

O Sistema de Computação Paralela NCP I *

C.L. Amorim R. Citro A.F. Souza
E.M. Chaves Filho
Núcleo de Computação Paralela
COPPE/Sistemas
Universidade Federal do Rio de Janeiro
C.P. 68511 Rio de Janeiro RJ 21945-970 Brasil
e.mail; amorim@rio.cos.ufrj.br

Resumo

Neste trabalho o computador paralelo de alto desempenho NCP I em desenvolvimento na COPPE/ UFRJ é apresentado. A principal característica de sua arquitetura é de ela oferecer suporte tanto ao modelo de memória privativa como ao de memória compartilhada.

Uma implementação limitada dessa arquitetura tem estado em operação ininterrupta nos últimos três anos e utilizada com sucesso numa grande variedade de aplicações em áreas de Engenharia e de Ciência da Computação.

Nós discutimos as principais características arquiteturais e a atual implementação de hardware baseado no transputer T800 e no Intel i860, assim como o software em desenvolvimento. Resultados preliminares de desempenho obtidos são analisados e conclusões extraídas.

Abstract

In this paper we introduce the NCP I, a high performance parallel computer under development at COPPE/ Federal University of Rio de Janeiro. A distinguishing feature of the NCP I architecture is that it supports the private memory model as well as the shared memory model.

A limited implementation of the NCP I has been successfully used in the past three years in a large variety of applications in Engineering and Computer Science.

We discuss the NCP I main architectural characteristics and its current hardware implementation based on both the transputer T800 and the Intel i860 as well as software been developed. Preliminary performance results are analysed and conclusions are drawn.

*Este projeto é parcialmente apoiado pela FINEP (Processo No. 5288062200)

1 Introdução

Nos últimos anos, a controvérsia sobre computadores MIMD de memória compartilhada versus memória privativa tem sido debatida na literatura e estamos distantes de uma resposta definitiva. Em geral, a maioria dos pesquisadores concordam que computadores paralelos de memória compartilhada colocam a maior parte da tarefa do processamento paralelo no projetista de hardware [1,2,3]. Eles são mais fáceis de programar embora a depuração de programas é mais difícil. Por outro lado, a expansibilidade arquitetural é uma das principais características de computadores paralelos de memória privativa, enquanto que multiprogramação e memória virtual têm sido virtudes mais relacionadas com projetos de memória compartilhada. Acima de tudo, existem questões mais complexas para serem enfrentadas pelos arquitetos de computadores paralelos tais como transportabilidade e a atual escassez de aplicações altamente paralelas [4,5,6].

Sob o ponto de vista do usuário, dado o *estado-de-arte* da tecnologia de computação paralela, é muito desejável que um computador paralelo dê suporte a ambos paradigmas de programação. Tal requisito deve ser considerado não só no nível de hardware por razões de custo e desempenho mas também no nível de software para permitir que programas sejam executados em ambos os modelos ou mesmo numa mistura dos dois. Tal flexibilidade arquitetural seria especialmente adequada a novas aplicações paralelas nas quais é difícil julgar antecipadamente a eficiência relativa das duas classes de computadores paralelos.

Mais ainda, é uma vantagem para o usuário se um computador paralelo consiga processar eficientemente uma mistura de programas das duas classes de aplicações em vez de ele ter que lidar com dois computadores paralelos separadamente ou ter de escolher um tipo de computador paralelo em detrimento do outro.

Portanto, se um computador paralelo é capaz de dar suporte a ambos os modelos eficientemente e talvez transparentemente, nós podemos dispor de um computador MIMD de propósito geral que explore o melhor dos dois modelos sem dobrar o custo e mais importante, a controvérsia acima torna-se menos crítica.

Vamos considerar agora uma importante área para a qual os projetistas de computadores paralelos têm dado muita atenção e para a qual os supercomputadores têm sido construídos: as aplicações científicas e de engenharia em grande escala.

