

C. L. AMORIM *

SUMÁRIO

A arquitetura dos processadores vetoriais pipelined (PVPs), popularmente referidos como supercomputadores, é colocada em perspectiva através de uma análise de sua estrutura e da interação de seus elementos básicos: 'pipelines' aritméticos, memória principal, memórias locais, 'pipelines' de instrução e instruções vetoriais. Os processadores vetoriais da Cray Research (CRAY1, CRAY X-MP), Control Data (CYBER 205) e Fujitsu (VP-200), atuais representantes do estado-da-arte são utilizados para consubstanciar a análise.

Os fatores que influenciam o desempenho desta classe de arquitetura assim como as principais linhas de pesquisa e desenvolvimento são apresentadas.

ABSTRACT

The architecture of pipelined vector processors (PVPs), usually referred to as supercomputers, is placed in perspective based on an analysis of the structure and interaction between its basic elements: arithmetic pipelines, main memory, local memories, instruction pipelines and vector instructions. The state-of-art vector processors from Cray Research (CRAY1, CRAY X-MP), Control Data (CYBER 205) and Fujitsu (VP-200) are used to illustrate and support the analysis.

The factors which influence the performance of such a class of processor architecture as well as the main research and development areas are introduced.

* Engenheiro Eletricista (PUC/RJ-1974), MSc (COPPE-1979) PhD (Imperial College-1984); Arquitetura de processadores pipelined e paralelos: modelagem, simulação e avaliação de desempenho; Professor Adjunto da UFRJ; COPPE/Sistemas, UFRJ, Caixa Postal 68511, CEP 21941, Rio de Janeiro, RJ.

Os supercomputadores que podem ser definidos como os computadores de maior desempenho numa determinada época, tiveram historicamente marcos bem distintos. Durante os anos 60, a competição entre a IBM e a Control Data produziram os excepcionais processadores escalares IBM 360/195 (uma síntese dos IBM 360/91 e IBM 360/85) e CDC 7600 (evolução do inovador CDC 6600). No início dos anos 70, sem a participação da IBM, surgiram os processadores vetoriais pipelined da Texas Instruments (TI ASC) e Control Data CDC STAR 100. Essa primeira geração de processadores vetoriais, com poucas máquinas produzidas, serviu para identificar principalmente os problemas relacionados com o processamento vetorial. Com o lançamento do CRAY1 pela Cray Research em 1976, iniciando a segunda geração dos processadores vetoriais ^[1], o mercado dos supercomputadores recebeu um novo impulso (em 1982 ± 70 CRAY1 haviam sido vendidos). A Control Data a partir da experiência do STAR 100 desenvolveu o CYBER 203 e em 1981 produziu o primeiro CYBER 205 para competir com o CRAY1. Em 1984, a Fujitsu com os processadores vetoriais VP-100 e VP-200 entrou fortemente no mercado ^[2].

Um fato ainda mais importante foi a recente criação pelo governo americano de vários centros espalhados pelo país dispondo de supercomputadores ^[3] ^[4]. Tal medida resultou da constatação de que tanto a qualidade da pesquisa americana como a competitividade de sua indústria estarão comprometidas se não se fizer frente às políticas nacionais da Alemanha, França, Inglaterra e Japão que vêm estimulando intensamente a exploração do potencial dos supercomputadores, colocando-os disponíveis aos pesquisadores de suas universidades e centro de pesquisas.

Acredita-se que assim como os microcomputadores vêm causando nos últimos anos uma revolução nas indústrias, os supercomputadores no outro extremo do espectro, irão causar nos próximos anos uma maior revolução ainda. Segundo o prêmio Nobel de física Ken Wilson ^[5], o uso de computadores na simulação de protótipos e otimização de novos produtos e processos é geralmente o método existente mais barato e mais rápido. Acontece que são poucos os cientistas e engenheiros provenientes das universidades que sabem algo sobre computação em larga escala ^[5] de modo a poderem utilizá-los eficazmente assim como contribuir com novas idéias para o desenvolvimento dos supercomputadores.

