositivos - que é a projeção estudada acerca dos efeitos ("failure effect") causados 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 de "software" (estratégias de caminhos das mensagens) e "hardware" (alterações das conexões - eliminação de processadores que apresentam problemas, etc...). Objetiva-se com tais precauções a construção de um sistema tolerante a falhas ("fault tolerant").

- Complexidade lógica - diz respeito à lógica de Interconexão usada e necessária a cada interface dos elementos da rede

- "Gargalos (bottlenecks)" - onde são considerados os elementos que podem vir a degradar o desempenho do Sistema como um todo.

A respeito das decisões que se precisa fazer quanto à criação de um esquema de classificação das estruturas ressalta-se:

- estratégia de transferência de mensagens:
 - direta, quando o elemento interventor no fluxo da informa
ção, se existir, não modifica o conteúdo da mensagem;
 - indireta, se o elemento retransmissor da comunicação executa
alterações na mesma (transformações de endereços, etc.)
- métodos de controle da transferência:
 - centralizado, no qual um mesmo elemento da rede despacha to
das as mensagens;
 - descentralizado; as comunicações são processadas pela utili
zação de diferentes elementos interventores no diálogo.
- escolha de tipo de caminho sobre o qual a transferência é feita:
 - dedicado, caminhos unidirecionais ou bidirecionais ponto a
ponto;
 - compartilhado, caminhos que são bidirecionais e atingem
mais do que dois pontos.

Na árvore formada pelas interconexões para comunicação de pro
cessadores, temos como folhas os nomes específicos dos sistemas classificados
quanto à sua arquitetura. Vide figura 1.

Descreve-se e analisa-se a seguir dez diferentes maneiras de
se realizar a ligação entre processadores, baseando-se na manipulação de men
sagens entre os dispositivos e na topologia do "hardware" de interconexão [1].

- 1- "D.D.L. - Direct Dedicated Loop" - estrutura de conexão que apre
senta características muito boas quanto à modularidade (custo e
"locais de expansão") por permitir a inserção de um elemento pro
cessador em qualquer lugar da rede. É pobre quanto à falha nos
dispositivos e reconfiguração da rede, devido às implementações
serem geralmente com caminhos unidirecionais, podendo melhorar
pela adoção de duplicação nas ligações e redundância de processa
dores. Sua complexidade lógica nas ligações é baixa, e um garga
lo em potencial pode tornar-se a largura da faixa de transmissão
das comunicações ("bandwith") pelo aumento das taxas de comunica
ção. Em geral tais sistemas são projetados para aplicações onde
a confiabilidade e fatores de performance não são demasiados exi
gentes.

- 2 - "D.D.C. - Direct Dedicated Complete" - estrutura regular de conexão na qual um elemento processador encontra-se simetricamente ligado a todos os outros, apresentando por isso uma "cost modularity" pobre e uma "place modularity" boa. Sua flexibilidade nas conexões é alta e possui também boas vantagens quanto aos efeitos de falha e possibilidades de reconfiguração. Não apresenta um gargalo óbvio e sua complexidade de ligação é relativamente baixa.
- Interesses bastante acadêmicos existem acerca de tal configuração, podendo-se já encontrar implementações práticas bem sucedidas de tal arquitetura (MERIT - Michigan University).
- 3 - "D.S.M. Direct Shared Memory" - estrutura de conexão por compartilhamento direto de memória que tem uma característica muito boa quanto à modularidade nos locais possíveis de expansão. Os custos de expansão dependendo da estrutura pela qual os processadores têm acesso ao sistema de memória pode ser pobre se, para cada processador, temos um caminho direto, e muito bom se tivermos o compartilhamento através de um bus.
- Apresenta uma qualificação boa a respeito de falhas (efeito no sistema e reconfiguração do mesmo) em relação aos processadores e pobre em termos de memória, com um possível gargalo na "largura da faixa de transmissão" da memória.
- 4 - "D.S.B. Direct Shared Bus" - arquitetura na qual os elementos processadores compartilham um "bus" comum na realização de suas comunicações, por isso apresenta pobres características quanto ao caminho das mesmas, visando-se modularidade, e uma boa qualificação em termos de adição de novos processadores.
- Características idênticas apresentam-se em relação à falha nos dispositivos e reconfiguração da rede. Um possível gargalo pode resultar no uso do "bus" sendo a sua complexidade lógica baixa.
- Largamente utilizada em pesquisas aeroespaciais onde ambos caminhos série e paralelos são usados.
- 5 - "I.C.D.S. - Indirect Centralized Dedicated Star" - Consiste de um dispositivo de comutação central ao qual cada processador está conectado por um caminho bidirecional. Apresenta boas vantagens em termos dos processadores com relação à modularidade

("cost and place") e falhas; tendo pobres características quanto ao comutador central nos mesmos aspectos. Um possível gargalo pode ser o recurso central e a complexidade na lógica das ligações é moderada.

- 6 - "I.C.D.L. - Indirect Centralized Dedicated Loop" - Consiste da implementação de conexões diretas inter dispositivos em uma estrutura I.C.D.S. Nesta configuração as mensagens são colocadas no loop pelos remetentes e removidas para uma operação de mapeamento de endereços por um dispositivo central de comutação sendo então recolocada no "loop" para seu destinatário. Sendo combinação de D.D.L. e I.C.D.S. tem as mesmas características do D.D.L. quanto as falhas em relação ao caminho de dados. Quanto ao comutador central ainda tratando-se de suas possíveis falhas apresenta características idênticas ao I.C.D.S. - (pobres).

Usuários maliciosos constituem-se em problemas potenciais podendo degradar a performance do sistema, e sua complexidade na lógica das conexões é moderada.

- 7 - "I.C.S. Bus with Central Switch" - A retransmissão de mensagens pelo processador central é a característica diferenciadora principal desta configuração em relação às do tipo D.S.B. Apresenta boas vantagens nos custos e nas localizações dos novos elementos, processadores nas expansões e características pobres com relação ao dispositivo central. Idênticamente em relação às falhas de elementos da rede, no sentido de uma reconfiguração, e com características pobres quanto ao efeito causado por falhas no processador central. Sua complexidade lógica é baixa e um possível gargalo pode ser o dispositivo central.

- 8 - "I.D.D.R. - Indirect Decentralized Dedicated Regular" - estrutura espacial regular de conexão que tem características de modularidade extremamente pobres advindas da rigidez da arquitetura - absoluta regularidade.

Quanto à reconfiguração por falhas é extremamente pobre, apresentando todavia uma possibilidade de superar os efeitos decorrentes do mau funcionamento pela implementação de diferentes caminhos de mensagens caracterizando assim a recuperação perante a falha de moderada a boa.

Para esta estrutura é difícil o reconhecimento de gargalos do sistema. Desperta especial interesse acadêmico.

- 9 - "I.D.D.I. Irregular Networks - Indirect Decentralized Dedicated Irregular" - Unicamente diferenciável do I.D.D.R. pela não rigidez (não regularidade) das conexões.
 Considerações de modularidade: "place", extremamente boas e "custos", boas.
 Os efeitos referentes à falhas são proporcionais e dependentes da estrutura.
 O gargalo não é um problema em potencial a menos de projetos específicos e usualmente a complexidade das ligações tende a ser extremamente alta.
- 10 - "I.D.S. - Bus Window - Indirect Decentralized Shared Bus" - Comutação de mensagens realizada por mais de um dispositivo, sendo a comunicação transmitida pelo compartilhamento do "bus". Quanto à modularidade suas características tendem a ser extremamente boas, com características deficientes quanto a falhas, devido a uma simples ligação poder afetar quase todo o sistema. Não configura facilmente, a menos de estruturas específicas, um gargalo e sua complexidade lógica é usualmente muito alta.

Segue-se um quadro resumo dessas estruturas de conexões propostas. Vide tabela 1.

Caberia ainda ressaltar que inúmeros trabalhos em estrutura de processadores encontram-se implementados na prática e podem ser mais profundamente pesquisados em [1].

A estrutura que mais se enquadra àquela por nós escolhida segue melhor detalhada na seção 2 deste trabalho.

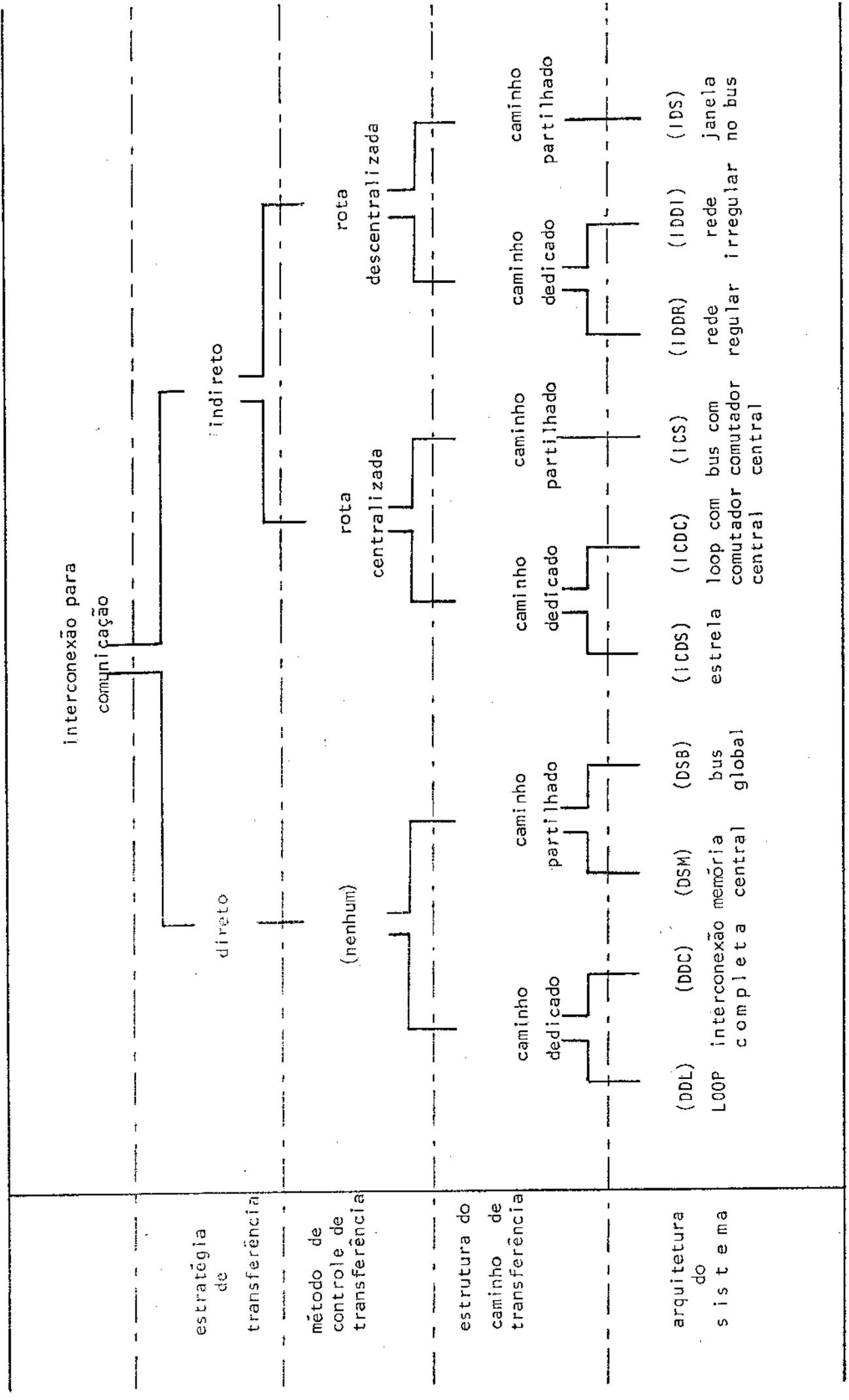
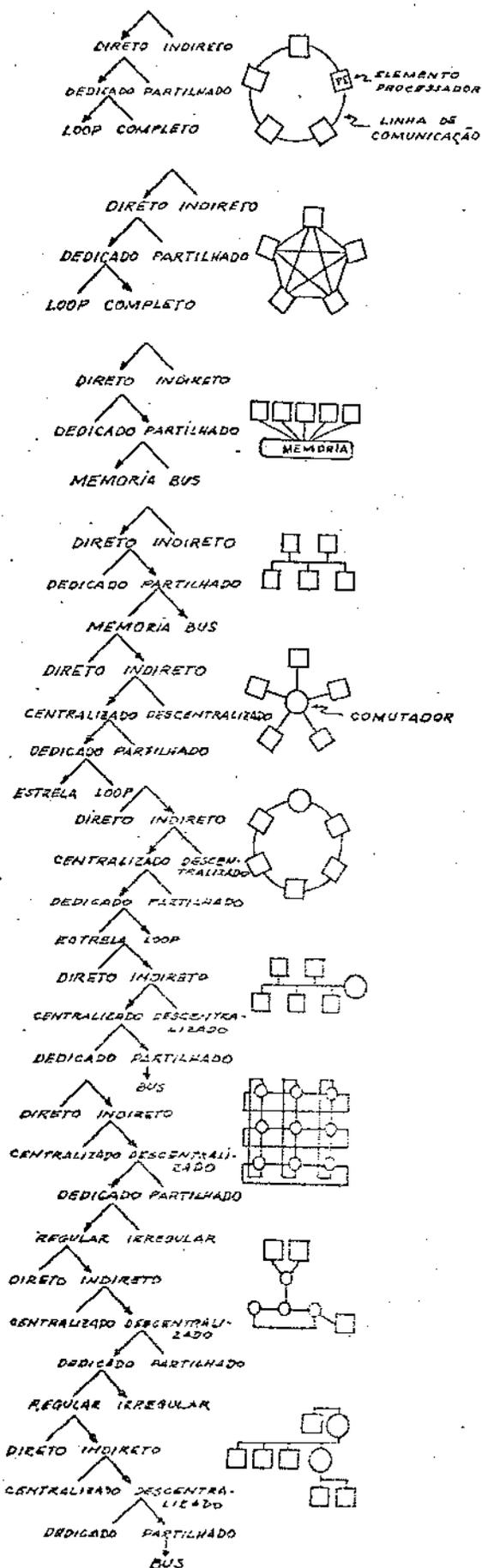


