Anales PANEL'81/12 JAIIO
Sociedad Argentina de informática
e Investigación Operativa. Buenos Aires, 1981

SSIP: SIMULADOR DE INTERCONEXÃO DE PROCESSADORES

M. Tazza

R. Weber

Ph. Navaux

Universidade Federal do Rio Grande do Sul
Pós-Graduaçao em Ciência da Computaçâo
Av. Osvaldo Aranha 99, Porto Alegre, Brasil

RESUMO

O estudo de sistemas a múltiplos processadores desenvolveu-se muito estes últimos anos com o advento dos microprocessadores. Por conseqüência, tornou-se necessário criar ferramentas capazes de melhor determinar as características e desempenhos necessários para estes sistemas.

Este artigo trata do desenvolvimento de uma destas ferramentas, um simulador de interconexão SSIP, que tornou-se necessário para o projeto de um sistema multimicroprocessador SMM em execução na Universidade Federal do Rio Grande do Sul (UFRGS). Este simulador permite testar diferentes interconexões entre processadores e memórias, de maneira a verificar seu desempenho relativo e portanto auxiliar na definição de uma arquitetura melhor adaptada ao SMM.

É apresentado o núcleo do SSIP que simula a interconexão entre os processadores e memórias de um dado sistema multiprocessador.

O SSIP foi definido buscando flexibilidade para permitir que a disciplina básica de estabelecer ligações entre processadores e memórias pudesse ser alterada facilmente sem necessidade de mudanças na estrutura do núcleo.

Este trabalho é parcialmente financiado pela FINEP - Financiadora de Estudos e Projetos.

OBJETIVO

O objetivo deste artigo é apresentar os resultados obtidos no desenvolvimento de um Sistema Simulador de Interconexão entre Processadores e Memórias (SSIP). Descreve-se o sistema simulado e a estrutura lógica do simulador, apresentando-se ao final futuros desenvolvimentos. A utilidade deste simulador é permitir o estudo comparativo de diversas estruturas de interconexão medindo-se seus desempenhos relativos.

I. INTRODUÇÃO

O advento do microprocessador abriu novas portas para o arquiteto de sistemas, permitindo obter sistemas mais potentes, melhor dirigidos para as aplicações, com melhor segurança e a um baixo custo. Os primeiros projetos desenvolvidos com microprocessadores visavam usá-los para substituir o "hardware" de controladores. Hoje, no entanto, de mais em mais desenvolvem-se projetos com múltiplos microprocessadores; estes sistemas permitem obter uma maior potência de saída devido ao processamento paralelo, bem como maior tolerância a falhas.

Dois tipos principais de sistemas decorrem do uso de múltiplos processadores: os sistemas de microprocessadores distribuídos e os sistemas multimicroprocessadores. No primeiro caso o processamento é distribuído entre processadores que estão fisicamente interconectados mas que possuem um grau de acoplamento lógico variável (Thu.78). Por outro lado um sistema multiprocessador é um sistema com processadores semelhantes que partilham o acesso à memória comum, canais de entrada e saída, unidades de controle e periféricos, com um único sistema operacional controlando-o (Ens.75). O trabalho apresentado neste artigo concerne este último ítem.

Um dos aspectos mais importantes no estudo dos sistemas com múltiplos processadores é a interconexão que permite a comunicação entre estes. Como conseqüência, diversos estudos e classificações (And.75) apareceram nos últimos anos tentando estabelecer o melhor esquema de interconexão para determinados sistemas (TRI.79).

No desenvolvimento do projeto de um Sistema Multimicroprocessador (SMM) na UFRGS foram sentidos também os problemas inerentes à escolha da interconexão melhor adaptada à comunicação entre os processadores e as memórias. Tornou-se portanto necessário simular primeiramente o multimicroprocessador e, especificamente, a interconexão, para permitir uma escolha da arquitetura final com uma melhor base de conhecimentos dos desempenhos. No estudo da arquitetura do sistema precisou-se de uma ferramenta de software que permitisse especificar uma determinada arquitetura e acompanhar certos aspectos do seu comportamento. Para obter esta ferramenta, era viável desenvolver um simulador baseado numa Linguagem de Descrição de Hardware (LDH) que fizesse o mapeamento entre a arquitetura descrita e uma representação

interna capaz de ser simulada e avaliada quanto ao seu desempenho.

Dentro das LDH, diversos níveis de descrição são possíveis, como pode ser visto na tabela 1. O nível da LDH aconselhada para o estudo do SMM seria, pela classificação de Barbacci, o nível 1. Este aborda os aspectos de funcionamento e comunicação entre processadores, memórias, chaves e periféricos e portanto adapta-se perfeitamente ao estudo necessário sobre a interconexão do SMM.

Nível	Elementos
1. De sistema	processadores, memórias, chaves, periféricos
2. De programação	instruções de máquina e operação
3. Transferência de registradores	registradores, transferências entre registradores
4. Circuitos de chaveamento	subníveis seqüencial e combinacional
5. Circuitos	diodos, transistores, resistores

Tabela 1 - Classificação de Barbacci

No entanto, decidiu-se pela implementação, primeiro, de um núcleo do sistema simulador que permitisse melhor avaliar as necessidades da LDH. Esta decisão foi conseqüência do objetivo de se ter, a curto prazo, um sistema para definir diferentes ligações entre processadores e memórias e que possibilitasse acompanhar seu comportamento quando submetido a pedidos de processadores solicitando um acesso a memórias através do sistema de interconexão. Este núcleo forneceria, posteriormente, os elementos para a definição de uma LDH que possibilitasse a simulação de todo o SMM.

II. O SISTEMA A SER SIMULADO

O sistema a ser simulado pode ser visto, de forma genérica, na figura 1.