Apesar disso as maiores indústrias têm se beneficiado da computação em larga escala. Em particular, as indústrias petrolífera (no tratamento de dados geológicos, simulação do fluxo de petróleo nas jazidas), aeronaval e automobilística (simulação do desempenho de protótipos), eletrônica (projetos de chips a três dimensões) e energética (simulação de reatores nucleares e redes de potência).

Espera-se que nos próximos 10 anos, na medida em que a utilização da integração em larga escala como vem ocorrendo com os supercomputadores em desenvolvimento pela Fujitsu, Hitachi e Control Data, computadores de altíssimo desempenho estarão disponíveis a preços 10 a 20 vezes menores aos atuais e comparáveis aos superminis disponíveis

hoje no mercado como o VAX da DEC.

No Brasil devido ao número relativamente pequeno de laboratórios de computação, os quais ainda sofrem de pequena capacidade computacional, poder-se-ia pensar numa política mais efetiva na qual o governo, as indústrias interessadas (por exemplo as já citadas) juntamente com os laboratórios das universidades poderiam atuar na área de forma integrada.

Ressaltada a relevância do assunto passemos ao objetivo do trabalho. Tecnicamente falando os supercomputadores CRAY1 (CRAY X-MP, sua versão mais nova), CYBER 205, Fujitsu VP-200 pertencem a classe dos processadores vetoriais 'pipelined' (PVP). Neste artigo, a estrutura dessa classe de processadores será analisada. Para tanto iremos concentrar a atenção nas propriedades e na interação entre os componentes básicos da arquitetura, ou sejam, a) 'pipelines' aritméticos, b) memória principal, c) memória local, d) 'pipelines' de instrução e as instruções vetoriais.

A sequência de apresentação é a seguinte. Após observar que os PVPs exibem um objetivo de desempenho comum, ou seja, os processadores são organizados para processarem eficientemente instruções vetoriais através do princípio de 'pipelining' sem penalizar o desempenho das instruções escalares, nós examinaremos operações vetoriais 'pipelined'. Em seguida, verifica-se que esta condição determinará os limites inferiores de desempenho dos outros componentes da arquitetura e assim passaremos a analisá-los.

2. PIPELINES ARITMÉTICOS

A necessidade de instruções vetoriais surge ao se efetuar operações repetidamente numa sequência de pares de operandos, por exemplo, numa adição de dois vetores (B & C) como representado no seguinte 'loop' FORTRAN:

```
DO 1 I=1,N
1  A(I)=B(I)+C(I)
```

Este 'loop' quando traduzido para linguagem de máquina corresponderá a uma sequência de instruções escalares que definirão a variável de incremento do 'loop', carregamento dos operandos nos registradores, adição de operandos, armazenamento do resultado e teste de fim de 'loop'. Tais regulares e relativamente simples instruções são bem adequadas para uma implementação rápida e eficiente em 'hardware'. Por outro lado, a frequente ocorrência desses cálculos particularmente em aplicações científicas onde a manipulação de matrizes numéricas é uma operação comum, justificaram o desenvolvimento de duas classes de processadores que representam duas estratégias para solucionar o problema. Uma é o processador vetorial paralelo tal como o Burroughs Scientific Processor [6], uma evolução do ILLIAC IV que emprega um número relativamente grande de elementos processadores (EP). A cada par de operadores é destinado um EP e a operação é executada simultaneamente nos EPs. A outra classe corresponde aos processadores vetoriais 'pipelined' que utiliza um número pequeno de 'pipelines' aritméticos para realizar os cálculos através do princípio de 'pipelining'.

'Pipelining' [7] [8] é definido genericamente como a divisão de uma função a ser realizada em sub-funções que podem ser executadas automaticamente por 'hardware' dedicado. Dessa forma sucessivas execuções da função podem ser conduzidas pelas sub-funções operando por superposição ('overlapped'). Um exemplo de aplicação desse princípio é 'pipeline' aritmético (PA).