Fig. 1 - Árvore das estruturas de conexão de processadores

Tabela 1. Análise das estruturas de interconexão de processadores



Modularidade		falhas		Complexidade Lógica	Possíveis Gargalos
Custo	Localização	Efeito	Reconfiguração		
muito bom	muito bom	pobre	pobre	baixa	largura da faixa de transmissão
pobre	bom	bom	bom	relativamente baixa	não são óbvios
depende da estrutura	muito bom	bom para os PEs pobre para a memória	bom para os PEs pobre para a memória	baixa	largura da faixa de transmissão da memória
bom para os PEs pobre para a linha de comunicação	bom para os PEs pobre para a linha de comunicação	bom para os PEs pobre para o bus	bom para os PEs pobre para o bus	baixa	bus
bom para os PEs pobre para o comutador	bom para os PEs pobre para o comutador	bom para os PEs pobre para o comutador	bom para os PEs pobre para o comutador	moderada	comutador
bom para os PEs pobre para o comutador	bom para os PEs pobre para o comutador	pobre para o comutador pobre para as linhas de dados	pobre para o comutador pobre para as linhas de dados	moderada	comutador e largura de faixa de transmissão
bom para os PEs pobre para o comutador	bom para os PEs pobre para o comutador	pobre	bom para os PEs pobre para o comutador	moderada	comutador e bus
extremamente pobre	extremamente pobre	de moderado a bom	excessivamente pobre	moderada	não são significativos
boa	tende a ser extremamente boa	proporcional à estrutura	proporcional à estrutura	usualmente muito alta	não são significativos
tende a ser extremamente boa	tende a ser extremamente boa	dependendo da estrutura pode ser pobre	dependendo da estrutura pode ser pobre	usualmente muito alta	não são significativos

BIBLIOGRAFIA

- [1] | Anderson, George A., Jensen, E. Douglas.
Computer Interconnection Structures: Taxonomy, Characteristics and Examples.
Computing Surveys, vol. 7 Nº 4. December 1975, pp. 197-213.
- [2] | Coffman Jr., Edward G. & Denning, Peter J.,
Operating Systems Theory.
Prentice Hall, 1973.
- [3] | Frank, H.
Computer Networks: Art to Science to Art .
Networks, vol. 5, pp.7-32.
- [4] | Hoare, C.A. & Perrot, R.H.
Operating Systems Techniques.
Academic Press, New York, 1972.
- [6] | Kimbleton, Stephen R., Schneider, G. Michel.
Computer Communication Networks: Approaches, Objectives and Performance Considerations.
Computing Surveys, vol. 7 nº 3 December 1975, pp. 129-173.
- [8] | Rosen, S.
Electronic Computers. A Historical Survey.
Computing Surveys, vol. nº 1. March 1969, pp. 7-36.
- [9] | Rosin, Robert F.
Supervisory and Monitor Systems.
Computing Surveys. vol. nº 1. March 1969, pp. 37-54.
- [11] | Suguy, Takao. & Nóbrega, Manoel da. & Netto, Márcio L.A.
Controle Distribuído e Descentralizado de Processos Industriais por Computadores - Parte 4.
XII Congresso Nacional de Processamento de Dados. SUCESU. Outubro 1979.

SEÇÃO 2 - RECURSOS COMPUTACIONAIS E REQUISITOS NECESSÁRIOS

2.1. Apresentação dos recursos

Objetivou-se levantar, e detalhar os requisitos necessários e associar os recursos existentes para a implementação de uma Rede de Micro-computadores, num projeto conjunto "Hardware/Software", desenvolvendo-se em paralelo o Sistema Operacional adequado para suportar a estrutura pretendida.

No laboratório de Micro-computadores da Faculdade de Engenharia da UNICAMP dispõe-se de um minicomputador Interdata 80 e de terminais remotos ligados ao DEC System-10, computador central do Campus Universitário; o computador terminal gráfico inteligente GT-40 da DEC usado para o desenvolvimento de aplicações gráficas em sistemas interativos de Banco de Dados e os micro-computadores: Intel 8080, Signetics 2650 (2 unidades), Zilog 80 e o Intel 8086.

Detalha-se a seguir alguns dos dispositivos mais diretamente envolvidos no projeto lembrando que o mesmo foi proposto em termos de portabilidade, modularidade e expansibilidade.

2.1.1. Interdata 80 [5]

É uma máquina de arquitetura e formato de instruções muito similar à série de computadores IBM 360-370. Tal estrutura associada a uma série de instruções que aumentam a eficiência do uso da memória e dos recursos tornam-na de fácil operação por qualquer usuário. Devido à extensa gama de "software de suporte" e de apoio existente, permite, ao mesmo tempo, o desenvolvimento de inúmeras outras aplicações, através da criação e teste de módulos de controle de outros dispositivos conectados pela utilização de interfaces de uso geral e de outras facilidades providas pelo sistema.

As figuras 2 e 3 apresentam um diagrama de blocos do Sistema e do processador.

A grande variedade de recursos e de ferramentas auxiliares para o desenvolvimento do trabalho e a possibilidade de utilização a curto prazo de parte desse material foram fatores adicionais que influenciaram na escolha desse processador central.

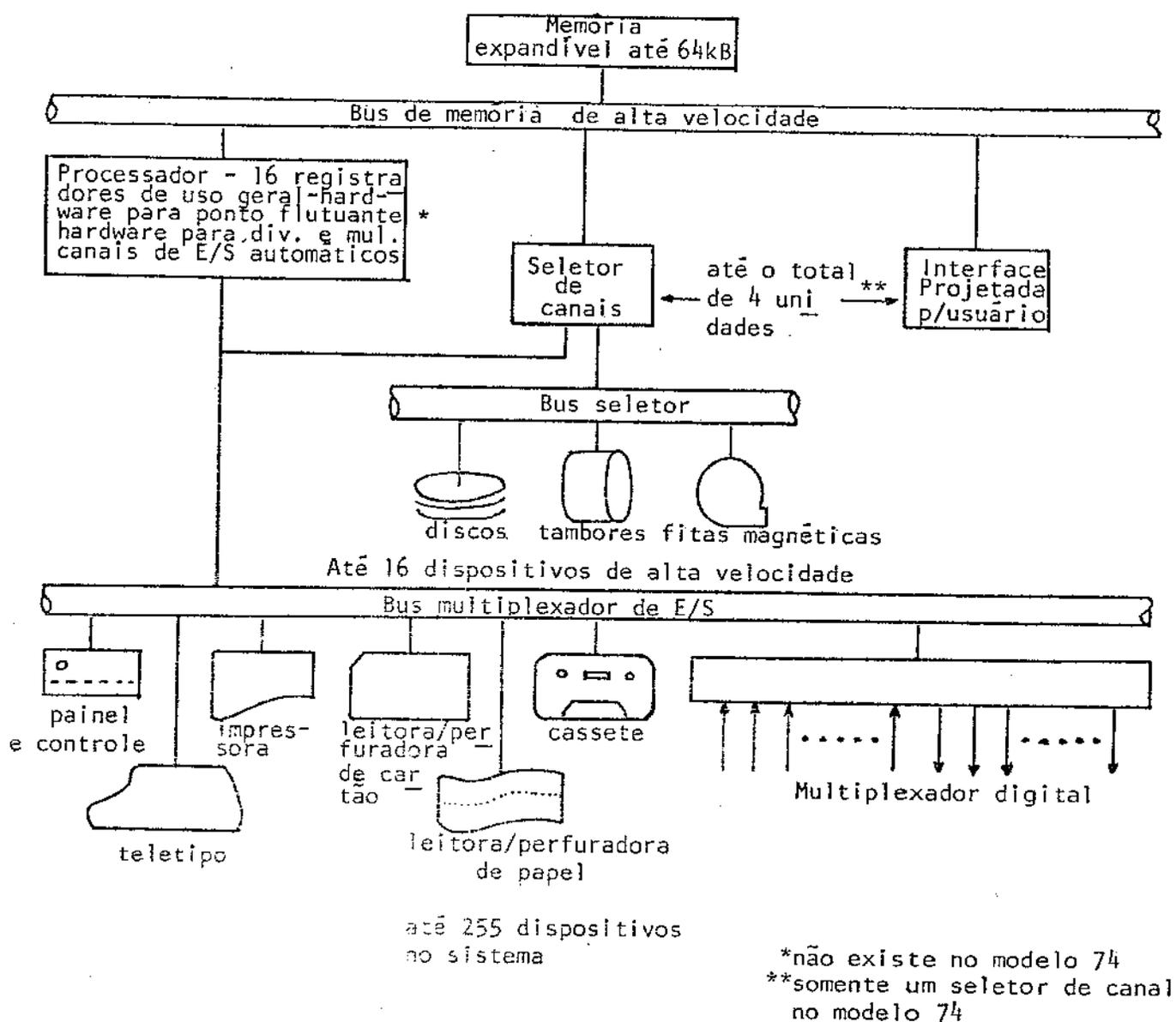


Fig. 2 - Diagrama de bloco do Sistema

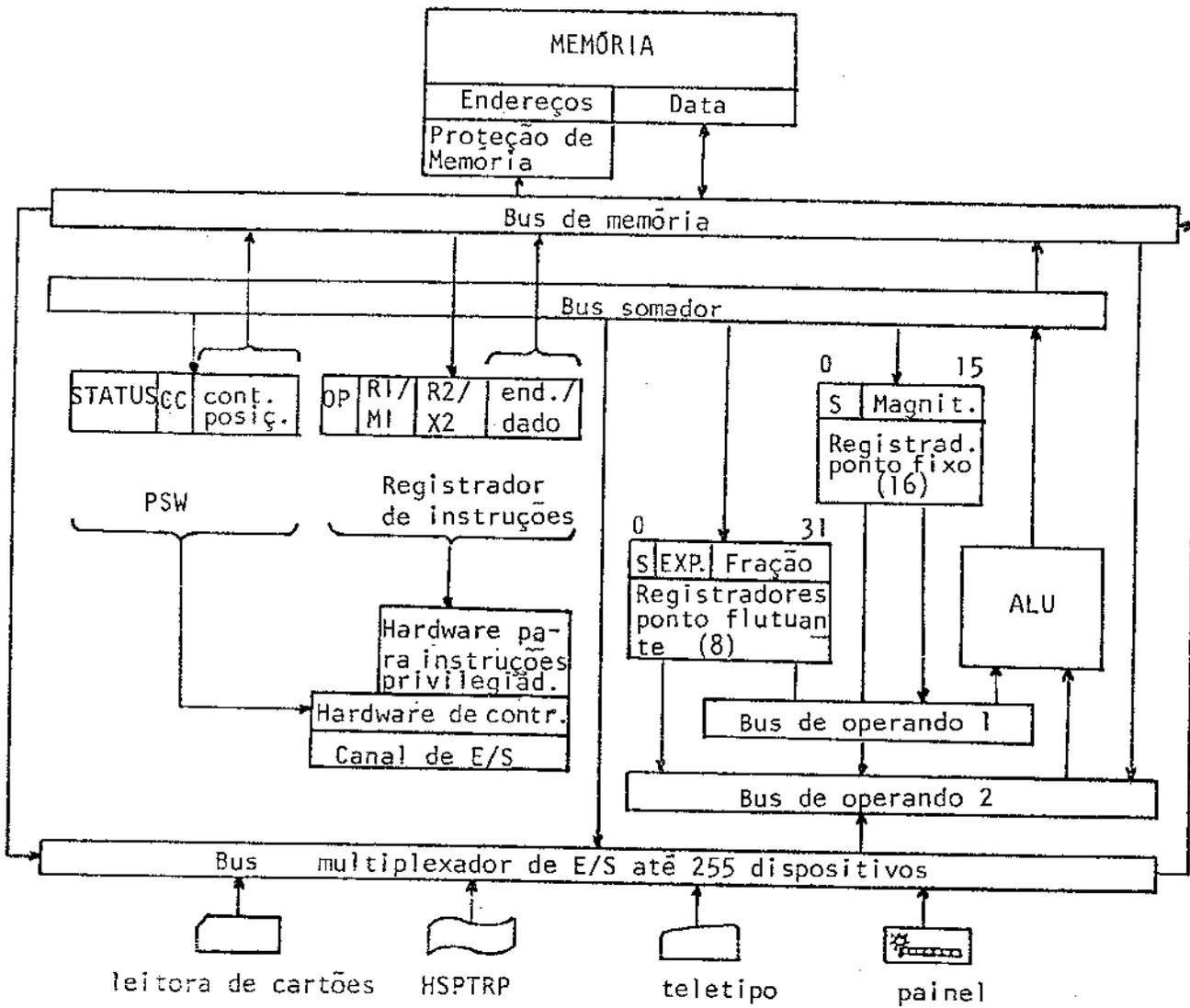


Fig. 3 - Diagrama de bloco do Processador

Analisando-se agora do ponto de vista de Engenharia de Software, o Interdata 80 possui um conjunto de 113 instruções, as quais estão enquadradas em quatro formatos: Registro a Registro (RR), Formato Curto (SF), Registro a Memória Indexada (RX) e Registro a Armazenamento (SR), que manipulam dados de três diferentes comprimentos de palavra: byte (8 bits), meia palavra (16 bits) e palavra completa (32 bits). Estes dados podem representar: números em ponto fixo, ponto flutuante ou dados lógicos.