Uma razão principal do porquê é que supercomputadores tais como CRAY XMP, CRAY YMP, Fujitsu VP e NEC SX [7] são altamente bem sucedidos é que eles dão suporte eficiente às duas classes de problemas numéricos de grande escala, ou seja, os relacionados aos métodos do contínuo e aos de partículas finitas [8,9], cada um exigindo uma capacidade computacional bastante distinta.

No modelo do contínuo, as estruturas de dados são mais regulares e envolvem tipicamente sistemas tridiagonais, grandes matrizes densas assimétricas e grandes tabelas de dados experimentais. Já o modelo de partículas é centrado em simulações Monte Carlo no qual uma vasta quantidade de memória é necessária para armazenar informações sobre a evolução de história do sistema dinâmico de partículas e detalhes de colisões.

Além de uma hierarquia de memória, os supercomputadores fazem uso de um conjunto de instruções vetoriais regulares e de instruções vetoriais do tipo agrega/espalha com vetores de controle/mascaramento para satisfazer eficientemente ambos os modelos. Na prática, devido a maior regularidade de suas estrutura de dados, os supercomputadores vetoriais são mais eficientes nas aplicações da classe de modelos contínuos.

Conseqüentemente, se os futuros computadores paralelos forem suplantados pelos processadores vetoriais pipelined, eles têm de alcançar alto desempenho em ambas as classes de problemas numéricos. A esse respeito, nós notamos que embora resultados promissores têm sido obtidos por alguns grupos utilizando computadores paralelos de memória privativa, pioneiramente na CALTECH [10], ou computadores paralelos de memória compartilhada, é improvável que ocorra uma ampla utilização deles nos próximos anos, a menos que um computador paralelo MIMD de propósito geral se torne disponível.

Nossa posição é que um computador paralelo MIMD deve dar suporte arquitetural a ambos os modelos de programação. Tal computador paralelo deve tirar vantagem das melhores características dos dois modelos incluindo expansibilidade, memória virtual, multiprogramação, facilidade de programação e de depuração. A concepção do projeto deve manter um balanceamento arquitetural entre os dois modos de processamento dependendo dos avanços da tecnologia de semicondutores em comunicação, processador, memória e dispositivos de entrada/saída.

Para explorar ao máximo as características do hardware, a interface com o sistema operacional deve prover uma visão integrada de forma que os compiladores, bibliotecas e ambientes de programação possam oferecer ao usuário um computador paralelo simples e eficiente.

Este artigo é organizado como se segue. Na Seção 2, nós descrevemos as características arquiteturais do NCP I, em seguida nós discutimos um modelo MIMD integrado para o NCP I e a nossa implementação atual. Na Seção 3, nós apresentamos o hardware, o software e aplicações em desenvolvimento no NCP I. Na Seção 4, resultados preliminares de desempenho são analisados. Finalmente, conclusões são apresentadas.

2 Os Modelos de Computação Paralela NCP I

O NCP I é um computador paralelo de alto desempenho orientado para o processamento em grande escala encontrado em aplicações científicas e de engenharia, incluindo Exploração Sísmica, Meteorologia, Oceanografia e Dinâmica dos Fluidos Computacional. Uma ilustração do computador NCP I com 16 nós de processamento é mostrada na Figura 1. O NCP I implementa o paradigma de memória privativa na metade superior da Figura 1 e o paradigma de memória compartilhada na metade inferior.

2.1 A Arquitetura do NCP I

A arquitetura do NCP I com 16 nós é mostrada na Figura 2. Em cada nó de processamento (NP), existem dois elementos de processamento. Um é o elemento de proces-

samento de memória privativa (EMP) que é responsável pela comunicação entre nós e pela carga computacional a ele atribuída. O segundo é o elemento de processamento de memória compartilhada (EMC) que divide a carga computacional com o EMP e controla o acesso do nó ao disco.

A estrutura da memória do NCP I é dividida em níveis interno e externo com diferentes tempos de acesso e espaços de endereçamento. No primeiro nível, internamente a cada nó, a memória é composta pela memória privativa do EMP, um buffer rápido de mensagens e a memória compartilhada entre os dois elementos de processamento. No nível externo, a memória centralizada é compartilhada com todos os EMC's e o computador hospedeiro. A Figura 3 ilustra o espaço de endereçamento da memória do NCP I visto pelos elementos de processamento.