Um PA básico aceita 2 operandos a cada ciclo de relógio T e produz 1 resultado C ciclos depois. O valor C depende do número de estágios que a operação requer. O desempenho de uma PA é geralmente medido em termos de seu 'throughput', ou seja, o número de resultados escalares produzido por uma unidade de tempo. Em PVPs, uma unidade comum é MFLOPS (milhões de operações de ponto flutuante por segundo) que para o PA descrito acima é dado por $1/T$ e que independe do número de estágios C .

As configurações típicas dos PVPs empregam mais do que um PA para obter maior desempenho. O CRAY1 (CRAY-X-MP, processador dual) como visto na figura 1, possui 12 unidades funcionais ou PAs unifuncionais onde cada PA só pode executar um tipo de operação. Seu ciclo de relógio é de 12.5 ns, (9.5 ns), com operandos de 64-bits, 'throughput' de 80 (105) MFLOPS por PA, estes poderão ser encadeados nas instruções vetoriais e multiplicar o desempenho do processador.

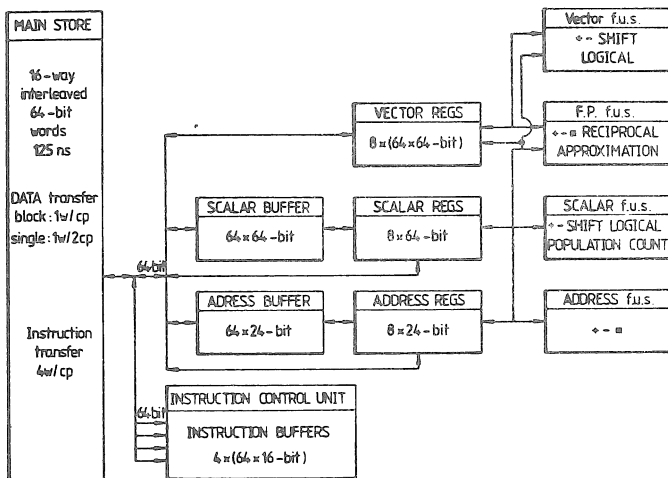


Figura 1: A arquitetura do CRAY1

No CYBER 205, representado na figura 2, os PAs (até 4) são multifuncionais estáticos ou seja, um PA só pode executar um tipo de operação de cada vez, mas pode ser configurado para realizar diferentes operações. O ciclo do relógio é de 20 ns, com operandos de 64-bits e 'throughput' de 50 MFLOPS por PA. No máximo duas instruções vetoriais podem ser encadeadas e assim dobrar o 'throughput' de um PA mas diferente de CRAY1 um dos

operandos dessas instruções tem de ser escalar.

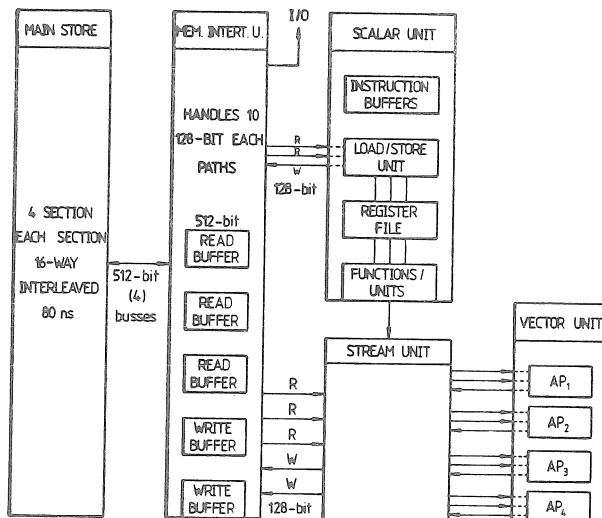


Figura 2: A arquitetura do CYBER 205

3. MEMÓRIA PRINCIPAL

Se o 'throughput' dos PAs forem considerados como o objetivo dos projetos dos processadores vetoriais, os requisitos da memória principal (MP) podem ser estimados. Para sustentar o fluxo máximo de processamento é necessário que a MP forneça 2 operandos e armazene 1 resultado a cada ciclo do relógio o que determina uma bandapassante da memória equivalente a três vezes (taxa de demanda) o 'throughput' do processador.