Na figura 1, P1, P2,... Pn representam processadores e M1, M2,...Mm representam módulos de memória. Os processadores e as memórias estão interligados através de uma estrutura de interconexão, que pode caracterizar diversas arquiteturas multiprocessadores. Por exemplo, a interconexão pode ser um barramento único, que interligue todos os elementos e seja utilizado de forma

multiplexada no tempo. Outra forma de estruturar a interconexão é através de uma rede crossbar (figura 2).

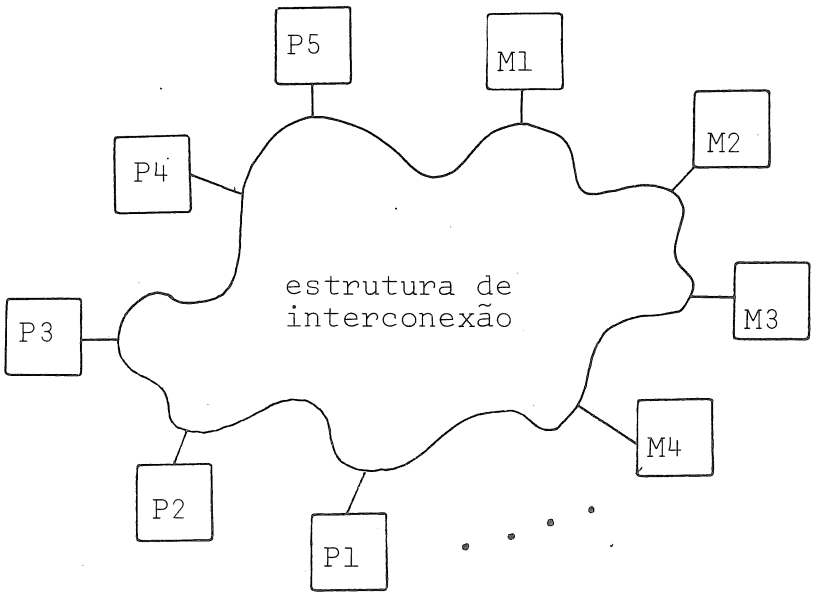


Figura 1 - Sistema multiprocessador

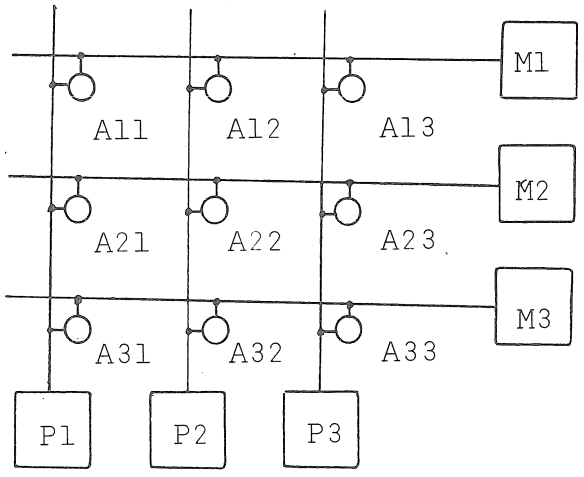


Figura 2 - Sistema com rede crossbar

A comunicação entre um determinado processador e uma memória é feita interligando-se os seus dois barramentos através de chaves. Por exemplo, se na estrutura da figura 3 o processador P2 deseja estabelecer contacto com a memória M1, isto é feito através da chave S21. Note-se que, estando esta ligação estabelecida, tanto o barramento que parte do processador P2 como o da memória M1 não podem mais ser utilizados por outros elementos.

Diversas outras estruturas de interconexão podem ser definidas, caracterizando assim diversos sistemas multiprocessado-

res (tri.79).

O Simulador reproduz o processo de estabelecer as ligações entre processadores e memórias para a estrutura de interconexão de uma determinada arquitetura multiprocessador. Desta maneira, o Sistema Simulador substitui a estrutura de interconexão mostrada na figura 1.

Para reproduzir o comportamento da interconexão quando solicitada para estabelecer o caminho, o Sistema Simulador necessita de vários elementos:

- Descrição interna da interconexão.
- Representação interna do estado de utilização da interconexão.
- Um conjunto de rotinas capazes de realizar análises e decisões quanto ao estabelecimento de um caminho de comunicação: os arbitradores.
- Um controle que reconheça os pedidos dos processadores, efetue as chamadas dos arbitradores para estabelecer a conexão desejada e simule o uso desta conexão.

Na próxima seção, descreve-se a estrutura global do sistema simulador. A notação utilizada é uma modificação de "funciogramas" (Ric.77). Um funciograma é um grafo com três tipos de nodos: ESTAÇÕES, ÁREAS e CANAIS DE AVISOS (figura 3).

Figura 3 - Elementos de um funciograma

As estações são elementos ativos e representam processamento. Podem ler/escrever em áreas de dados. Estas operações são representadas por arestas dirigidas da área para a estação (leitura) ou da estação para a área (escrita). A aresta dirigida nos dois sentidos indica leitura e escrita.