Os formatos de instrução especificam três informações: a operação a ser executada, o endereço do primeiro operando e o endereço de segundo operando. O primeiro operando é normalmente o conteúdo de um Registro Geral, assim como o segundo operando, embora este ainda possa ser o conteúdo de uma posição de memória, ou uma constante da palavra da instrução.

Dispõe ainda de 16 Registradores funcionando como acumuladores ou registradores índices em toda operação lógica ou aritmética. Cada registrador geral tem 16 bits consistindo de 2 "bytes".

As posições de memória são "bytes" numerados consecutivamente de 0000 até 65.535.

Na seção seguinte analisa-se, em detalhes, as modificações que se necessitou fazer para anexar módulos que operassem a estrutura de Rede por nós escolhida, assim como discute-se o procedimento de operação do novo Sistema Operacional projetado.

2.1.2. Signetics 2650 | 10 |

É um microprocessador de propósitos gerais, constituído por um único circuito integrado tendo um conjunto fixo de instruções (75) e operando em paralelo sobre dados binários de largura igual a 8 bits, possui ainda capacidade de endereçamento de 32.768 "bytes" de memória acessíveis em 4 páginas de 8.192 "bytes" cada.

Seus 8 modos de endereçamento são uma valiosa ferramenta da qual dispõe o programador no sentido de otimizar seus programas.

O comprimento variável das instruções (1, 2 ou 3 "bytes") (permite uma sensível economia de memória resultando em diminuição de tempos de execução dos programas.

As figuras 4 a 7 resumem também algumas outras características fundamentais do microprocessador Signetics 2650, apresentando diagrama de blocos da arquitetura do microprocessador, registros disponíveis para utilização; assim como o formato das instruções e uma especificação das mesmas.

2.1.3. Outros dispositivos

Os outros microprocessadores não foram diretamente envolvidos na realização desse projeto inicial: tinha-se em mente suas características fundamentais e procurou-se dentro dos objetivos de generalidade e modularidade propostos, delinear estratégias de implementação que se adequassem mais convenientemente a futuras configurações e adaptações.

2.2. ESTRUTURA ESCOLHIDA. ESPECIFICAÇÃO

Tendo como objetivo a utilização de minicomputador Interdata 80 como um computador principal (host computer), assim como a idéia de tal dispositivo exercer o controle do fluxo de informações, o encadeamento e monitoração do processamento distribuído realizado em um nível hierárquico inferior pelos microcomputadores; a estrutura que escolhemos dentre as mencionadas adequa-se mais propriamente àquela denominada "Indirect Centralized Shared Bus with Central Switch - (ICS- Bus with Central Switch)", explicitada a seguir.

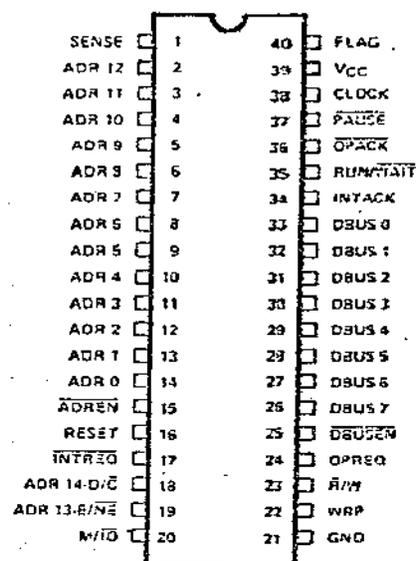
Tal estrutura de interconexão entre processadores, baseada na manipulação de mensagens entre os dispositivos e na topologia do "hardware" das interconexões resulta numa arquitetura em que a interação se realiza através do compartilhamento de um "bus" de comunicação conectado a um processador central.

Então, quando um dispositivo pertencente à rede deseja estabelecer um diálogo com o processador central para uma troca de dados ou informações sobre o andamento de determinados processos, este primeiro adquire o controle "bus" e então transmite a mensagem para o computador central.

O tratamento e a supervisão do processador central às tarefas em andamento em toda a rede pode resultar em reenvio de mensagens (dados ou tarefas e seus futuros destinos).

CARACTERÍSTICAS

- . Processador de propósitos gerais
- . Único "chip"
- . Conjunto fixo de instruções
- . Operações paralelas dados binários 8 - bit
- . Pastilha 40 pinos
- . Tecnologia MOS
- . Compatível a entrada e saída TTL
- . Única fonte de alimentação de +5 volts
- . 7 registradores de propósito geral
- . Pilha de endereços de retorno no chip 8 níveis de profundidade
- . Endereçamento 32K byte
- . Linhas de endereços e dados separadas
- . Instruções de comprimento variável de 1, 2 ou 3 bytes
- . 75 instruções
- . Tempo do ciclo de máquina de 2.4 μ s à frequência do relógio de 1.5 MHz
- . Instruções diretas levam 2,3 ou 4 ciclos
- . Única fase nível TTL do relógio de entrada
- . Lógica estática
- . "Buses" de saída "Three-state"
- . Modos de Endereçamento: Registrador, Imediato, Relativo, Absoluto Indireto e Indexado
- . Vetor de Interrupções



Configuração dos pinos.