A memória principal do CRAY1 (CRAY X-MP) é composta de 'chips' de 1K bipolar, com 1 milhão de palavras, intercalada em 16 bancos com ciclo de 50 ns (38 ns) e bandapassante apenas de 80 MP/s (milhões de palavras de 64-bits por segundo). Este desbalanceamento entre a bandapassante de memória e o 'throughput' do PA foi corrigido no CRAY X-MP cuja bandapassante é de 420 MP/s com 4 barramentos independentes para aproveitar a intercalagem dos bancos e ainda suportar acessos de entrada e saída. Para aliviar o provável engarrafamento da memória principal, o CRAY1 dispõe de uma memória local formado por oito registradores vetoriais, cada um podendo armazenar até 64 elementos de 64-bits. Instruções vetoriais permitem transferir blocos de dados (vetores) entre as duas memórias.

A memória do CYBER 205 é composta de 'chips' de 4K bipolar, com ciclo de 80 ns e é dividida em 4 seções independentes, cada uma com 1M palavras de 64-bits. Cada seção é intercalada de 16 modos e acessada através de uma barra de 512-bits de largura. Cada seção é organizada em 16 pilhas de 8 bancos de memória, cada um com 16K palavras de 32-bits. A intercalagem é obtida pelo fornecimento por cada pilha de uma palavra de

32-bits em paralelo.

A interface de memória gerencia 4 barras, que servem, cada uma a uma única seção e organiza o despacho de pedidos de acesso vindo do resto do sistema. Estes pedidos são feitos através de 10 barras de dados, cada um com 128-bits de largura, dos quais 5 são dedicados ao processamento vetorial, 3 ao processamento escalar e 2 ao processamento de E/S. Os pedidos de acesso a memória são montados em 5 'buffers' de dados, cada um com 512-bits de largura dentro da interface. Cada seção fornece uma bandapassante de 400 MP/s e portanto pode satisfazer a taxa de demanda dos PAs.

É importante observar que a intercalagem da memória no CYBER 205 e no CRAY1 é implementada diferentemente. No primeiro, palavras de memória com endereços consecutivos poderão ser lidas num ciclo. No segundo, palavras com endereços regularmente espaçados (espaçamento de 0 a 15) poderão ser acessadas. Esta última forma de intercalagem é claramente mais flexível e oferece significativa vantagem para o processamento vetorial como será visto adiante.

4. MEMÓRIAS LOCAIS

Memórias locais (ML) permitem que se utilize eficientemente a bandapassante da memória principal e desempenham um papel importante nos PVPs como ilustrado pelos registradores vetoriais presentes no CRAY1. É interessante notar que o mecanismo de registradores vetoriais introduzido no CRAY1, foi ainda mais explorado no recente Fujitsu VP-200 [2] que dispõe de 8K palavras de 64-bits para configurar tais registradores, ou seja, variando o número de elementos, de 8 a 256 registradores vetoriais poderão ser definidos.

Outras formas especializadas de MLs, baseadas nas características de acesso a dados e instruções pelos programas, têm sido utilizadas. O processamento de instruções endereçadas sequencialmente promoveu o uso de 'buffers' de instruções (BI) que são capazes de guardar com antecedência varias instruções que eventualmente serão processadas, a menos que ocorra uma ramificação. Esta capacidade extra pode ser explorada eficientemente por 'pipelines' de instrução como veremos adiante. No CRAY1 existem 4 BIs, cada um dispõe de 64 palavras (uma palavra pode conter até 4 instruções). No CYBER 205, 64 palavras são utilizadas como BI (cada palavra pode conter até 2 instruções).