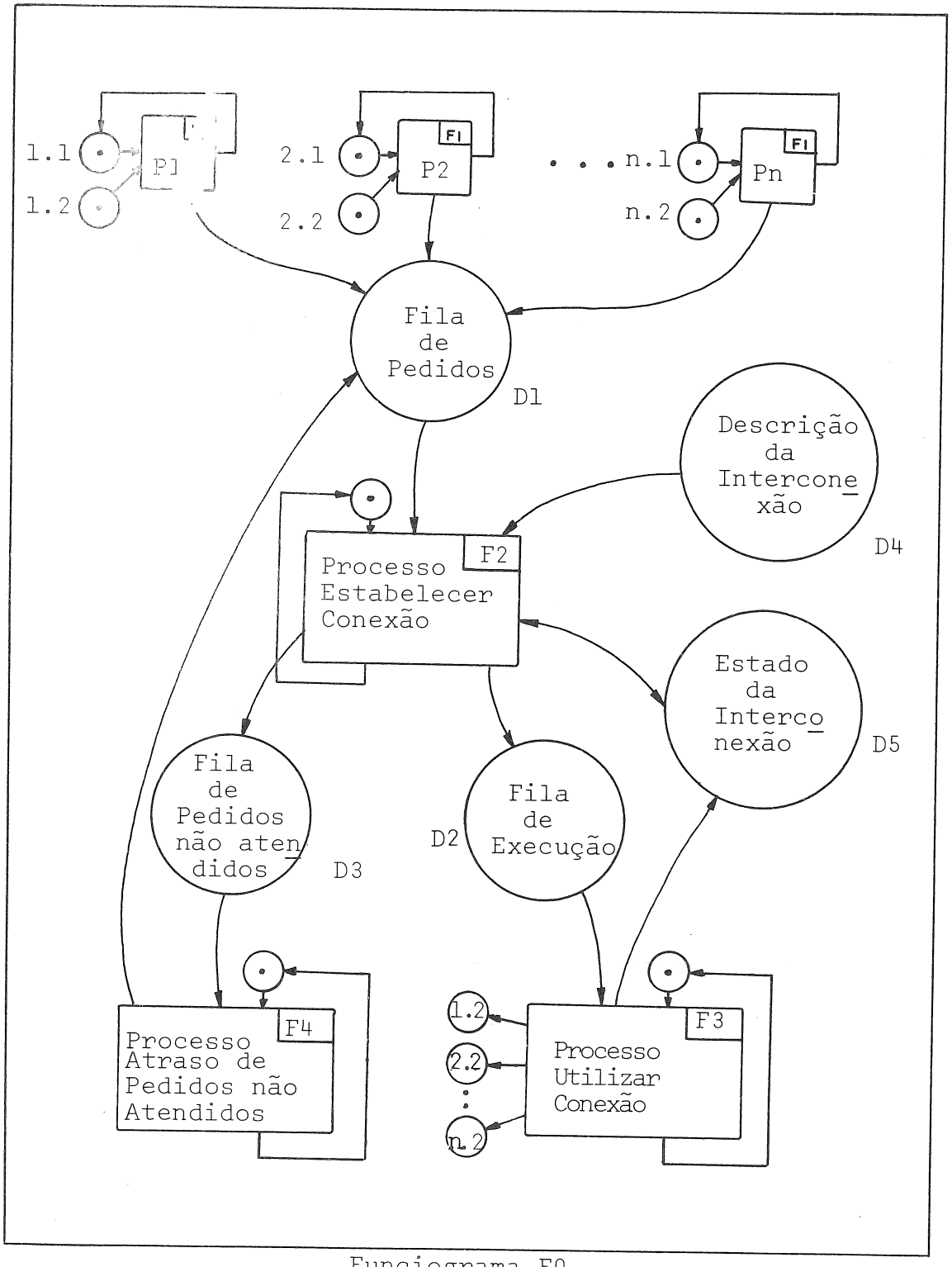
As áreas de dados são elementos estáticos, isto é, armazenam informações que podem ser acessadas pelas estações.

Os canais de aviso estabelecem o fluxo de controle entre as estações: uma estação só pode ser executada se os canais de aviso na sua entrada estiverem marcados. Uma estação sinalizada, após sua execução, coloca uma marca nos canais de aviso da sua saída.

III. ESTRUTURA E FUNCIONAMENTO DO NÚCLEO DO SIMULADOR

III.1 ORGANIZAÇÃO BÁSICA

A organização básica do simulador pode ser vista no funciograma F0.



Funciograma F0

As estações P1,P2,...Pn representam os processadores; sua estrutura interna será descrita no funciograma F1(seção III.2). A partir de uma situação inicial, isto é, estação habilitada, o processador gera um PEDIDO (o processador P1 deseja se comunicar com a memória Mj).

Em paralelo com os processos que representam os processadores, corre o processo ESTABELECER CONEXÃO, descrito em detalhes no funciograma F2(seção II.3). Este processo, basicamente, é um laço onde são retirados pedidos da área D1 (fila de pedidos) e, para cada PEDIDO, é feita a tentativa de estabelecer um caminho entre o processador Pi (identificado no PEDIDO) com a memória Mj. Se for possível estabelecer o caminho Pi-Mj, o conjunto de arestas que o compõe é colocado na área D2 (fila de execução). Se, por outro lado, não é possível, no momento, estabelecer Pi-Mj o próprio PEDIDO é colocado na área D3 (fila de pedidos não atendidos). As áreas D4 (descrição do esquema de interconexão) e D5 (descrição do estado) são consultadas no momento de estabelecer o caminho.

Em paralelo com os processos acima, corre o processo UTILIZAR CONEXÃO, descrito em detalhes no funciograma F3(seção III.4). Este processo é um laço onde se simula a utilização do caminho estabelecido entre um processador e uma memória. Após esta utilização é liberado o caminho (acesso a área D5), permitindo seu uso por outros pedidos. A utilização do caminho (troca de dados entre o processador P1 e a memória Mj) é simulada retardando sua liberação por um tempo IM1, que acompanha o pedido desde a sua geração. Após, o processador Pi é sinalizado, podendo enviar novo PEDIDO.

O processo PEDIDOS NÃO ATENDIDOS também corre em paralelo com os já descritos e é detalhado no funciograma F4(seção III.5). Sua função é a de recolocar os pedidos não atendidos por falta momentânea de recursos na área D1 para que o processo ESTABELECER CONEXÃO possa, numa nova volta do laço, tentar atendê-lo.