Fig. 4 - Características principais dos S2650

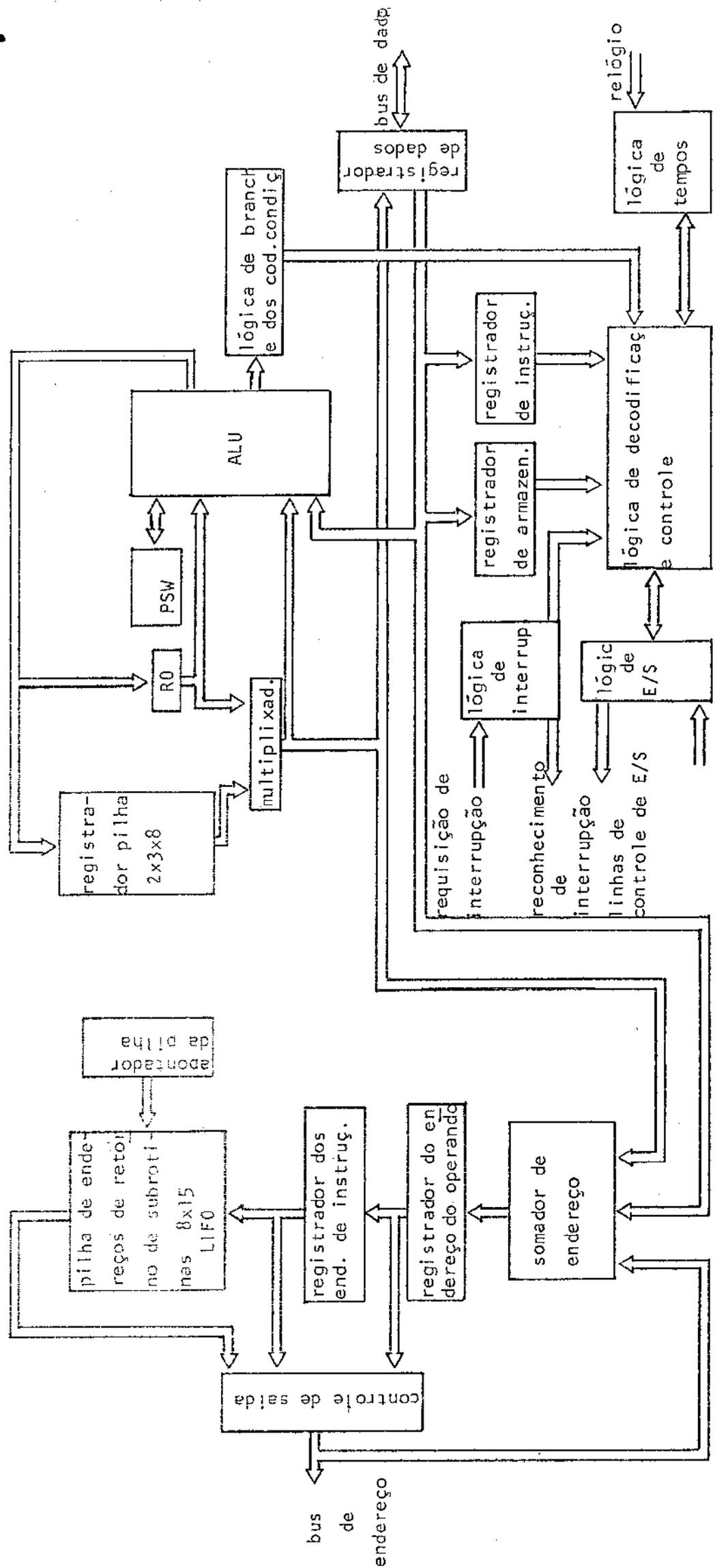


Fig. 5 - Diagrama de blocos do S2650

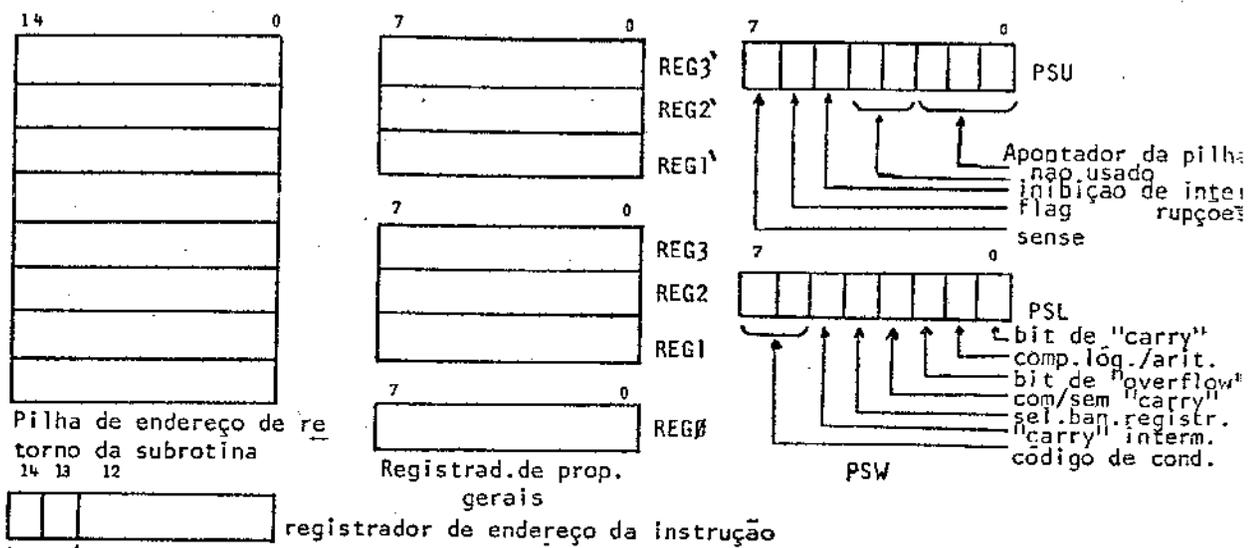


Fig. 6 - Principais registradores do S8650

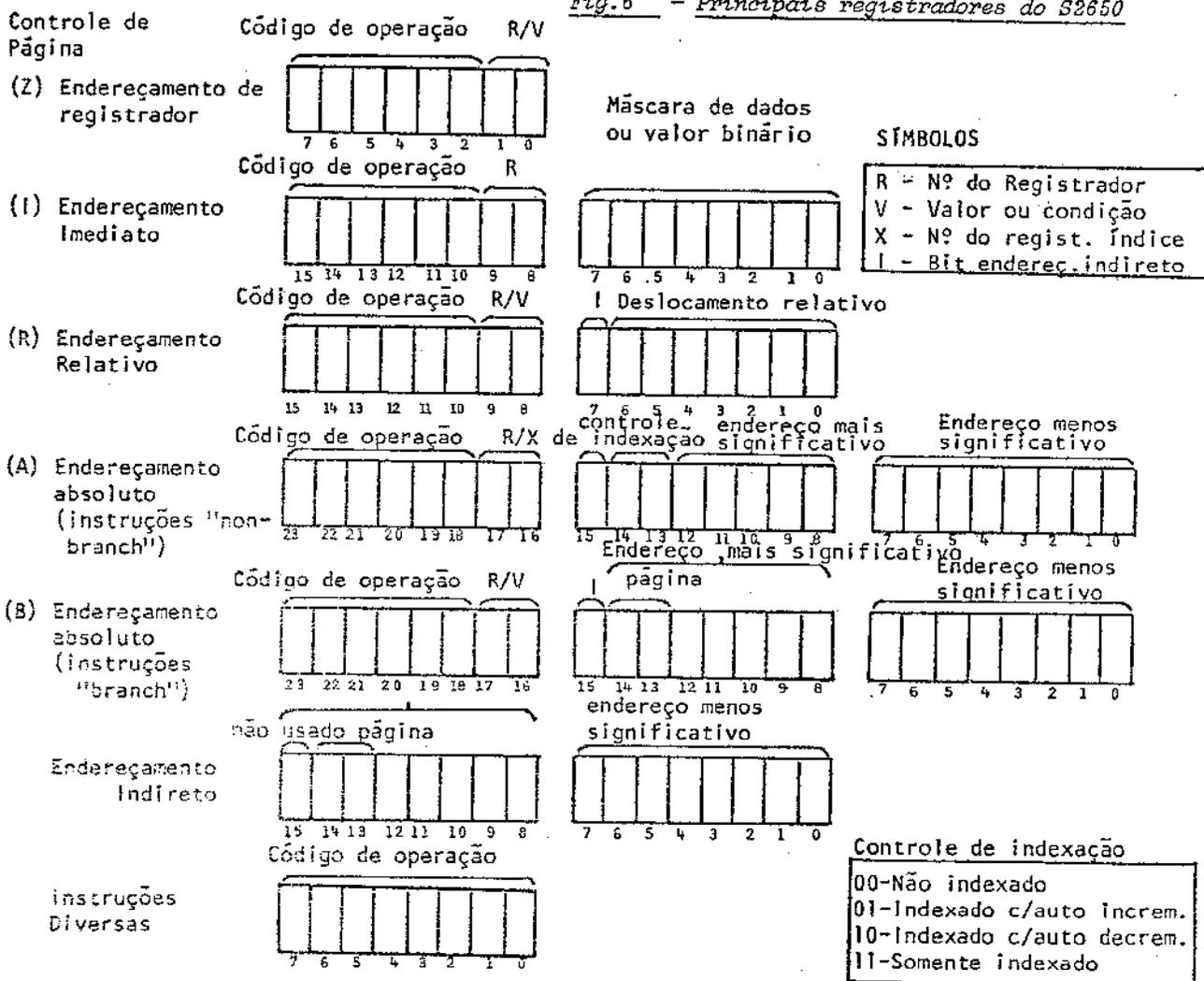


Fig. 7 - Formato das Instruções

Como mencionado nas características desse tipo de conexão, o compartilhamento do "bus" não contribui significativamente para um gargalo, mas é passível de balancear sua performance com a do processador central. Pois, o risco do gargalo não é aumentado em relação à estrutura tipo ICDS-Star, pelo fato de dupla saturação ("bus" e comutador) simultânea, ou pela saturação numa primeira etapa do processador central. Ressalta-se também a possibilidade de redução dos custos de conexões futuras no sentido de se desenvolver uma interface para um dispositivo que se ligará ao "bus" e não ao processador central.

Nossa preocupação baseou-se na criação de uma interface de propósitos gerais, adaptável o mais facilmente possível a um novo periférico que venha a fazer parte da Rede de Microcomputadores projetada.

2.2.1. Requisitos necessários

Apresenta-se agora alguns dos requisitos que um Sistema Operacional para Redes de Processadores deve satisfazer [6]:

- capacidade de suportar primitivos de comunicação de usuários;
- capacidade de suportar primitivos de migração (controle de movimentação) de "jobs";
- capacidade de suportar primitivos de migração de dados, e
- capacidade de suportar primitivos de controle.

Podendo se entender por primitivos, o conjunto de elementos necessários à implementação de uma função específica a um nível mais baixo não acessível pela tarefa do usuário.

Em termos das exigências a se satisfazer para a criação de um Sistema Operacional que suporte a estrutura pré-fixada, considera-se os seguintes pontos mais diretamente relacionados com os dois últimos itens acima:

- . uma solução que garanta a maior independência possível entre os processadores tendo em mente a aplicação da rede ao controle de processos. Tal independência refere-se à liberdade de cada processador na execução de sua tarefa específica dentro da atividade requerida, por exemplo: cômputo de algoritmos de controle de processos, limitando-se todavia, sem trazer prejuízos à perfor

mance do sistema, a uma troca de informação (dados e respostas) para a tomada de alguma decisão e conseqüente progresso na execução do seu procedimento.

- . protocolo de diálogos realizados entre esses processadores e o computador central, o mais simples, completo e geral possível.
- . resolução de problemas de competição no acesso a regiões comuns através das quais se realiza a troca bidirecional das informações.

Além desse objetivos primordiais pode-se considerar a necessidade do máximo aproveitamento possível do Sistema Operacional já existente no computador central, assim como a cuidadosa análise de estratégias de alocação de recursos através de algoritmos de exclusão mútua. Há que se tentar também para a definição conjunta com um engenheiro de hardware de uma "Interface Generalizada" para a Rede de Microcomputadores por nós elaborada. Em todas as fases de desenvolvimento do trabalho buscou-se o aproveitamento de experiências pessoais adquiridas no uso de outros sistemas de computação.

Na seção seguinte detalha-se os módulos do novo Sistema Operacional desenhado, bem como exemplifica-se seu funcionamento.

BIBLIOGRAFIA

- |3| Anderson, George A. & Jensen, E, Douglas
Computer Interconnection Structures: Taxonomy, Characteristics and Exam
ples.
Computing Survey, vol. 7 n° 4, December 1975, pp. 97-213.
- |4| Hoare, C.A.R. & Perrot, R.H.
Operating Systems Techniques
Academic Press, New York, 1972.
- |5| Interdata User's Manual.
Publication Number 29-261R01. February 1973.
- |6| Kimbleton, Stephen R. & Schneider, G. Michael.
Computer Communication Networks: Approaches, Objectives and Performance.
Considerations.
Computing Surveys, vol. 7 n° 3. September 1975, pp. 129-173.
- |10| Signetics 2650. Microprocessor Manual.
Signetics Corporation, 1975.

SEÇÃO 3. UM SISTEMA OPERACIONAL PARA REDE DE MICRO-PROCESSADORES

3.1. Detalhes de Implementação

Antes de se detalhar um Sistema Operacional para Redes de Micro-processadores, explicitar-se-á a arquitetura do Sistema proposta.

Como já mencionado, objetiva-se o desenvolvimento final de uma rede de processadores que realiza o Processamento distribuído e descentralizado no controle de processos por computadores [11]. Assim é que dentro da estrutura hierárquica prevista, (fig.8) tem-se o Interdata 80 em um segundo nível, como um processador central de uma estrutura (Indirect Centralized Shared Bus com comutador central) realizando toda a lógica de troca de informações.

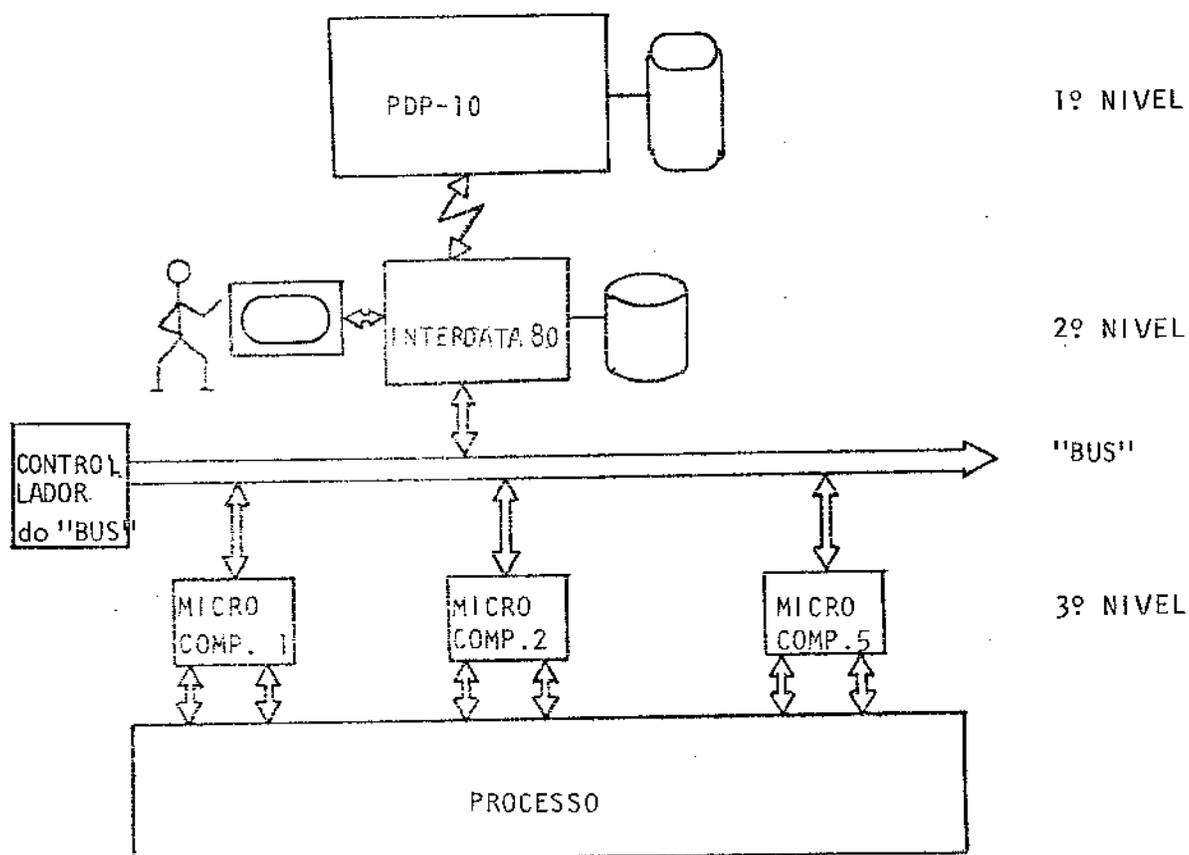


Fig. 8 - Estrutura da Rede de processadores da FEC- UNICAMP

Tal estrutura como já mencionado na seção 2 - Estrutura Escalhada e Requisitos Necessários, necessita para a sua operação de um Sistema Operacional que realize todos esses protocolos de comunicação, tratando também dos problemas concernentes a processos paralelos, gerenciamento de recursos e memória, tratamento e escalonamento "(scheduling)" de tarefas e outros procedimentos afins.

Estas questões foram resolvidas pela adoção das seguintes estratégias:

- A solução para o problema de independência dos processadores foi obtida pelo uso de interrupções no sequenciamento do diálogo;
- Quanto a implementação de soluções que resolvessem problemas de exclusão mútua na aquisição do controle de áreas comuns de dados, decidiu-se pelo uso de variáveis assinaladoras de estado (V.A.E.), que são uma restrição ao conceito dos semáforos de Dijkstra admitindo-se somente os valores: 0 - livre e 1 - ocupada. As situações no acesso a essas áreas comuns são resolvidas pela atribuição de prioridades ao processador central e aos micro-processadores de acordo com a dinâmica de andamento dos processos;
- Os protocolos de comunicação foram desenvolvidos analisando-se os processos de atendimento a interrupções por parte dos elementos de rede, e caracterizam-se por minimizar o tempo de ocupação do processador central em qualquer evento ocorrido.

Descreve-se agora os mecanismos de interrupção referentes ao processador central e ao microcomputador envolvido no projeto da Rede, analisando-se os detalhes de implementação particulares de cada elemento da rede.

3.1.1. Interdata 80

A estrutura de interrupção do Processador Central fornece rápida resposta a eventos internos e externos que requerem os serviços de rotinas especiais.

Na resposta à interrupção, o processador preserva o "estado corrente" da tarefa em execução, e dele próprio, saltando então para a

rotina de serviço requerida para o atendimento do evento ocorrido.

As tabelas 2 e 3 informam os tipos de interrupção e seus "bits" associados na "PSW (Program Status Word)" do processador, assim como a alocação das palavras de memória mais importantes usadas pelo Sistema Operacional; têm interesse apenas o caso das interrupções externas. Tabela 4.

O Interdata 80 tem a possibilidade de, no caso de interrupções permitidas, atender às mesmas de duas maneiras diversas, conforme o estado dos "bits" 1 & 4 da "PSW". Temos então:

- "External Interrupt" no qual o processador armazena a PSW atual na posição referenciada pela "Input/Output Old PSW location". A nova PSW é carregada da "Input/Output New PSW Location". Assim, o Sistema deve reconhecer a interrupção, identificar o dispositivo e tomar as ações apropriadas.

Esta técnica foi incluída nos novos modelos para manter a compatibilidade com os antigos;

- "Immediate Interrupt Service" no qual, a partir do reconhecimento do sinal de interrupção, quando esta é permitida, o sistema tem acesso à "Interrupt Service Pointer Table" (localizações $00D0_{16}$ a $02CF_{16}$), no ponto indicado pelo valor $(00D0_{16})$ somado ao dobro do endereço hexadecimal do periférico, daí retirando o apontador para a rotina de interrupção do mesmo. De forma esquemática verifica-se, para esse segundo caso:

$$\begin{array}{l} \text{Current PSW (3:31)} \rightarrow (\text{Service Pointer}) \text{ Old PSW save area} \\ \text{Current PSW (4:15)} + (\text{Service Pointer}) + 4 \\ \text{Current PSW (16:31)} + (\text{Service Pointer}) + 6 \end{array} \} \text{ new PSW}$$

Através desse mecanismo de interrupção, uma rotina para cada dispositivo é inicializada em um determinado instante.

Dentre os Sistemas Operacionais suportados pelos Interdata 80, vide Tabela 5 optou-se pela utilização do "DOS (Disc Operating System)" por sua versatilidade na manipulação do disco "(unidade de memória secundária) como também pela possibilidade de utilização de "overlays", comandos chamados ao supervisor para ativação de programas, manutenção de arquivos, tratamento de E/S por interrupção, etc.