Uma propriedade dos acessos a operandos é que elas tendem a se agrupar num pequeno conjunto de segmentos de memória por um período de tempo e depois migrar para um outro conjunto e assim por diante. Esta propriedade tem sido explorada nos processadores escalares de alto desempenho através do uso de MLs denominadas cache de dados [9].

Nos PVPs, as instruções vetoriais com acesso independente a memória principal exacerbam o problema de coerência dos dados o que faz com que somente formas simples de cache de dados programáveis sejam utilizadas. Assim, no CRAY1 são utilizados 2 conjuntos de 64 elementos (registradores B e T) que servem de memória temporária por exemplo, para preservar os 16 registradores de trabalho (8-A e 8-S) entre chamadas de subrotinas de modo a evitar o acesso a memória principal. Na verdade, a impossibilidade de indexar

os registros (B e T) em tempo de execução (eles têm que ser explicitamente alocados durante a compilação) dificultam sua utilização. O CYBER 205 possui 272 registradores escalares de trabalho.

5. 'PIPELINES' DE INSTRUÇÃO

Para descrever e caracterizar o processamento de instruções nos PVPs é útil considerar separadamente cada uma das fases do processamento, ou seja, despacho de instruções, processamento escalar e processamento vetorial.

Os 'pipelines' de instrução (PI) controlam o carregamento e o despacho das instruções. O tráfego de instruções entre os BIs e a memória principal é feita em blocos (técnica de 'lookahead' [10]) e utilizam eficientemente a bandapassante da memória: por exemplo no CRAY1 (4 barras de 64-bits, 320 MP/s) e o CYBER 205 (1 barra de 128-bits, 100 MP/s).

Instruções escalares e vetoriais são decodificadas e despachadas pela unidade de controle centralizada do PI. Assim que uma instrução é despachada, o 'pipeline' aritmético selecionado pelo PI assumirá o controle da execução. Isto permite que as instruções seguintes sejam processadas e entregues ao PA apropriado, aumentando o 'throughput' do processamento de instruções.

O processamento escalar tanto no CRAY1 como no CYBER 205 possui um conjunto dedicado de PAs. Isto permite que a cada ciclo uma instrução seja despachada e a execução de várias instruções superpostas. No entanto isto pode causar riscos ('hazards') entre as instruções na medida em que as operações compartilham recursos e necessitam de diferentes tempos de execução e podem não seguir a sequência predefinida e o resultado não ser o esperado. Para prevenir tal situação ocorra ambos PVPs empregam mecanismos de reserva sobre os recursos (exemplo: registradores, unidades funcionais) que estão sendo utilizados. Assim, por exemplo, instruções não serão despachadas de qualquer dos seus registros operando tiver sido reservado.

O processamento vetorial no CYBER 205 é realizado nos múltiplos e independentes PAs de ponto flutuante, cada um deles contém 5 unidades funcionais que se comunicam através de uma via de dados comum. No CRAY1 existem 3 unidades funcionais vetoriais e o processamento vetorial ainda compartilha das 3 unidades funcionais de ponto flutuante utilizadas por operações escalares. A diferença significativa entre os dois PVPs é que enquanto no CYBER 205 os vetores operando e resultado das operações vetoriais residem na memória, no CRAY1 eles se localizam nos registradores vetoriais. A consequência é que o tempo de inicialização ('start up') do processamento vetorial é significativamente maior no CYBER. Assim para pequenos vetores (menor do que 100 elementos) o CRAY1 apresenta melhor desempenho.

6. PROPRIEDADES DAS INSTRUÇÕES VETORIAIS

O repertório das instruções dos PVPs [11] (219 instruções no CYBER, e 120 no CRAY1 aproximadamente) compreendem instruções escalares e vetoriais. Os conjuntos de

instruções escalares são do tipo convencional de três endereços e não requerem uma melhor descrição. Ao contrário dessas, as instruções vetoriais apresentam propriedades únicas que valem a pena avaliar.