Estando estes processos em paralelo, e acessando áreas comuns, é necessário que exista exclusão mútua sobre certas áreas:

Área D1: fila de pedidos - acessada para escrita pelos processos P1,P2,...Pn e pelo processo PEDIDOS NÃO ATENDIDOS; acessada para leitura pelo processo ESTABELECER CONEXÃO. Esta área é estruturada como uma fila circular em alocação seqüencial, e para a sua manipulação existem um algoritmo de inserção e um de retirada; qualquer acesso será feito apenas através destes algoritmos. O número M de nodos será igual ao número de processadores Pi, i=1,2,...N; neste caso a estrutura nunca entrará em overflow; se esta situação tiver que ocorrer no estudo de um caso, a fila terá que ser redimensionada, tronando M menor que N. Pode ocorrer a situação de underflow. Um nodo da fila, chamado PEDIDO, tem a estrutura lógica apresentada na figura 4.

Todos os campos acima são montados em tempo de geração do

CEDINFOR

pedido pelo próprio processador Pi.

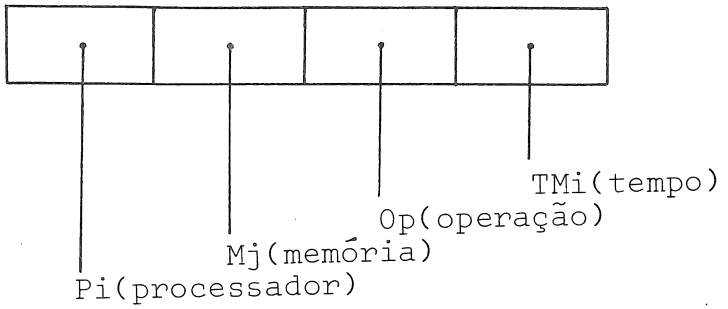


Figura 4 - Estrutura lógica de um PEDIDO

Área D2: fila de execução - acessada para escrita pelo processo ESTABELECER CONEXÃO; acessada para leitura pelo processo UTILIZAR CONEXÃO. Esta área foi estruturada como fila e para sua manipulação existem algoritmos de inserção e de retirada. A estrutura é a mesma da área D1. A estrutura do nodo, entretanto, será diferente: além dos campos descritos na figura 4 existe um novo campo, de ELO, necessário para a liberação do caminho após sua utilização; esta informação é criada internamente ao processo ESTABELECER CONEXÃO.

Área D3: fila de pedidos não atendidos - acesso para escrita pelo processo ESTABELECER CONEXÃO; acesso para leitura pelo processo PEDIDOS NÃO ATENDIDOS. Manipulada através de algoritmos de inserção e retirada em filas. É idêntica à área D1.

Área D5: estado da interconexão - esta área reflete a utilização dos barramentos durante uma simulação. É uma área protegida contra acesso simultâneo pelos dois processos que podem modificá-la: ESTABELECER CONEXÃO e UTILIZAR CONEXÃO. A área tem um papel dinâmico, refletindo a alocação e liberação de segmentos de barramento para o atendimento de pedidos.

Dentro do processo ESTABELECER CONEXÃO o acesso a D5 é feito pela estação MONTACAMINHO (ver funciograma F2). Esta estação procura segmentos livres de barramentos que estabelecem a ligação entre o processador Pl e memória Mj, especificados no pedido retirado da fila. Quando o caminho for estabelecido, os segmentos de barramentos utilizados estarão marcados como "não disponíveis" dentro da área D5.

Na estação UTILIZAR CONEXÃO o acesso a esta área é feito pela estação LIBERACAMINHO (funciograma F3) que libera todos os segmentos de barramentos utilizados para estabelecer o caminho Pi-Mj marcando-os como "disponíveis".

Uma única área não necessita exclusão mútua:

Área D4: descrição da interconexão - esta área é estática,

criada na inicialização do sistema e não mais modificada. Ela descreve a representação interna da interconexão, mostrando as ligações processadores-arbitradores-memórias. O primeiro passo na simulação do comportamento de uma interconexão será o de gerar a descrição interna desta interconexão. No futuro isto será feito por meio de uma LDH.

A interconexão é descrita como um grafo, onde os nodos representam os elementos processadores, arbitradores e memórias. As arestas representam seções de barramento. Os nodos adjacentes permitidos são: ligações processador-arbitrador, arbitrador-arbitrador e arbitrador-memória.

A representação deste grafo é feita por matriz de adjacência. Nas linhas são definidos os processadores, arbitradores e memórias, enquanto que nas colunas são definidos os arbitradores. Por exemplo, a estrutura mostrada na figura 5 é descrita pela matriz da figura 6.

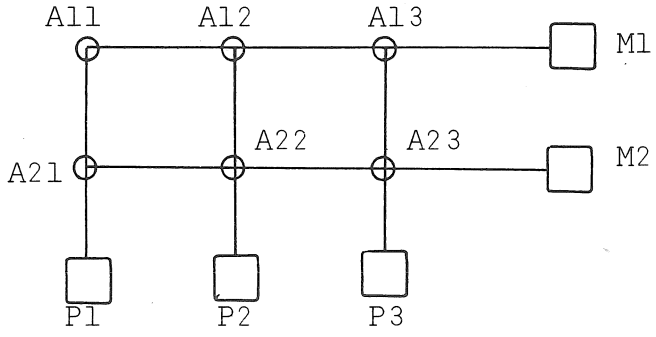


Figura 5 - Esquema de Interconexão

	P1	P2	P3	A11	A12	A13	A21	A22	A23	M1	M2
A11											
A12											
A13											
A21											
A22											
A23											

Figura 6 - Matriz de adjacência para a interconexão

A visão seqüencial do sistema simulador descrito no funciograma F0 pode ser obtida:

1. Um processador Pi gera um pedido onde está especificada uma memória e uma operação (escrita/leitura).

2. O pedido é inserido na área D1.

3. O processo ESTABELECER CONEXÃO retira o pedido da fila e tenta montar a conexão entre o processador e a memória especificados no pedido, utilizando informações das áreas D4 e D5.