Tabela 2 - Tipos de interrupção

INTERRUPÇÃO	BIT DE CONTROLE DO PSW
External	1
Machine Malfunction	2
Fixed Point Divide Fault	3
Automatic I/O Service	4
Floating-Point Arithmetic Fault	5
Channel Termination	6
Protect Mode	7
Illegal Instruction	Não pode ser desativado
Channel Termination Queue Overflow	Não pode ser desativado
Supervisor Call	Não pode ser desativado

BIT	NOME	COMENTÁRIOS
0	WT Wait State	O bit de wait é set para parar a execução do programa. Quando este bit está set no PSW corrente, não ocorre nenhuma execução de programa, mas o processador irá responder a toda interrupção de Entrada/Saída ou mau funcionamento da máquina, se estas estão permitidas.
1	ET External Interrupt Enable	O bit de interrupções externas é set para fazer o processador responsável pelos sinais de interrupção do 'Multiplexor Bus'
2	MM Machine Malfunction Interrupt Enable	O bit de mau funcionamento da máquina permite a ocorrência de uma interrupção se uma falha na alimentação é detetada, se a máquina está equipada com a opção de paridade de memória e ocorre um erro de paridade de memória, ou durante a reinicialização do processo seguindo uma queda de alimentação.
3	DF Fixed Point Divided Fault Interrupt Enable	O bit de assinalamento de uma interrupção por falha na Divisão Ponto Fixo permite tal evento quando tal instrução é, executada e o resultado não pode ser expresso e, 16 bits.
4	AS Automatic Input/Output Service Enable	O bit de permissão de serviço de entrada/saída automático permite ao processador reconhecer interrupções de entrada/saída e servi-las automaticamente.
5	FP Floating-Point Arithmetic Fault Interrupt Enable	O bit de interrupção por falha na aritmética de ponto flutuante permite ao processador reconhecer interrupção se ocorre 'overflow' ou 'underflow' durante alguma operação de ponto flutuante.
6	CT Channel Termination Interrupt Enable	O bit que permite a interrupção de término de operação do canal refere-se ao canal automático de Entrada/Saída, o qual pode ser usada conjuntamente com o Serviço automático de Entrada/Saída.
7	PM Protect Mode	O bit de modo de proteção da memória permite sua proteção e a deteção de instruções privilegiadas. Quando ele está ativado o processador é dito estar no modo supervisor.
8 - 11	Unused	Deve ser zero.
12	C Carry/Borrow	Os bits de código de condição são set ou ajustados após a execução de instruções pelo processador
13	V Overflow	
14	G Greater than Zero	
15	L Less than Zero	

Tabela 4 - Alocação das palavras de memória do Interdata 80

Function	Hexadecimal Memory Address	Assignment
Floating-Point Registers (Models 70, 80 only)	00-03	Floating-Point Register, R0
	04-07	Floating-Point Register, R2
	08-0B	Floating-Point Register, R4
	0C-0F	Floating-Point Register, R6
	10-13	Floating-Point Register, R8
	14-17	Floating-Point Register, R10
	18-1B	Floating-Point Register, R12
	1C-1F	Floating-Point Register, R14
Power-Fail Locations	20-21	Unassigned
	22-23	Register Save Pointer
	24-27	Current PSW Save Area
Interrupt PSWs	28-2B	Old PSW FLPT Arithmetic Fault Interrupt
	2C-2F	New PSW FLPT Arithmetic Fault Interrupt
	30-33	Old PSW Illegal Instruction Interrupt
	34-37	New PSW Illegal Instruction Interrupt
	38-3B	Old PSW Machine Malfunction Interrupt
	3C-3F	New PSW Machine Malfunction Interrupt
	40-43	Old PSW External Interrupt
	44-47	New PSW External Interrupt
48-4B	Old PSW Fixed-Point Divide Fault Interrupt	
4C-4F	New PSW Fixed-Point Divide Fault Interrupt	
Reserved	50-7F	Bootstrap Loader and Device Definition Table
Channel I/O Termination Parameters (Models 70, 80 only)	80-81	Termination Queue Pointer
	82-85	Old PSW Channel I/O Termination Interrupt
	86-89	New PSW Channel I/O Termination Interrupt
	8A-8B	Overflow Termination Pointer
	8C-8F	Old PSW Termination Queue Overflow Interrupt
90-93	New PSW Termination Queue Overflow Interrupt	
Supervisor Call Parameters	94-95	Supervisor Call Argument Pointer
	96-99	Old PSW Supervisor Call
	9A-9B	New PSW (Status and Condition Code) Supervisor Call
	9C-9D	New PSW (Location Counter) Supervisor Call 0
	9E-9F	New PSW (Location Counter) Supervisor Call 1
	A0-A1	New PSW (Location Counter) Supervisor Call 2
	A2-A3	New PSW (Location Counter) Supervisor Call 3
	A4-A5	New PSW (Location Counter) Supervisor Call 4
	A6-A7	New PSW (Location Counter) Supervisor Call 5
	A8-A9	New PSW (Location Counter) Supervisor Call 6
	AA-AB	New PSW (Location Counter) Supervisor Call 7
	AC-AD	New PSW (Location Counter) Supervisor Call 8
	AE-AF	New PSW (Location Counter) Supervisor Call 9
	E0-B1	New PSW (Location Counter) Supervisor Call 10
	B2-B3	New PSW (Location Counter) Supervisor Call 11
	B4-B5	New PSW (Location Counter) Supervisor Call 12
B6-B7	New PSW (Location Counter) Supervisor Call 13	
B8-B9	New PSW (Location Counter) Supervisor Call 14	
EA-BB	New PSW (Location Counter) Supervisor Call 15	
BC-CF	Reserved	
Interrupt Service Table	D0-D1	Service Pointer, Device 0
	D2-D3	Service Pointer, Device 1
	D4-D5	Service Pointer, Device 2
	•	
	•	
	•	
2CC-2CD	Service Pointer, Device 254	
2CE-2CF	Service Pointer, Device 255	

Tabela 5 - Sistemas Operacionais do Interdata 80

	BOSS/4B	BOSS/4A	BOSS	DOS	RTOS	RTEX
Program Number	03-021	03-020	03-019	03-022	04-017	04-019
User Manual Publication Number	29-216	29-216	29-292	29-293	29-240	29-294
CPU Models Supported	4*	4*	5, 50, 70, 74, 80	5, 50, 70, 74, 80	70, 80	50
Multiprogramming	No	No	No	No	Yes	Yes
Program Segmentation	No	No	No	Yes	Yes	No
Memory Protect Support	No	No	No	No	Yes	No
Bulk Storage Supported	None	None	Disc, Drum	Disc	Disc, Drum	None
Other Devices Supported	TTY, HSPTR/P CR, LP, MT	TTY, HSPTR/P CR, LP, MT	TTY, HSPTR/P CR, LP, MT, Cassette	TTY, HSPTR/P CR, LP, MT, Cassette	TTY, HSPTR/P CR, LP, MT, Digital MUX, Clocks, Interrupt Module	TTY, Interrupt Module, Clocks, Special Communications Devices
I/O Method	Wait	Interrupt	Wait	Interrupt	Interrupt	Interrupt
Bulk Storage File Management	N/A	N/A	Numbered Files	Named Files with Automatic Space Allocation and Overflow	Numbered Files	N/A
Bulk Storage I/O Method	N/A	N/A	Direct	Automatic Blocking/ Deblocking	Direct	N/A
Bulk Storage File Protection	N/A	N/A	No	Yes	Yes	N/A
Number of Files per Bulk Storage Device	N/A	N/A	SYSGENable to 256	400	32	N/A
Number of Logical Units	8	6	SYSGENable to 256	SYSGENable to 16	16 per task	16 per task
SYSGEN Capability	No**	No	Yes(source)	Yes(source)	Yes (source or object)	Yes(source or object)
BATCH - Mode Capability (Transfer Command)	No	No	Yes	Yes	No	No
SVC's Supported	1, 2, 3	1, 2, 3	1, 2, 3	1, 2, 3, 4, 5, 6***	1, 2, 3, 5, 6, 8, 10***	1, 2, 3, 6, 10, 15***
Memory Required (KB)	3	4.5	2.5-4.5	4-10	16(min)	10(min)

* On Models 3, 50, 70, 74, and 80, BOSS/4B is superseded by BOSS, and BOSS/4A is superseded by DOS.

** A SYSGENable version of BOSS/4B is available from the INTERDATA user's group.

***SVC 6 is not compatible between DOS, RTOS or RTEX.

As rotinas escritas para tratar dispositivos do usuário anexados ao Sistema podem ser elaboradas, de uma maneira mais adequada, a dotando-se as mesmas convenções de ligação obedecidas pelos "drivers" do Interdata.

Os seguintes passos são então cumpridos na elaboração desta lógica:

- O programa do usuário executa uma instrução de chamada ao supervisor "(Supervisor Call - SVC)" para que este execute Entrada / Saída;
- O supervisor armazena os registros do programa de usuário (programa de aplicação), estabelece a relação entre endereços lógicos e físicos, manipula os apontadores da área de dados a ser utilizada, saltando para a rotina de tratamento do dispositivo com interrupções externas permitidas "(enabled)";
- Esta rotina faz algum processamento para verificação inicial, e executa BALR, DVR, INTRET o qual retorna ao programa supervisor;
- A instrução SINT (Simulate Interrupt) é então executada na subrotina SVCRET do executivo que bloqueia ("disabled") futuras interrupções para não haver outro atendimento imediato. O supervisor carrega os registros do dispositivo com as informações pertinentes ao mesmo através do seu "DCB - Device Control Block," que é uma região indicada por apontador na "Service Pointer Table" e que contém tais dados.
- Agora o dispositivo está sob controle de sua rotina de tratamento; esta pode ativá-lo e executar outra BALR DVR, INTRET para esperar interrupção pelo periférico quando este estiver pronto ("ready");
- Quando toda a tarefa referente a este periférico estiver cumprida, a sua rotina de supervisão de operações executa ATL SPTR, LIOTRM, que causa uma interrupção de término de trabalho, adicionando tal elemento a uma lista dos dispositivos disponíveis, retornando através da instrução BALR DVR, INTRET ao módulo final da rotina do dispositivo.
- As interrupções externas estão novamente permitidas "(enabled)" e o tratamento final refere-se à verificação de erros e retorno ao supervisor com condições de estado da operação através da instru

ção BR UBSY.

Este é o procedimento geral de funcionamento dos dispositivos conectados ao processador central (Interdata 80), e que deve ser seguido na reconfiguração do Sistema pela conexão de um novo periférico.

As figuras 9 e 10 esquematizam tal rotina a ser adotada para realizar o objetivo de adicionar ao Sistema Operacional módulos para dispositivos do usuário, e a tabela 6 explica os registradores do "DCB" e seus usos.