A estrutura de comunicação do NCP I é organizada para lidar com três tipos de demanda de comunicação durante a execução de processos paralelos. O barramento rápido do sistema e de E/S dá suporte às transferências entre processadores e à memória global e ao tráfego intenso de blocos de dados que podem ser produzidos por processos de E/S. O buffer rápido de mensagens implementa um mecanismo de comunicação para troca rápida de mensagens entre o EMP e o EMC. A rede de comunicação entre nós (RCN) oferece possibilidade de troca rápida de mensagens entre os nós de processamento.

O computador hospedeiro é interligado aos NPs através do barramento do sistema e também através da RCN. O barramento do sistema fornece comunicação entre o hospedeiro e os EMCs enquanto que o RCN provê comunicação paralela entre o hospedeiro e os EMPs.

A arquitetura do NCP I é expansível através da replicação de clusters. A RCN torna a expansão mais fácil do que o barramento do sistema. Este não consegue operar mais do que dezesseis nós sem degradar o desempenho do cluster. Portanto, cada cluster tem de ter um barramento e um chaveamento entre barramentos. Cada cluster pode ter seu próprio controlador de disco, embora o hospedeiro e a interface da rede de comunicação remota (ex. Ethernet) devem ser localizados no cluster hospedeiro. Teoricamente, o NCP I pode ter 128 clusters.

2.2 Em Direção a um Modelo MIMD Integrado

A estrutura do computador NCP I descrita é projetada para lidar com as duas abordagens para a computação MIMD. Para isso, nós podemos classificar as tarefas paralelas de acordo com suas demandas sobre os recursos do sistema, como se segue:

- **Tarefas Computacionais:** se referem a tarefas intensivas em computação, pouca comunicação e pouco tráfego de E/S.
- **Tarefas de Comunicação:** nós definimos como sendo tarefas com comunicação intensiva, pouca computação e pouco tráfego de E/S.

- **Tarefas de Entrada/Saída:** elas correspondem a tarefas intensivas em E/S, pouca comunicação e computação (ex. tarefas de gerências de arquivos e de memória virtual).
- **Tarefas de Controle e de Sincronização:** tipicamente, tarefas para inicializar recursos do sistema, escalonadores de tarefas e controle de acesso a recursos compartilhados.

As colunas 2 e 3 da tabela 1 mostram a atribuição dos diferentes tipos de tarefa segundo a visão mestre-escravo embutida nos modelos de memória privativa e de memória compartilhada. No modelo de memória privativa, o nó de processamento está sob o controle mestre do EMP. Neste caso, o EMP executa as tarefas de comunicação e compartilha a carga computacional com o EMC que também executa as tarefas de E/S. No modelo de memória compartilhada, a diferença é simplesmente que a tarefa de controle mestre é agora assumida pelo EMC. O modelo integrado do NCP I é apresentado na próxima coluna da tabela 1. Como pode ser visto, nesse modelo, o papel de ambos somente difere na atribuição das tarefas de comunicação e de E/S que agora pertencem ao EMP e EMC, respectivamente. Os outros dois tipos de tarefas podem ser dinamicamente alocados a eles em função de suas capacidades computacionais.

O software básico (ex. compiladores e sistemas operacionais) podem explorar as melhores características de cada um dos dois modelos de modo a minimizar overheads introduzidos pela computação paralela tais como comunicação, sincronização e complexidades de programação e depuração. Por exemplo, enquanto o desenvolvimento de programas é executado no ambiente de memória compartilhada, a depuração do programa pode ser feita no ambiente de memória privativa, obviamente novas ferramentas de software são necessárias para mudar de ambiente. Tarefas de granularidade fina e de muita sincronização deve ser atribuída ao EMC enquanto que outras tarefas poderiam ser alocadas a qualquer um dos dois elementos.