4. Se o caminho é estabelecido, este é colocado na área D2.

5. Se o caminho não pode ser estabelecido, o pedido original será inserido na área D3.

6. O processo UTILIZA CONEXÃO retira um pedido com conexão já estabelecida, da área D2 e simula a operação esperando um tempo Ti antes de liberar o caminho, utilizando o parâmetro e, que aponta para a lista dos segmentos utilizados na conexão (atualização da área D5) e avisar o processador Pi.

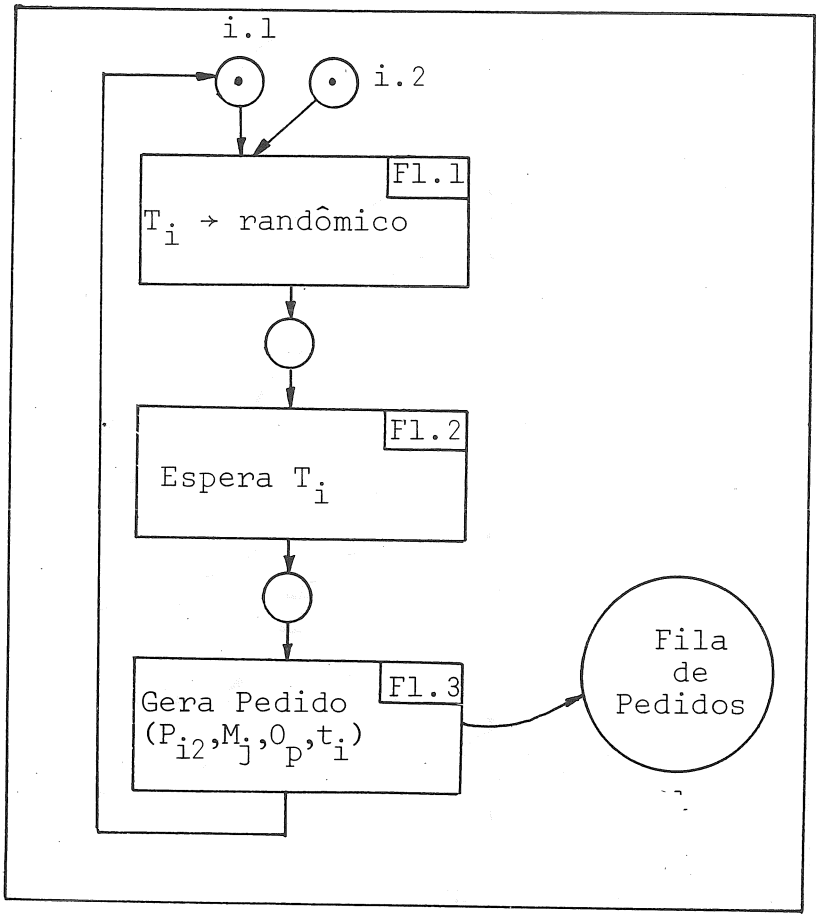
7. O processo PEDIDOS NÃO ATENDIDOS retira um pedido da área D3 e o insere novamente na área D1.

III.2 ESTRUTURA DE UM PROCESSADOR PI

Cada processador Pi do funciograma geral (F0) é representado por uma estrutura igual à descrita no funciograma F1.

Inicialmente ambos os canais de aviso I.1 e I.2 estão sinalizados, permitindo o disparo do processo. Após o disparo e execução das três estações internas, a estação "gera pedido"(F1.3) coloca a marca novamente no canal I.1, mas o processo não está habilitado até que a marca I.2 seja também colocada; isto será feito pela estação F3 do funciograma F0, após o atendimento do pedido.

A estação F1.1 coloca numa variável Ti um valor randômico; este valor será utilizado para simular tempo de processamento, isto é, o processador Pi, durante este tempo não envia nenhum pedido. Após esta espera o processador gera novo pedido de leitura/escrita na estação F1.3. O pedido tem a estrutura descrita na figura 4 gerando a quádrupla (Pi,Mj,Op,TMi). O TMi tem a função de simular a quantidade de dados sobre a qual a operação Op, leitura/escrita, é realizada. Na estação UTILIZAR CONEXÃO do funciograma F0 os segmentos de barramento utilizados ficarão marcados como "não disponíveis" durante este tempo TMi. Finalmente, inclui-se o pedido gerado na fila D1 do funciograma F0 de onde será retirado por ESTABELECER CONEXÃO.