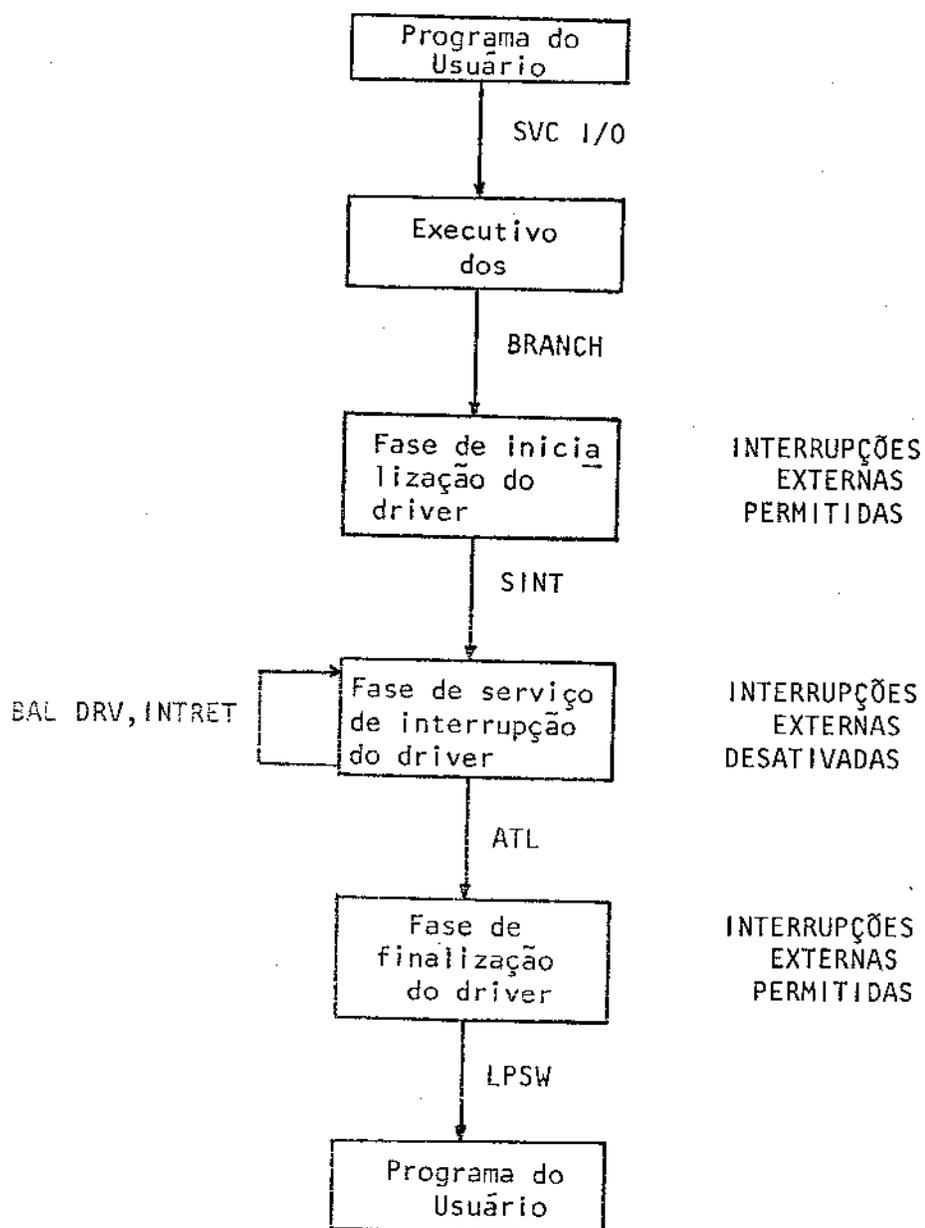


Fig. 9 - Fluxo geral da lógica de E/S

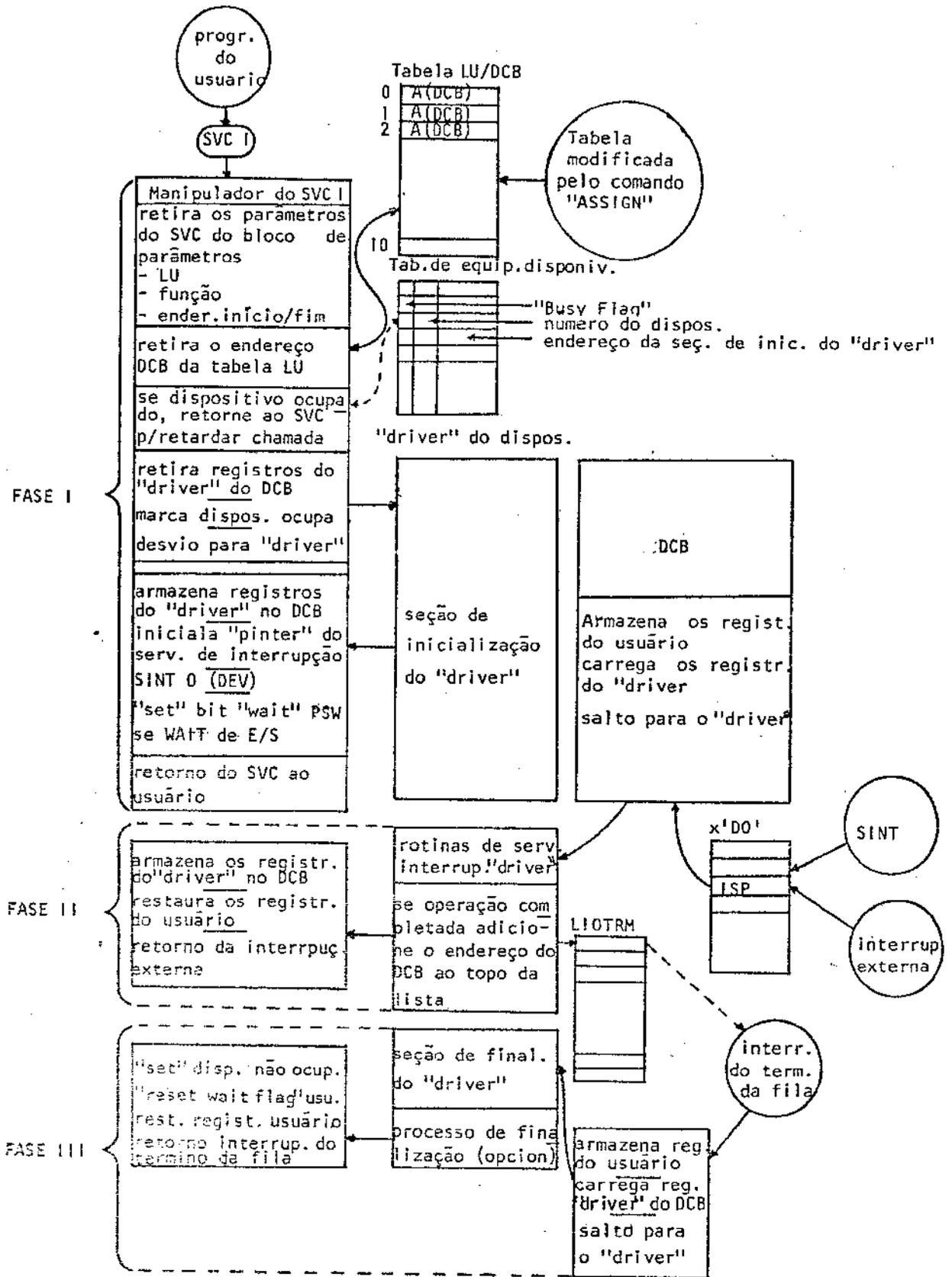


Fig. 10 - Fluxo detalhado da lógica E/S

Tabela 10 - DCB de um dispositivo-registradores e seus usos

Registrador	Mnemônico	Conteúdo das Constantes	Significado
0	DVR		endereço de re-entrada de um "driver" sempre que se sai do mesmo; DVR deverá estar apontando para o ponto de retorno.
1	INTRET	SVCRET ou INTRTN	endereço de retorno da rotina do SVC (primeira entrada no "driver") ou o endereço da rotina de interrupções externas (entradas subsequentes).
2	AX2		Apontador para o bloco de parâmetro do SVC (supervisor call-chamada do supervisor).
3	FCN		conteúdo da primeira meia-palavra do bloco de parâmetros do SVC (código da função e unidade lógica)
4	CBA		endereço do início do "buffer" do usuário
5	ONE	1	constante 1
6	FBA		endereço do final do "buffer" do usuário
7	SPTR	A(DEVISP)	apontador para a entrada do serviço de interrupção do "Device Control(DCB)".
8	DEV	DEV	número do dispositivo
9	AET	A(DEVBSY)	apontador para a "flag" de ocupação do dispositivo na tabela.
10	USBY	A(UBSY1)	apontador para a rotina comum de finalização do "driver".
11	ERRTN		registrador do "status" do erro.
12	AC0		acumulador geral
13	AC1(RTN)		acumulador geral (endereço de retorno da subrotina).
14	AC2		acumulador geral
15	AC3		acumulador geral

Nota-se que o tempo em que as interrupções não estão permitidas para imediato atendimento foi minimizado, implicando em maximização do uso do processador. Esclarecimentos mais detalhados com um exemplo podem ser vistos em [5].

É necessário lembrar que a comunicação entre os elementos da rede e o processador central se efetua aleatoriamente, ora um, ora outro comandando as requisições, e que o acesso às áreas comuns é outro ponto crítico a gerenciar e supervisionar.

As seguintes modificações no Sistema Operacional para adicionar os módulos referentes à operação e tratamento dos microcomputadores foram realizadas:

- O "Pointer" do dispositivo na "Service Pointer Table" é agora permanentemente mantido, uma vez que tal apontador era inicializado e utilizado somente durante a transferência, por comando do Interdata, e após tal procedimento colocava-se novo "pointer" NOP para não operação. Esta alteração foi necessária porque na nova estrutura as comunicações são executadas a partir de qualquer elemento (processador central ou microprocessador);
- Poder-se-ia já modificar a lógica do "DCB (Device Control Block)" do dispositivo, rotina que elabora a mudança de contexto na interrupção, armazenando os registradores da tarefa corrente e restaurando os do dispositivo, mas preferiu-se mantê-la apenas alterando a inicialização de alguns registros do periférico, em vista da possibilidade de ser requerida uma ativação do módulo antes do Interdata utilizá-lo, com o consequente preenchimento dos registradores.

Neste mesmo sentido deixou-se de utilizar algumas rotinas já disponíveis e compatíveis com outros dispositivos do sistema, pois precisava-se agora adequá-las a suportar também a estrutura composta pelos microcomputadores; particularmente as rotinas comuns de término de operação, nas quais é marcado "NOP" no "Pointer".

Resumidamente, o atendimento a alguma interrupção ativada pelos microcomputadores resulta nos seguintes procedimentos:

- realizada a mudança de contexto com armazenamento dos regis tro s do programa em execução e restauração dos registros da ro tina do dispositivo, o programa salta para a rotina SENSE3, na qual é conhecido o código que explica o motivo da interrupção. Esta rotina, tendo acesso ao vetor de interrupção VETINT, reti ra o endereço correspondente ao trecho de programa que fará o tratamento do evento requisitado.

Um esboço gráfico de tal operação está contido na figura 11 .

A flexibilidade de tal implementação reside no fato do usuário, desejando adicionar novas funções, poder alterar a interface no sentido de gerar sinal e código correspondente à nova interrupção. Inicializando em VETINT (vetor de interrupções) o endereço correspondente à nova ro tina que executará o atendimento desse evento, e escrevendo o trecho de pro grama correspondente à sua manipulação.

Os comandos do módulo correspondente à rotina de término de execução de uma determinada transferência, podem continuar a ser usados, pois os mesmos preservam o apontador da "Service Pointer Table" que é essencial ao funcionamento adequado do sistema.

O módulo de supervisão de um microcomputador Signetics 2650 segue anexo em linguagem assembler do Interdata 80, com comentá rios sobre suas operações.

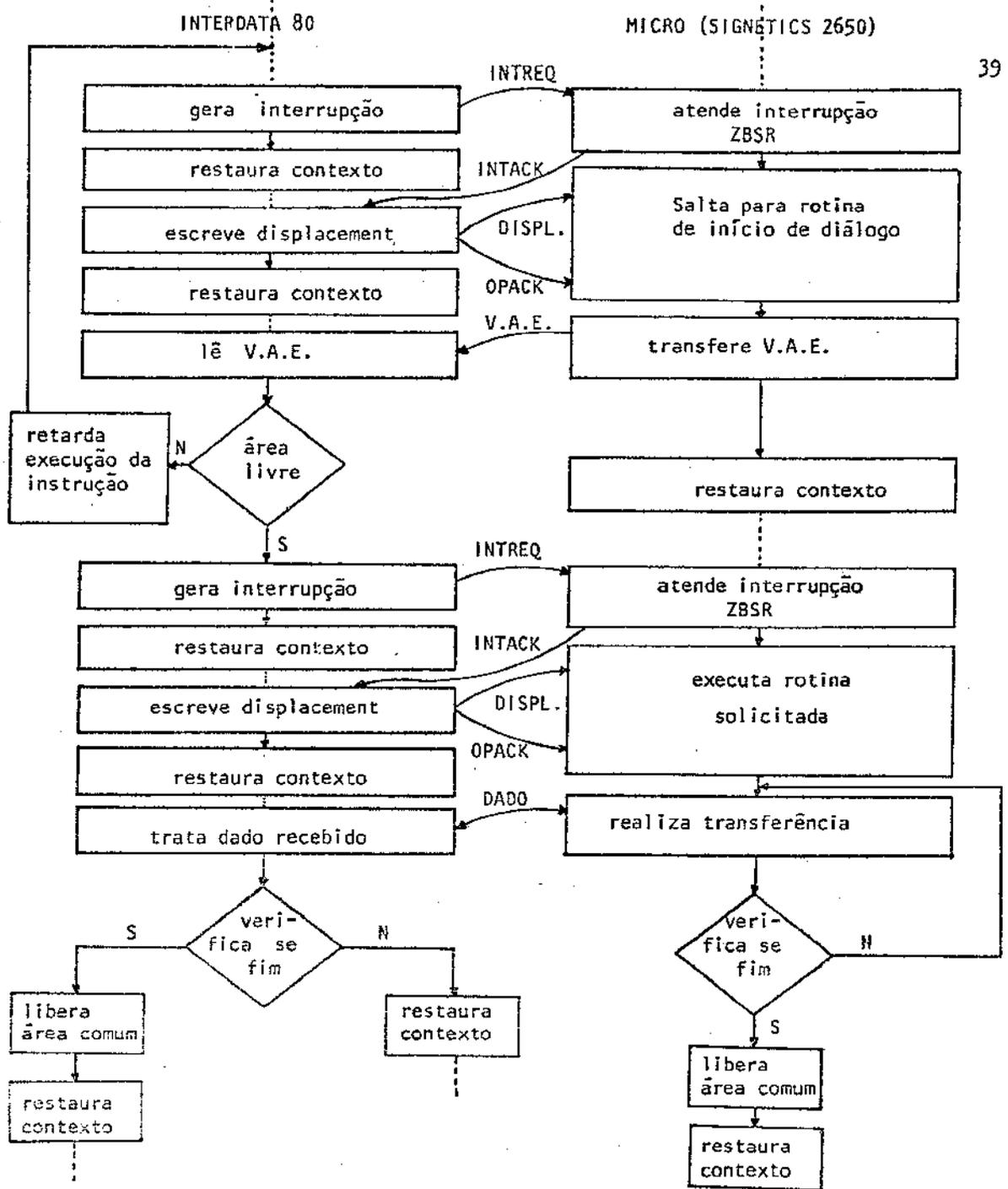
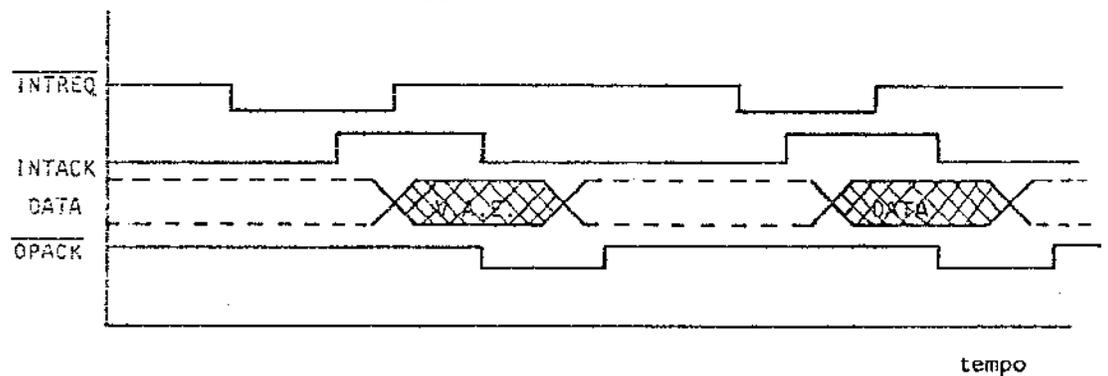