Naturalmente, o sistema operacional, os compiladores e as bibliotecas têm de ser orientados para o modelo integrado do NCP I. Uma solução imediata é adaptar um sistema operacional EMC ou EMP para fornecer facilidades do outro modelo. Assim, um S.O. para EMC deve ser estendido para prover serviços tais como roteamento de mensagens e topologias de comunicação padrões enquanto que um S.O. para EMP deve ser estendido para oferecer um sistema de memória virtual [11].

2.3 O Modelo Paralelo Atual

Como o projeto e a implementação de um sistema operacional para a nova estrutura computacional estava além do tempo previsto do projeto, foi decidido primeiro se implementar uma versão limitada do NCP I. Devido a maior simplicidade do projeto de hardware, nós iniciamos, segundo a figura 1, a implementação de cima para baixo, i.e., do EMP para o EMC.

Para avaliar a estrutura do hardware do NCP I, nós escolhemos um modelo computacional de forma que o EMC pudesse desempenhar o papel de um processador pipelined numérico. Esta implementação segue no nível do nó, os princípios básicos para a computação de alto desempenho baseada no êxito da experiência do CRAY 1 e seus sucessores [12]. Claramente, isto deve ser considerado no nível de modelo. Nossa atual implementação do NCP I usa microprocessadores comerciais de alto desempenho, i.e., o transputer T800 [13] como o EMP e o Intel i860 [14] como o EMC para simular um modelo de processamento vetorial.

De acordo com esse modelo, no nível arquitetural do nó de processamento, as instruções escalares e de controle são atribuídas, principalmente ao EMP enquanto que instruções vetoriais são simuladas no EMC. Registradores vetoriais são simulados no buffer rápido de mensagens e vetores longos são mantidos na memória compartilhada.

No nível básico de programação, o NCP I foi influenciado pela tecnologia transputer e sua linguagem de programação paralela OCCAM [15]. Processos concorrentes podem ser atribuídos tanto ao EMP como ao EMC, sendo que para este um núcleo para prover o modelo OCCAM de processos foi implementado [16]. Assim, a carga computacional pode ser compartilhada com os dois processadores, com processos de granularidades grossa e média sendo atribuídos aos EMC e EMP, respectivamente.

Enquanto o processamento pipelined é fornecido no nível do nó, o processamento paralelo está sendo implementado através dos nós via a RCN, via o barramento do sistema ou ambos.

3 A Implementação do Sistema NCP I

3.1 O Hardware

O sistema NCP I é um computador MIMD de memória privativa/compartilhada. O sistema consiste de nós de processamento idênticos aos quais cada nó contém dois módulos:

- O módulo do Elemento de Memória Privativa (EMP) que inclui um microprocessador transputer T805-20, 4 Mbytes de memória, 4 elos com a rede de comunicação, 64 Kb EPROM, registradores de comando e estado e uma interface com o EMC.
- O módulo do Elemento de Memória Compartilhada (EMC) inclui um microprocessador Intel i860 XR (33 MHz), 8 Mbytes de memória 3-PORT, 16 Kbytes de memória estática dual-port, memória de paridade, 64 Kb EPROM, interface com o EMP e com o barramento de E/S e do sistema.

A figura 4 mostra a estrutura do hardware de um nó de processamento. A interface com o hospedeiro, um IBM PC-AT compatível, é feita através da placa interface localizada num slot do PC que serializa o dado paralelo proveniente do barramento PC.

O modelo de memória compartilhada do NCP I é baseado em barramento. Devido à natureza experimental e às restrições de orçamento, optamos pela utilização do barramento comercial VME [17] como via de acesso para E/S e para o sistema.

Atualmente, o protótipo com 16 nós possui as seguintes características:

- Desempenho de pico de 800 MIPS (RISC) e 960 MFLOPS (precisão dupla).
- 64 Mbytes de memória privativa e 128 Mbytes de memória compartilhada.
- taxa de entrada/saída de 40 Mbytes/segundo.
- Rede de comunicação entre os nós com *throughput* de 160 Mbytes/segundo.