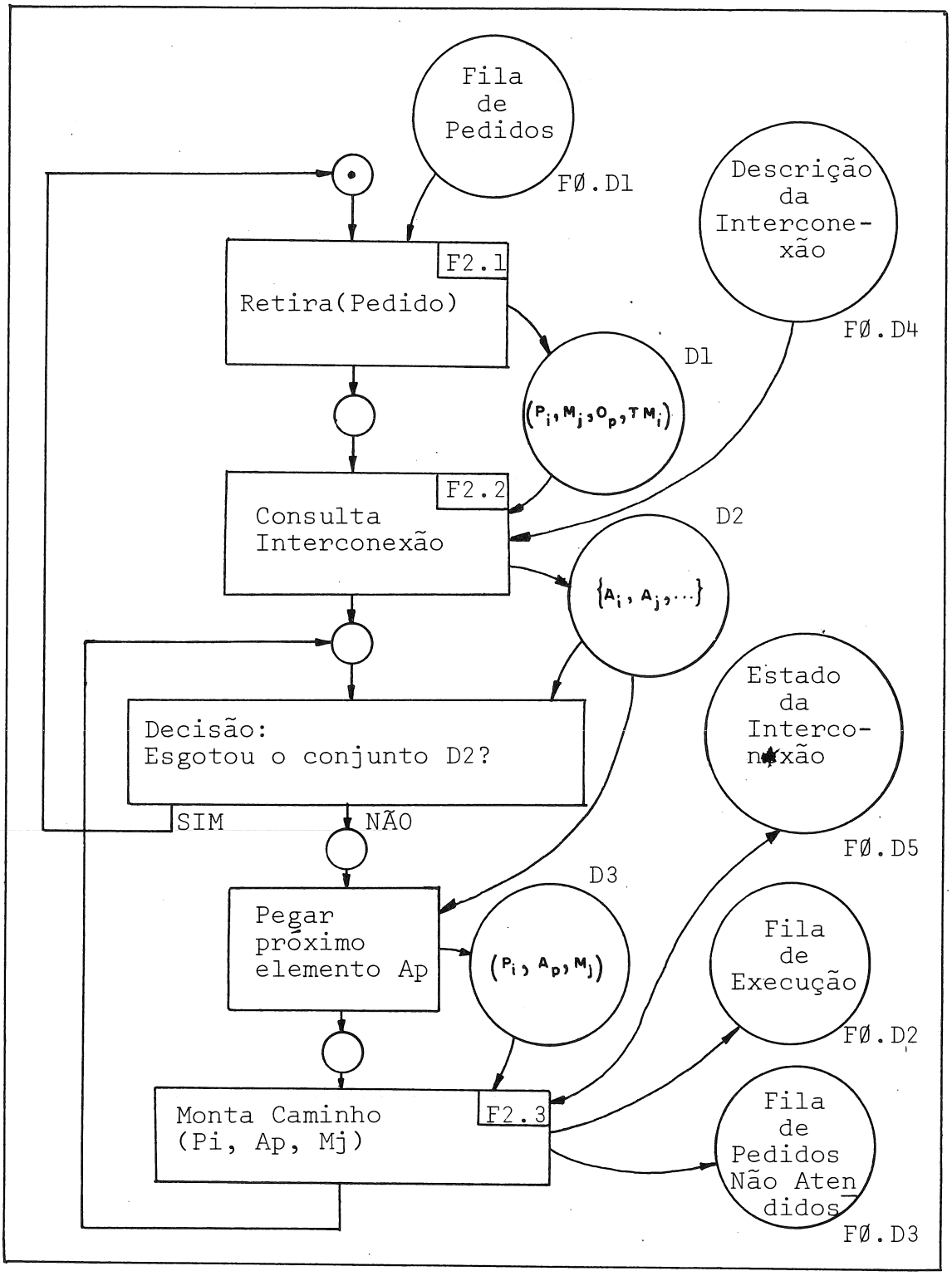
Funciograma F1

III.3 ESTABELECIMENTO DA CONEXÃO

O processo descrito no funciograma F2 é o responsável pela tentativa de estabelecimento do caminho Pi-Mj. Se esta for bem sucedida, o caminho montado será inserido na fila de execução. Caso contrário, o pedido original é colocado na fila de pedidos não atendidos.

Inicialmente a estação F2.1 está habilitada. Ela retira um pedido da área F0.D1 onde estão os pedidos gerados pelos processadores. A estação F2.2 - Consulta interconexão - verifica que arbitradores, ligados ao processador Pi, podem ser utilizados para estabelecer uma conexão entre Pi e Mj; o conjunto destes arbitradores é colocado na área D2, definindo um conjunto controlador de laço.

A estação F2.3 - Montacaminho - tenta estabelecer uma ligação entre Pi e Mj, utilizando arbitradores que tomarão suas decisões baseados nas informações de estado de segmentos de barramentos, descritos na área F0.D4. Como resultado é colocada na área F0.D2 a seqüência de segmentos de barramento utilizados para a montagem do caminho.



Funciograma F2

Na tentativa de montar Pi-Mj leva-se em conta o estado atual dos segmentos de barramento na interconexão (figura 7). Os parâmetros de entrada para montacaminho são: Pi,Ap,Mj. O parâ-

metro de saída é o ponteiro E (elo) para o cabeça de lista do caminho montado; este endereço será necessário para executar a liberação do barramento (ver F3).

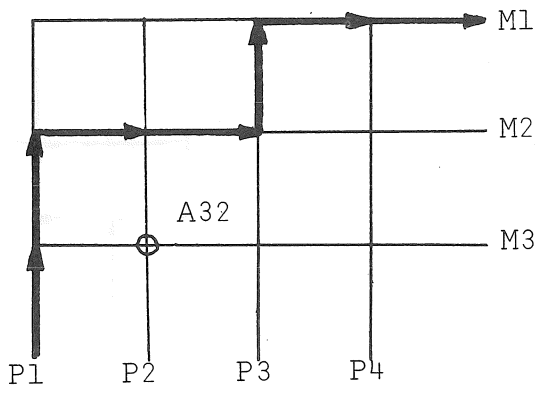
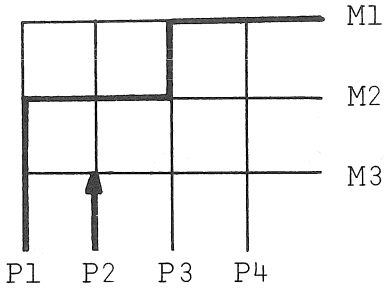


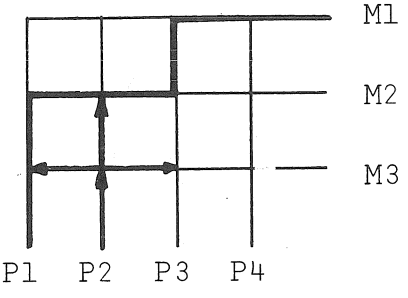
Figura 7 - Interconexão com P1-M1 estabelecido

Se for necessário estabelecer P2-M2, com a interconexão no estado descrito pela figura 7, MONTACAMINHO é ativada com os parâmetros P2, A32, M2 (figura 8a).