Fig. 11 - Diagrama de blocos do protocolo das comunicações
(o INTERDATA 80 comanda)



SP1DCB	SENSE3, INTRTN, 0, 0, 0, 1, 0, SP1ISP, DZ, SP1BSY, UBSY5, 0, 0, 0, 0, 0	
SP1ISP	DC 0, 0, 0	EXTERNAL INTERRUPT PSW SAVES AREA
	STM 0, RSAVE	SAVE USER REGISTERS
	LM 0, *-42	LOAD DRIVER REGISTERS
	BR DVR	GO TO DRIVER
	IF SPENT1	
AREA1	DS BUFM1	DEFINE COMMON AREA
AREA2	EQU *	
ENTRY1	LHR AC0, FCN	IGNORE COMMANDS
	BMR UBSY	
	THI AC0, X'1000'	ONLY BINARY VALID
	BZ UBSY4	
	BALR RTN, INTRET	RETURN TO USER
	DC DEV, INTREQ	INTREQ REQUEST
	LHR DEV, DEV	
	AC DEV, DEACTIV	DEACTIVATED INTREQ
	BALR RTN, INTRTN	RETURN TO USER
	MHR DEV, DISPL1	DISPLACEMENT TO INITIAL CONVERSATION
	DC DEV, OPACK	GENERATE OPACK SIGNAL
	BR INTRET	RETURN TO WAIT INTERRUPT IN RESPONSE
RDSTUS	RHR DEV, AC2	READ STATUS OF COMMON AREA
		0 - FREE; 1 - BUSY;
	DC DEV, OPACK	IF BUSY AREA, JUMP TO HANG UP
	LHR AC2, AC2	THE CALL FOR TRANSFER
	BNZ BSYXIT	
	DC DEV, INTREQ	
	LHR DEV, DEV	
	DC DEV, DEACTIV	
	BALR RTN, INTRTN	
	MHR AC0, AC0	
	BM REDDI	IF MINUS READ DATA
	MH DEV, DISPL2	DISPLACEMENT FOR READ
		ROUTINE OF SIGNETICS
	DC DEV, OPACK	
	DC DEV, RDHW	
	LB AC2, 0(CBA)	
	XHI AC2, X'00FF'	COMPLEMENT DATA, BECAUSE 52650
		LOGIC IS DIFFERENT OF 180
	MHR DEV, AC2	
	DC DEV, OPACK	
	CLHI AB, X'00'	COMPARE WITH DELIMITER CHARACTER
	BE HSRPDM	
	BXLE CBA, INTRTN	
	B HSRPDM	
REDDI	MH DEV, DISPL3	DISPLACEMENT FOR WRITE ROUTINE
		OF SIGNETICS
	DC DEV, OPACK	
	BR INTRET	
REDD	DC DEV, WRHW	
	RHR DEV, AC2	
	XHI AC2, X'00FF'	
	STB AC2, 0(CBA)	
	CLHI AC2, X'FF'	
	BE HSRPDM	
	BXLE CBA, INTRTN	
	B HSRPDM	

INT0	WH DEV, SEMAF DC DEV, DPACK LH AC2, SEMAF BNZR INTRET AHM ONE, SEMAF LHI AC2, X'80' STB AC2, SP1DCB LH CBA, AREA1 LH FBA, AREA2 BR INTRET	WRITE SEMAPHORE OF THE AREA MAKE AREA BUSY
UBSY4	LHI ERRTN, X'C000' BS *+6	
UBSY0	LHI ERRTN, X'A000'	
UBSY5	LHR ERRTN, ERRTN BZS UBSY12 OHR ERRTN, DEV	
UBSY12	STH ERRTN, 2(AX2) XHR RTN, RTN STB RTN, 0(A7) STH A2, SEMAF LM PSW, X'96' THI FCN, X'800' BZS ZERO NHI PSW, X'7FFF'	AT END OF TRANSFER, MARK FREE AREA KILL WAIT FLAG
ZERO	STH A2, SP1DCB+6 B TSCHEO	
SENSES3	SSR DEV, AC2 AHR AC3, AC3 BZR RTN LH RTN, INTVET(AC3)	ROUTINE TO SENSE STATUS OF THE DRIVE
	BR RTN	
SEMAF	DC 0	
INTRET	DC X'206E'	
DPACK	EDU INTRET+1	
DISPL1	DC X'0000'	
DISPL2	DC X'0000'	
DISPL3	DC X'0000'	
WRHW	DC X'0000'	
RDRW	EDU WRHW+1	
INTVET	DC 0 AC(UBSY0), AC(UBSY4), AC(UBSY5), AC(UBSY12), AC(RDRW), AC(SENSES3), AC(INT0)	COMMANDS TO UNIVERSAL INTERFACE MODULE(U. I. M.) INTERRUPT VECTOR DEVICE UNAVAILABLE STATUS AND ADDRESS OF THIS ROUTINE MICRO COMPUTER WANTS TO READ DATA ADDRESS OF THE WRITE DATA ROUTINE OF 180 MICRO COMPUTER WANTS TO WRITE DATA, ADDRESS OF THE READ DATA ROUTINE OF 180 MICRO COMPUTER WRITES SEMAPHORE; ADDRESS OF THE ROUTINE OF THE READ SEMAPHORE OF 180. MICRO COMPUTER WANTS TO READ SEMAPHORE ADDRESS OF THE ROUTINE TO WRITE SEMAPHORE OF 180

```
SUPTOP  IF  1  
        EQU  *  
        END
```

```
        IF SPENT1  
LHR  RTN, SP115F  
LHR  AC2, SPENT1  
ARR  AC2, AC2  
ARI  AC2, X'0000'  
STH  AC2, SPT  
STH  RTN, SPT  
SPT  DC  0
```

ROUTINE TO INITIALIZE POINTER
IN THE SERVICE POINTER TABLE,
INSERTED AT THE BEGINNINGG
EXECUTABLE PART OF
OPERATING SYSTEM

3.1.2. Signetics 2650

A sequência de eventos ocorrido no caso de interrupção requerida por um periférico (INTREQ ativo), estando as interrupções permitidas, é a seguinte:

- O processador termina a execução da instrução corrente.
- Aciona o "Interrupt Inhibit" não permitindo futuras interrupções.
- Insere o primeiro "byte" da instrução "ZBSR (Zero Branch-to-subroutine Relative)" no registrador de instrução, no lugar do que deveria ser o código da próxima instrução sequencial a ser executada.
- O processador tem acesso ao "bus" de dados para ler o segundo "byte" da instrução ZBSR.
- O processador executa a instrução ZBSR preservando o endereço da instrução seguinte à interrompida no "Register Address Stack (RAS)", e procede à execução da instrução à "página" 0, byte 0 ou ao endereço relativo à "página" 0, byte 0 fornecido pelo dispositivo que interrompeu.

Os comentários seguintes referem-se a esse segundo "byte" da instrução (deslocamento), através do qual, pelo periférico provocador da interrupção (processador central ou futuramente outro processador da rede), associa-se o endereço, de uma rotina "simetricamente" correspondente à tarefa requerida. Por exemplo: deseja-se efetuar uma leitura de dados; passa-se então ao S2650 o "displacement" da rotina de escrita de dados. Tal raciocínio é análogo para outras tarefas.

No S2650 desenvolveu-se rotinas para:

- atendimento de "interrupção inicial" (verificação do estado da área comum) e consequente manipulação da variável assinaladora de estado (V.A.E.);
- leitura de dados e armazenamento em área pré-definida.
- escrita de dados a partir de áreas pré-fixadas;
- mudança de contexto do programa em execução para alguma das rotinas acima e restauração de contexto do programa que estava sendo trabalhado.

Tais procedimentos interligados com o programa de aplicação do usuário carregado no micro, executam as tarefas de:

- no caso do Interdata 80 comandar as transferências, este processador central, pelo envio de um deslocamento ("displacement"), conveniente-endereço da rotina simétrica à função requerida; dirige as operações do Signetics para armazenar o contexto do programa em execução, atender à interrupção solicitada e restaurar o contexto da tarefa suspensa; objetivando assim cumprir o protocolo das comunicações.
- na situação do comando das operações ser realizado, pelo próprio micro-processador, o programa de aplicação, pelo preparo adequado dos limites da área, e pelo uso das instruções (REDC ou WRTC) toma conhecimento; da disponibilidade do processador central, ativando rotinas de consulta do Interdata 80, para realizar apropriadamente a transferência das informações.

A figura 12 representa graficamente o protocolo de comunicação das operações comandadas pelo microcomputador e o módulo de envolvido em linguagem assembler do S2650, segue anexo.

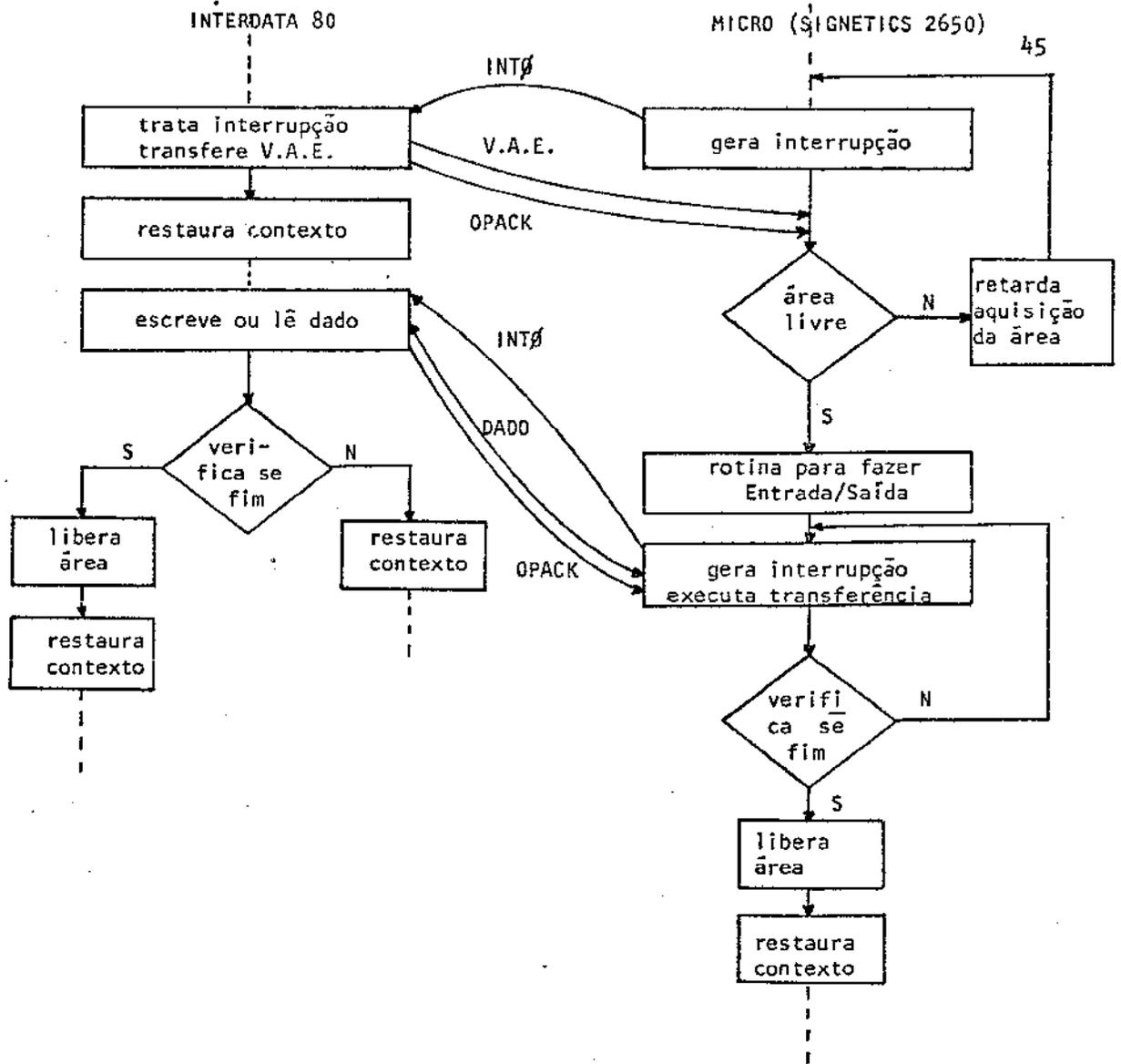
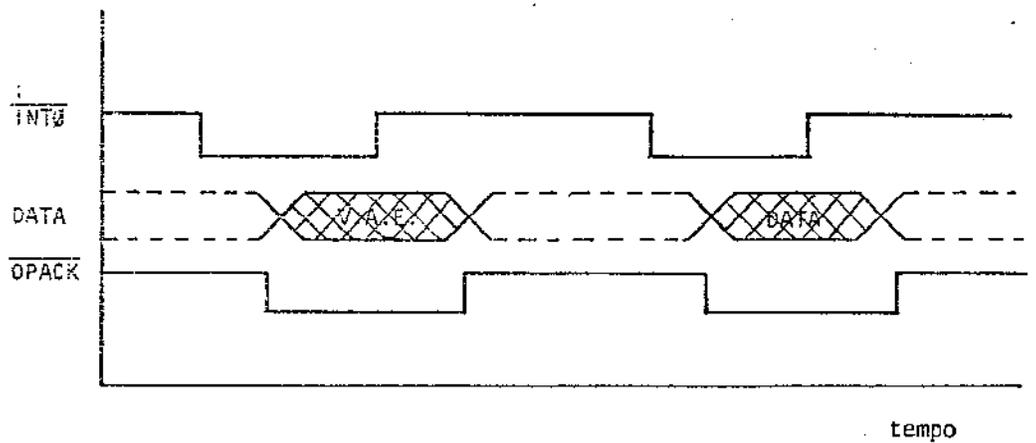