Dada a natureza experimental da atual implementação do NCP I, nós decidimos minimizar o tempo de projeto e o custo em vez de adotarmos soluções sofisticadas para atingir balanceamento arquitetural. Por exemplo, os potentes microprocessadores utilizados no NCP I produzem 800 MIPS de desempenho de pico, em contraste com a banda passante agregada de comunicação de somente 200 Mbytes/segundo. Nós esperamos isolar os efeitos da tecnologia no desempenho ao projetar o software básico, de maneira que versões futuras do NCP I possam explorar melhor novos dispositivos lógicos.

Paralelamente ao desenvolvimento do módulo EMC, foi instalado um processador de E/S de alta velocidade alternativo baseado em transputer e obedecendo o padrão SCSI, com 600 Mbytes de disco. Essa solução permitiu evitar o baixo desempenho de acesso ao disco através do computador hospedeiro IBM-PC.

O gabinete do NCP I comporta até 16 nós de processamento mais a fonte de alimentação, disco, ventiladores e cablagem, medindo 56 cm x 52 cm x 73 cm. Maiores detalhes do projeto encontram-se em [18].

3.2 Pesquisa e Desenvolvimento em Software

A maior motivação para a atual estratégia de implementação do hardware do NCP I foi a de se poder utilizar inicialmente ambientes de programação paralela comerciais disponíveis para transputers¹. O sistema operacional Helios [19] do tipo Unix distribuído foi selecionado assim como compiladores C e Fortran [20] e o ambiente STRAND [21], instalados no NCP I. Recentemente, foi instalado software para interconectar o NCP I à Rede Internet, através de protocolo TCP-IP.

Dessa forma, os demais pesquisadores conseguiram rapidamente adquirir experiência em programação paralela e concentrar seus esforços em objetivos de médio prazo relacionados à tecnologia de construção de programas paralelos tais como: linguagens [22], compiladores [23,24], paralelizadores [25], balanceamento de carga [26], ambientes e ferramentas [27,28,29] para programação paralela assim como algoritmos paralelos [30] e aplicações [31,32]. Uma documentação mais completa das publicações encontra-se em [33].

¹enquanto se desenvolvia ambiente para o i860

4 Resultados Preliminares

O NCP I tem sido utilizado intensamente como veículo de ensino, pesquisa e desenvolvimento em computação paralela. Além disso, em [34] foram feitas comparações de desempenho com sistemas comerciais numa aplicação real.

Esses experimentos mostraram que:

1. O desempenho do microprocessador Intel i860 é cerca de sete vezes superior ao do transputer T805 em tarefas computacionais e é equivalente ao do T805 em tarefas de comunicação e de entrada e saída.
2. O desempenho do processador de E/S é equivalente ao de máquinas paralelas comerciais (ex. Intel iPSC).
3. O desempenho do nó de processamento do NCP I pode ser melhorado bastante se a estrutura interna dos microprocessadores T800 e i860 forem efetivamente exploradas pelo programador e/ou compilador. Por exemplo, no transputer com a utilização da RAM interna (4 Kbytes) e no i860 com a utilização do modo dual de processamento e dos pipelines aritméticos.

Esses resultados sugerem que todas as tarefas computacionalmente intensivas e de sincronização sejam atribuídas ao i860 e as de E/S e de comunicação ao T805 para essa classe de aplicações.

Outra consequência importante foi para a evolução do projeto arquitetural do NCP I em direção a uma máquina MIMD de propósito geral. Os experimentos revelaram que é viável a implementação de um sistema de memória virtual (SMV) paginada para o NCP I. Nesse caso, os EMPs (T805) dariam o suporte necessário de um SMV para ser utilizado pelo EMC (i860). Estudos nessa direção estão sendo conduzidos.

5 Conclusões

O computador paralelo de alto desempenho NCP I representa um esforço na direção de um computador MIMD de propósito geral no qual ambos os modelos de programação paralela de memória privativa e de memória compartilhada, possam ser eficientemente integrados e utilizados transparentemente.

Uma implementação limitada do NCP I utilizando oito nós de processamento segundo um modelo de memória distribuída, tem sido utilizada ininterruptamente e com sucesso nos últimos três anos numa grande e variada quantidade de aplicações e com desempenho equivalente a sistemas comerciais, baseados em transputers e/ou em Intel i860. Uma versão otimizada de protótipo está sendo montada para ser colocada na rede internet à disposição de outros grupos de pesquisa em processamento paralelo, o que permitirá uma avaliação ainda mais ampla do protótipo.