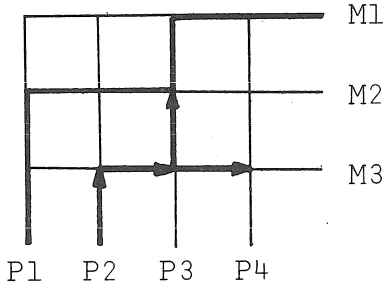
MONTACAMINHO verifica que segmento de barramento está livre, a partir de A32, o marca como "NÃO DISPONÍVEL" e continua o processo a partir do novo arbitrador Aij. O processo de busca do segmento livre pode ser realizado de diversas formas. Por



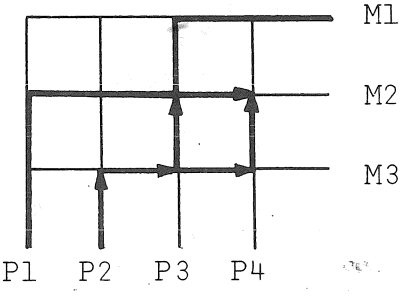
a)



b)



c)



d)

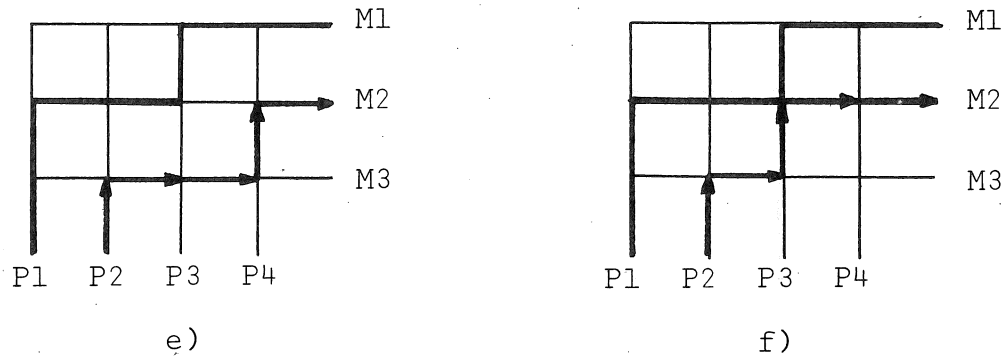


Figura 8 - Montagem do caminho P2-M2

exemplo, diversos caminhos podem ser testados em paralelo e a primeira tentativa bem sucedida desativa todos os outros processos, liberando os segmentos utilizados nas tentativas não sucedidas. Dentro desta filosofia, o processo iniciado na figura 8a teria a seqüência mostrada na figura 8b a 8f.

Observa-se que a partir da situação descrita na figura 8d pode se chegar tanto à situação (e) como à situação (f), na dependência do processo que primeiro tiver acesso ao segmento do barramento A24-M2, marcando-o como "não disponível" e eliminando portanto o recurso para o outro processo.

III.4 USO DA CONEXÃO

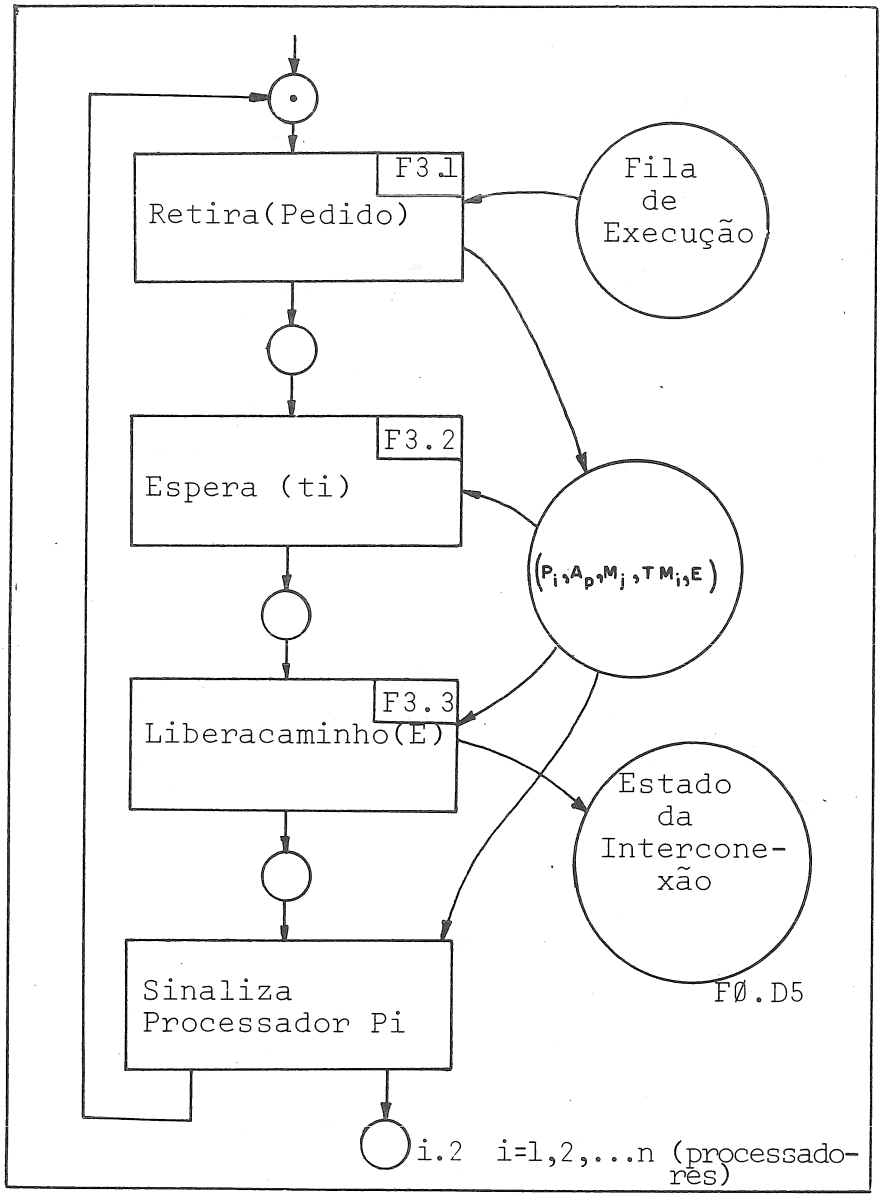
O processo descrito no funciograma F3 simula a utilização do caminho Pi-Mj.