Fig. 12 - Diagrama de blocos do protocolo das comunicações (o Micro comanda)



```

R0 : EQU 0          INICIALIZACAO DOS REGISTROS
R1 : EQU 1
R2 : EQU 2
R3 : EQU 3
UN  : EQU 3
EQ  : EQU 0
RS  : EQU #10
FIN : EQU #FF
ORG  #00
RTIN: DA INIT1     ENDERECO DAS ROTINAS
RTRD: DA READ
RTWR: DA WRTE
ORG  #20

```

```

INIT1: STRA R0, RGSV1  ROTINA DE ATENDIMENTO DA INTERRUPCAO
      SPSL              INICIAL DO PROTOCOLO DE CONVERSACAO
      STRA R0, POSI+1  DOS PROCESSADORES.
      SPSU              1) SALVA OS REGISTROS DO PROGRAMA
      STRA R0, RGSV1+2 DO USUARIO EM EXECUCAO NO SIGNETICS
      SPSL RS
      STRA R1, RGSV1+3
      STRA R2, RGSV1+4
      STRA R3, RGSV1+5
      PPSL RS
      STRA R1, RGSV1+6
      SYRA R2, RGSV1+7
      STRA R3, RGSV1+8
      LODA R0, SEM     CARREGA SEMAFORO DA AREA E
      WRTO R0          TRANSFERE AO INTERDATA
      BCFA EQ, RTRRS
      ADDZ R0, 1      SE AREA LIVRE ASSINALA OCUPADA
      STRA R0, SEM
      LODA R0, AREA1
      STRA R0, LIM1
      LODA R0, AREA2
      STRA R0, LIM2
      SUBI R0, AREA1
      STRA R0, NCHAR
      STRA R2, PGT
      COHA R2, NCHAR
      BCTA EQ, INIT
      BCTA UN, WRCC

```

```

RTRRS:  PPSL R5          ROTINA RESTAURADORA DOS REGISTROS
        LODA R1, RGSV1+6 DO PROGRAMA QUE ESTAVA SENDO EXECUTADO
        LODA R2, RGSV1+7
        LODA R3, RGSV1+8
        CPSL R5
        LODA R1, RGSV1+3
        LODA R2, RGSV1+4
        LODA R3, RGSV1+5
        LODA R0, RGSV1+2
        LPSU
        LODA R0, RGSV1
        CPSL H'FF'
POSI:   PPSL #00
        RETE UN

READ:   STRA R0, RGSV    ROTINA DE LEITURA DO 52650
        CPSU #00
        BCTA UN, RTSV    SALTO PARA ROTINA DE SALVAR REGISTROS
RECO:   REDD R0
        LODA R2, POT
        STRA *R2, LIM1(1)
        COMI R0, FIM
        BCTA EQ, INIT
        STRA R2, POT
        COMA R2, NCHAR
        BCTA EQ, INIT
        BCTA UN, RECO
INIT:   LODI R2, 255     INICIALIZA O CONTADOR QUE SERA
        STRA R2, POT     O INDICADOR DA AREA
        BCTA UN, RGRS    SALTO PARA ROTINA DE RESTAURAR REGISTROS
WRTE:   STRA R0, RGSV
        PPSU #00
        BCTA UN, RTSV    SALTO PARA ROTINA DE SALVAR REGISTROS
WRCD:   LODA R2, POT
        LODA *R2, LIM1(1)
        WRTD R0
        COMI R0, FIM
        BCTA EQ, INIT
RTSV:   SPSL            ROTINA QUE SALVA REGISTROS DO
        STRA R0, POSI2+1 PROGRAMA QUE ESTAVA EM EXECUCAO
        SPSU
        STRA R0, RGSV+2
        CPSL R5
        STRA R1, RGSV+3
        STRA R2, RGSV+4
        STRA R3, RGSV+5
        PPSL L5
        STRA R1, RGSV+6
        STRA R2, RGSV+7
        STRA R3, RGSV+8
        TPSV #00
        BCTA EQ, WRCD
        BCTA UN, RECO

```

```

RGRS:  PPSL R5          ROTINA QUE RESTAURA REGISTROS
        LODA R1, RGSV+6 DO PROGRAMA A SER REEXECUTADO
        LODA R2, RGSV+7
        LODA R3, RGSV+8
        CPSL R5
        LODA R1, RGSV+3
        LODA R2, RGSV+4
        LODA R3, RGSV+5
        LODA R0, RGSV+2
        LPSU
        LODA R0, RGSV
        CPSL H'FF'

PDSI2: PPSL #00
POT:   DC 255          APONTADOR DO BYTE DENTRO DA AREA
RGSV1: AREA 9         AREA PARA SALVAR REGISTROS DA
                     ROTINA INICIAL DE INICIALIZACAO DO
                     PROTOCOLO DE CONVERSACAO

RESV:  AREA 9         AREA PARA SALVAR REGISTROS DAS
                     ROTINAS DE LEITURA OU ESCRITA

SEM :   DC 0          SEMAFORO DA AREA COMUM
                     0 PARA LIVRE;
                     1 PARA OCUPADO

LIM1:  DC 0          LIMITE INFERIOR DA AREA COMUM
LIM2:  DC 0          LIMITE SUPERIOR DA AREA COMUM
NCHAR: DC 0          NUMERO DE CARACTERES A SER TRANSFERIDO
                     TAMBEM USA-SE DELIMITADOR 'FF'

AREA1: AREA 255      AREA COMUM PARA TRANSFERENCIA DE
                     DADOS-LIMITE INFERIOR
AREA2: DC *          LIMITE SUPERIOR AREA COMUM
        END

```

3.2. Comentários sobre alocação de recursos dos Processadores.

Analisando-se globalmente a intercomunicação entre o processador central e os dispositivos podemos observar que qualquer deles pode transferir uma região de dados para a área comum do outro, e ler a área comum do outro, armazenando em qualquer área de dados.

A aquisição da área comum do processador destino é realizada através de uma "interrupção inicial" que verifica o seu estado e se a mesma está livre ocupa-a; se ocupada a requisição é então retardada para posterior atendimento.

No caso da aquisição da própria área comum, parte-se da hipótese que o usuário conhecendo as particularidades do sistema deve implementar os procedimentos necessários para tal função, como por exemplo:

- O usuário do Interdata 80 desejando usar sua própria área comum pode adicionar uma unidade à variável assinaladora de estado (V.A.E.), através da instrução "AHM ONE, SEMAF (add halfword memory one to semaphore)" e partindo da convenção: 0- área liberada; 1- área ocupada, verifica-se que se o usuário obtiver 1 no instante seguinte à adição a área será sua; em caso contrário, pela obtenção do valor 2, teremos uma interrupção ocorrida entre execução das instruções soma e de teste passando então a área a ser ocupada pelo microprocessador;
- O usuário do S2650 pode facilmente verificar a possibilidade de ocupação de sua própria área comum pela inserção de blocos de instruções que:
 - . inibem interrupções;
 - . verificam variável assinaladora de estado ocupando ou não a área;
 - . restauram a permissão de interrupções.

Tal sequência de procedimentos pode ser facilmente implementada em qualquer ponto do programa de aplicação.

Resta ainda mencionar que na interrupção inicial o dispositivo informante da variável assinaladora de estado da área pretendida marca-a ocupada pelo outro se a mesma estiver livre; caso contrário transfere o estado

de ocupação da mesma ao periférico inquiridor.

Observa-se então que esses casos de alocação de recursos podem se tornar "deadlocks" em potencial, cuja solução será a extensão do diálogo através da troca de prioridades dos periféricos, como decorrência do andamento dos processos, ou ainda em situação de empate de altas prioridades, prevalecendo a atividade do Processador Central.

3.3. Considerações Gerais sobre a Operação do Sistema.

O novo Sistema Operacional projetado, mantém as características fundamentais do "DOS - Disc Operating System" original, com o acréscimo dos módulos de tratamento dos microcomputadores, cuja função principal é a maior liberação possível do processador central, através do uso racional dos mecanismos de interrupção e das rotinas dos novos periféricos que interferem o mínimo possível em cada uma das fases do diálogo.

Em termos de "Software" pode-se ainda esclarecer que cada dispositivo está associado à sua área comum "(buffer)" de armazenamento, e cada um dos mesmos tem sua respectiva variável assinaladora de estado.

Quanto à expansão, o computador central Interdata 80 permite a ligação de 255 outros periféricos ao seu "Multiplexor Bus". Assim o usuário pode, num trabalho conjunto "Hardware/Software", aumentar a Rede seguindo procedimentos já descritos quanto às novas alterações.

O Sistema Operacional é gerado a partir de informações do usuário sobre a existência ou não de micro-processadores, compilando então esses módulos que, no tempo de carregamento e inicialização, realizam o acerto dos apontadores na "Service Pointer Table", o que o coloca em condições para executar sua tarefa na Rede.

BIBLIOGRAFIA

- | 5| Interdata User's Manual.
Publication Number 29-261R01, February 1973.

- |10| Signetics 2650. Microprocessor manual.
Signetics Corporation, 1975.

SEÇÃO 4 . CONCLUSÕES

Apresentam-se a seguir algumas considerações sobre os resultados obtidos deste trabalho, assim como enunciam-se os possíveis e prováveis melhoramentos suscitados pela pesquisa.

Dentre os esforços que se estão empreendendo para a implementação de uma Rede de Processadores no Laboratório da Faculdade de Engenharia de Campinas, este projeto contribuiu para a realização de parte desse objetivo, qual seja o estabelecimento das conexões Interdata 80 e micro-processadores, ligando o 2º ao 3º nível da estrutura proposta (figura 8, seção 3).

O Sistema Operacional desenvolvido, pode ser utilizado para a ligação com o DEC10 (1º nível). Tal ligação pode ser realizada considerando o processador como mais um micro-processador. O módulo que realiza o tratamento desse novo dispositivo pode ser elaborado a partir dos procedimentos descritos na seção 3.

Atualmente implementam-se as ligações entre os micro-processadores, objetivando a minimização de um possível gargalo no comutador (processador central).

Claramente, após a realização deste trabalho, já se visualizam os pontos nos quais se pode aperfeiçoar o "software" desenvolvido.

Sallenta-se, e já se está a estudar, em vista do crescimento da Rede de Processadores, as seguintes questões principais:

- refinamento do protocolo de comunicações visando máxima independência dos dispositivos;
- estudo do balanceamento de carga dos elementos constituintes da Rede, objetivando ótimo uso dos recursos;
- estudo dos requisitos necessários à realização dinâmica no Sistema Operacional (reconfiguração) perante falhas nos dispositivos.

Preve-se o cumprimento desses tópicos em trabalhos futuros.

BIBLIOGRAFIA

- | 1 | Anderson, Georg A. & Jensen, E, Douglas
Computer Interconnection Structures: Taxonomy Characteristics and Examples.
Computing Surveys, vol. 7, nº4, December 1975, pp.197-213.
- | 2 | Coffman Jr., Edward G. & Denning, Peter J.
"Operating Systems Theory"
Prentice Hall, 1973.
- | 3 | Frank, H.
Computer Networks: Art to Science to Art.
Networks, vol. 5, pp. 7-32.
- | 4 | Hoare, C.A.R. & Perrot, R.H.
Operating Systems Techniques
Academic Press, New York, 1972.
- | 5 | Interdata User's Manual.
Publication Number 29-261R01, February 1973.
- | 6 | Kimbleton, Stephen R. & Schneider, G. Michael.
Computer Communication Networks: Approaches, Objectives and Performance Considerations.
Computing Surveys, vol.7 nº 3. September 1975, pp. 129-173.
- | 7 | Nievergelt, J. & Farrar, J.C.
What machines Can and Cannot Do.
Computing Surveys, vol. 4 nº 2, June 1972, pp. 81-96.
- | 8 | Rosen, S.
Electronic Computers - A Historical Survey
Computing Surveys, vol. 1 nº1 , March 1969, pp. 7-36.
- | 9 | Rosin, Robert F.
Supervisory and Monitor Systems.
Computing Surveys, vol. 1 nº1, March 1969, pp. 37-51.
- | 10 | Signetics 2650. Microprocessor Manual.
Signetics Corporation, 1975.

- [11] Suguy, Takao & Nóbrega, Manoel da & Netto, Márcio L.A.
Controle Distribuído e Descentralizado de Processos Industriais por
Computadores. Parte IV.
XII Congresso Nacional de Processamento de Dados - SUCESU - Rio de
Janeiro - 1979.