6 Agradecimentos

Nós gostaríamos de agradecer aos outros membros do projeto NCP I: Valmir Barbosa, Edil Fernandes, Leila Eizirik e Felipe França, pelas contribuições ao trabalho aqui descrito. Elas encontram-se documentadas nas referências. Nós somos gratos aos Diretores da COPPE, Prof. Luiz Pinguelli Rosa e Prof. Nelson Maculan pelo suporte dado durante o desenvolvimento deste projeto. O Projeto NCP I foi muito ajudado pelo grupo de alunos pós-graduados e graduados: Cristiana Bentes, Gilberto Yamashiro, Júlio Barros, José Queiróz, Marta Andrade, Delfim Xavier, Emerson Ferreira, Juliana Serrano do Carmo e os técnicos em eletrônica, Eduardo S. Pereira e Adilson Magalhães.

7 Referências

- [1] G.S. Almasi and A. Gottlieb, *Highly Parallel Computing*, The Benjamin/Cummings Publishing Co., Chapter 10, 1989.
- [2] A.H. Karp, *Programming for Parallelism*, IEEE Computer 20, pp. 43-57, May 1987.
- [3] D.A. Patterson and J.L. Hennessy, *Computer Architecture a Quantitative Approach*, Morgan Kaufmann Publ. Inc., Chapter 10, 1990.
- [4] D.J. Kuck et alli, *Parallel Supercomputing Today and the CEDAR Approach*, Experimental Parallel Computing Architectures, J.J. Dongarra (Editor), Elsevier Science Pub-B.V. (North-Holland), pp. 1-23, 1987.
- [5] G.F. Pfister et alli, *The IBM Research Parallel Processor Prototype (RP3)*, Proceedings of the 1985, Int. Conf. on Parallel Processing, August 1985.
- [6] M. Annaratone, M. Fillo, K. Kakabayashi and M. Veredaz, *The K2 Parallel Processor: Architecture and Hardware Implementation*, K2 Project - 1989 Project Report, Integrated System Lab., Swiss Federal Institute of Technology.
- [7] C. Lazou, *Supercomputers and Their Use*, Oxford University Press, NY, 1986.
- [8] G. Rodrigue, E.D. Giroux and M. Pratt, *Large-Scale Scientific Computation*, IEEE Computer, pp. 65-80, October 1980.
- [9] H.S. Stone, *High Performance Computer Architecture*, Addison-Wesley Pub. Co., Chapters 4 and 5, 1987.
- [10] G.C. Fox, *The Hypercube as a Supercomputer*, Proc. of Second Int. Conf. Supercomputing, L.P. Kartashev and S.I. Kartashev (Eds.) Vol. 1, pp.181-194, 1987.