Inicialmente é retirado um pedido da fila de execução; pode ocorrer underflow, fazendo com que o processo seja suspenso. No pedido retirado, o parâmetro TMi simula a utilização do caminho. Os segmentos de barramento que o compõem permanecem marcados como "não disponíveis" durante este tempo TMi.

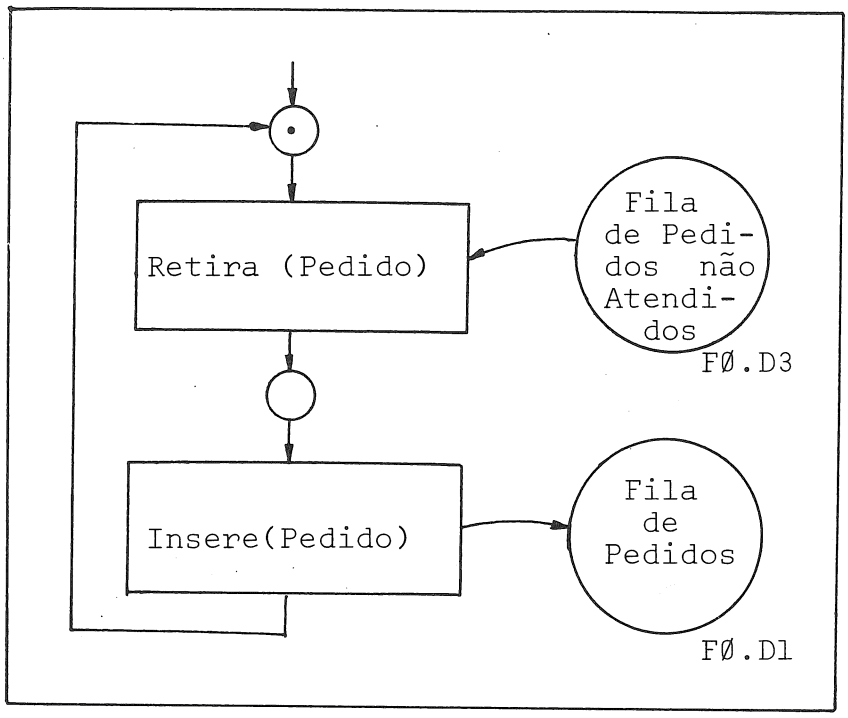
A estação F3.3 - Liberacaminho - utiliza o parâmetro E do pedido (Pi,Mj,Op,TMi,E) para marcar os segmentos de barramento como "disponíveis". Desde a geração do pedido pelo processador Pi até a liberação do caminho, o processador permanece inativo.

III.5 PEDIDOS NÃO ATENDIDOS

O processo F4 tem a função de retirar pedidos da fila F0.D3, que contém os pedidos que não foram atendidos pela estação montacaminho, e inseri-los novamente na fila F0.D1-fila de pedidos, para uma nova tentativa de estabelecer a conexão.



Funciograma F3



Funciograma F4

IV. CONCLUSÕES

O SSIP é uma ferramenta de grande flexibilidade para o arquiteto de sistemas de múltiplos processadores, permitindo o estudo e a avaliação de diferentes interconexões. Como tal, ele pode ser utilizado não só na análise de interconexões para multiprocessadores como também na de redes de computadores ou de sistemas distribuídos.

Atualmente o SSIP está sendo desenvolvido para a definição da arquitetura mais viável do projeto SMM. Após esta etapa a ferramenta será generalizada permitindo o seu uso em ensino e pesquisa. A experiência ganha nesta primeira etapa,fornecerá os subsídios para melhorar o SSIP e implementar uma LDH que, utilizando o núcleo básico definido, permitirá definir diferentes estruturas de interconexão e arbitradores para uma dada aplicação.

Não há dúvida que os sistemas de múltiplos processadores terão no futuro uma participação importante - e crescente - no mercado de máquinas. É portanto imprescindível o desenvolvimento de ferramentas tais como o SSIP que permitam o estudo aprofundado da organização de tais máquinas e testes conclusivos de desempenho relativo entre diversos esquemas de interconexão.

O SSIP está atualmente na etapa final de sua definição devendo entrar em fase de programação.

BIBLIOGRAFIA E REFERÊNCIAS

(And.75) ANDERSON, G.A. E JENSEN, E.D.
"Computer inter-connection structures: Taxonomy,Characteristics and examples", Computer Surveys, v7, n4, dec75, p197-213.

(Bar.75) BARBACCI, M.R.
"A Comparison of register transfer languages for describing Computer and digital systems", IEEE Transaction on Computer, feb75. p137-150.

(Ens.75) ENSLOW, P.H.
"Multiprocessor architecture - a survey", 1975 Sagamore Computer Conference on Parallel Processing, p63-70.

(Har.77) HARTENSTEIN, R.W.
"Fundamentals of structured hardware design". North Holland 1977.

(Kle.79) KLEIN, D.D.
"MMPS - a reconfigurable multi-microprocessor simulator system". Afips-ncc79, p199-203.

(Nil.71) NILSSON, N.J.
"Problems-solving methods in artificial intelligence". New York, McGraw-Hill, 1971.

(Ric.77) RICHTER, G.
"O sentido e o valor do banco de dados", Dados e idéias, n6, jun/jul77, p2-14.

(Thu.78) THURBER, K.J.
"Computer Communication Techniques", Computer architecture news, ACM-Sigarch, v7, n3, oct78, p7-16.

(Tri.79) TRIPATHI, A.R. & LIPOVSKI, G.J.
"Packet switching in banyan networks", 6th annual computer architecture, april 79, p160-167.