- [11] K. Li and R. Schaefer, *A Hypercube Shared Virtual Memory System*, Proc. 1989 Int. Conf. on Parallel Processing, pp. 1125-132, August 1989.
- [12] C.L. Amorim, *Simulated Performance of a Class of Vector Processors*, Proceedings of the Second International Conference on Supercomputing, L.P. Kartashev and S.I. Kartashev (Eds.), Vol. 3, pp. 350-358, 1987.
- [13] *IMST800 Transputer Preliminary Data Sheet*, Inmos Limited, England, March 1988.
- [14] *i860 64-Bit Microprocessor Data Sheet*, Intel Corporation, USA, October 1989.
- [15] D. May, *OCCAM*, ACM SIGPLAN Notices, 18(4), pp. 69-79, 1983.
- [16] Seidel, C.B., *Ambiente para Programação Paralela no Processador i860*, Tese de M.Sc., COPPE/Sistemas, Abril de 1993.
- [17] *VMEbus Specification Manual Rev. B*, Signetics, USA, 1984.
- [18] C.L. Amorim et alli, *The NCP I Parallel Computer*, COPPE/Sistemas, Relatório Técnico, ES-241-1991, Abril de 1991.
- [19] Helios Group, *The Helios Operating System*, Distributed Software Ltd., version 1, rel. 1.0, August 1988.
- [20] 3L Ltd., *Parallel C and Parallel Fortran User's Guides*, 1988.
- [21] Artificial Intelligence Ltd., *STRAND User Manual*, Buckingham Release, June 1990.
- [22] Sales, C.L. et alli, *Uma Linguagem Intermediária para Compilar ACTUS II em OCCAM 2*, IX Congresso da SBC, Uberlândia, MG, Julho de 1989, pp. 171-186.
- [23] Favre, L.E. et alli, *Um Compilador para a Linguagem Híbrida de Programação Paralela C-ACTUS*, submetido para publicação.
- [24] Queiroz, J.D., *Projeto e Desenvolvimento de um Compilador OCCAM II para o Ambiente NCP I/i860*, Projeto de Final de Curso, Departamento da Ciência da Computação, IM, UFRJ, 1993.
- [25] Maciel, P.M.C.P.F. e Amorim, C.L., *Otimização de Programas ACTUS*, III Simpósio Brasileiro de Arquitetura de Computadores, Rio de Janeiro, Novembro de 1990, pp. 263-281.
- [26] Canaley, L.M.M.B., *Alocação de Tarefas em Sistemas Distribuídos*, Tese de M.Sc., COPPE/Sistemas, Maio de 1993.

- [27] Castro, M.C.S. e Amorim, C.L., *Uma Biblioteca de Operações Vetoriais e Matriciais para um Multiprocessador Hipercúbico baseado em Transputers*, IX Congresso da SBC, Uberlândia, MG, Julho de 1989, pp. 458-470.
- [28] V.C. Barbosa, L.M.A. Drummond and A.L.H. Hellmuth, *An Integrated Software Environment for Large-Scale Occam Programming*, Proc. EUROMICRO'91, Viena, Austria, September 1991.
- [29] Albuquerque, A.T.C., *Um mecanismo de Reexecução Determinística de Programas Paralelos*, Tese de M.Sc., COPPE/Sistemas, Agosto de 1992.
- [30] L.P. Freitas and V.C. Barbosa, *Experiments in Parallel Heuristics Search (Extended Abstract)*, Proc. Int. Conf. Parallel Processing, Chicago, USA, August 1991.
- [31] V.C. Barbosa and P.M.V. Lima, *On the Distributed Parallel Simulation of Hopfield's Neural Networks*, Software - Practice and Experience, 20, pp. 967-983, October 1990.
- [32] Mattoso, M.L.Q. e Amorim, C.L., *Uma Experiência na Implementação de Operadores de Álgebra Relacional no Computador Paralelo NCP I*, Anais VI Simp. Bras. Banco de Dados, Manaus, Maio de 1991.
- [33] *Núcleo de Computação Paralela*, Relatório de Atividades, 1989-1992, COPPE/UFRJ.
- [34] Castro, M.C.S. e Amorim, C.L., *Uma Avaliação Comparativa de Desempenho do Computador Paralelo NCP I8 e um Processamento Sísmico*, submetido para publicação.

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

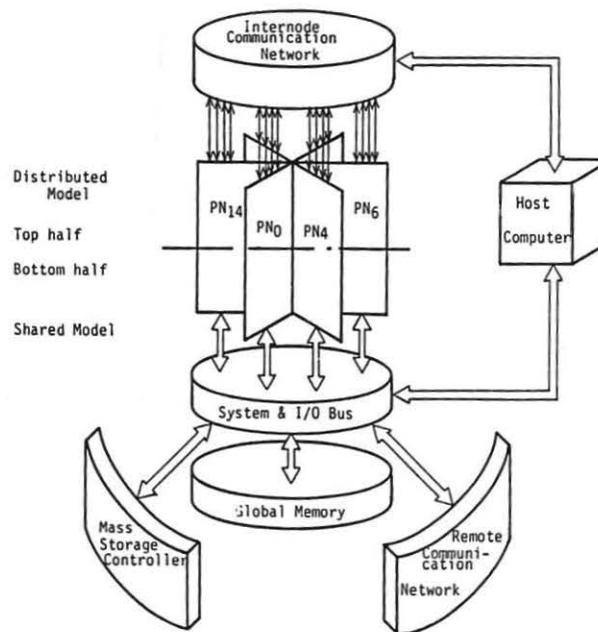


Figure 1. A Spatial View of the NCP I Parallel Computer

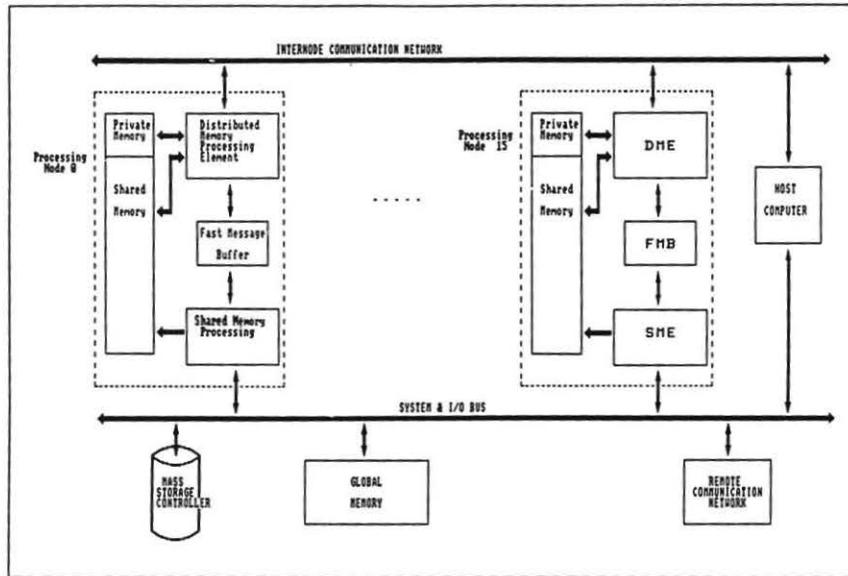


Figure 2: Block Diagram of the 16-node NCP-1 Architecture

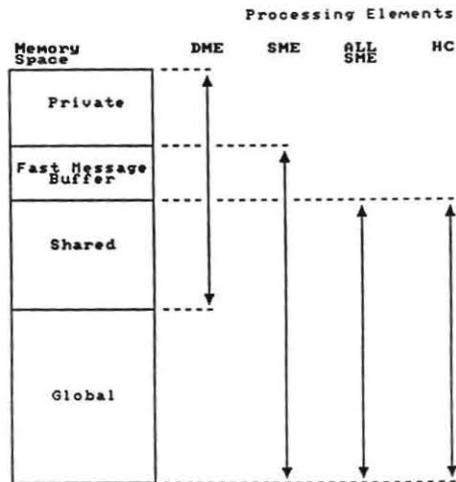


Figure 3: PE Memory Addressing Space

Table 1
PE Task Assignment and Parallel Computing Model

Processing Tasks	Parallel Model	Distributed Memory Model		Shared Memory Model		NCPI Integrated Model	
		DME	SME	DME	SME	DME	SME
Computational		Yes	Yes	Yes	Yes	Yes	Yes
Communication		Yes	No	Yes	No	Yes	No
Input/Output		No	Yes	No	Yes	No	Yes
Master Control		Yes	No	No	Yes	Yes	Yes

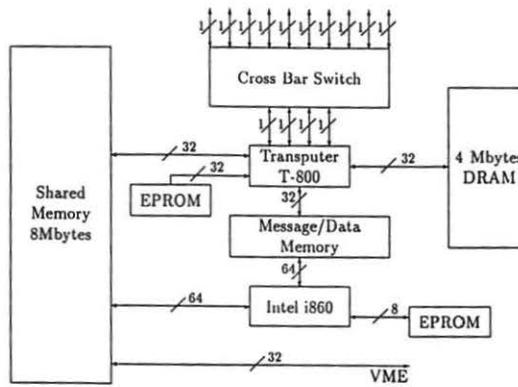


Figure 4: Single Node Hardware Structure