Instruções vetoriais têm o formato de três endereços que definem 2 vetores operandos e 1 vetor resultado. Informações vetoriais básicas consistem de: número de elementos, endereço inicial na memória e incremento para indicar os elementos seguintes. Nesse particular o CRAY1 é diferente do CYBER 205 e permite incrementos não unitários fazendo com que matrizes possam ser acessadas por linha, coluna, diagonais, etc... e utilizar em tais situações a bandapassante de memória eficazmente. Por exemplo:

(VL é um registrador de 6-bits que indica o comprimento dos vetores operandos no CRAY1)

V_i Ao, A_k transfere (VL) palavras para o registrador vetorial V_i a partir do endereço (Ao) incrementado por (A_k)

Em contraste com o CYBER que pode processar com uma única instrução vetorial vetores de até 64K elementos, o CRAY1 opera com vetores de no máximo 64 elementos. Para processar vetores maiores eles terão que ser divididos e operados em grupos de 64.

Além dos vetores densos (que exibem uma configuração de endereços regulares na memória) existe uma outra importante classe constituída pelos vetores esparsos nos quais um grande número de elementos são zeros. Para processar eficientemente esses vetores é necessário evitar o processamento de operações nos elementos nulos. A solução típica é associar um vetor de controle (formado de bits) a cada vetor que participará da operação. Um elemento do vetor é selecionado para processamento de acordo com o valor do bit correspondente do vetor de controle.

Operações envolvendo vetores de controle permitem vetores esparsos serem compactados, utilizados nesta forma em operações vetoriais e depois expandido para as formas originais. O CYBER 205 dispõe desta facilidade que também é empregada para superar o problema de endereçamento contíguo a memória pois qualquer configuração de 1's pode ser especificada. Porém a menos que a memória aceite tal 'string' de controle, palavras consecutivas serão acessadas de qualquer forma. No CRAY1 o registrador VM de 64-bits serve de forma mais rudimentar ao propósito do vetor de controle em algumas poucas operações de 'merge' dois exemplos são dados a seguir:

CRAY1: $V_i V_j!V_k \& VM$ (coloque em V_i o elemento de V_j ou V_k de acordo com o valor correspondente bit de VM.)

CYBER 205: $MPYUV (A,X),(B,Y),C,Z$ (multiplique os vetores especificados por (A,X) e (B,Y) sob controle do vetor Z e coloque o resultado no vetor especificado por C e C+1)

A,B,C,X,Y,Z são registradores

(A,X): 48-bits do registrador A indica o endereço base do vetor

16-bits do registrador A indica o comprimento do vetor

16-bits do registrador X indica o deslocamento do endereço base

Existem ainda dois outros grupos de instruções vetoriais. O primeiro envolve a utilização de um vetor índice que indica quais os elementos de um vetor operando que irão participar da operação, como numa consulta indireta a uma tabela. Essas operações são referidas como REUNE ('gather') e DISPERSA ('scatter'). O outro grupo corresponde a operações aritméticas denominadas REDUÇÕES VETORIAIS que produzem um resultado escalar a partir de vetores operandos, por exemplo, o somatório dos elementos de um vetor e o produto interno de dois vetores. Somente o CYBER 205 dispõe desses grupos de instruções vetoriais.

7. CONSIDERAÇÕES SOBRE O DESEMPENHO DOS PVPs

Somente a partir do insucesso da primeira geração de PVPs diante dos processadores escalares de alto desempenho é que modelos analíticos foram utilizados para avaliar o impacto escalar nos PVPs $|_{12}| |_{13}|$. O fato é que os programas não podem ser completamente escritos utilizando instruções vetoriais e portanto o aumento do desempenho que pode ser obtido depende não só da quantidade do processamento vetorial mais também da eficiência do processamento escalar.

Estudos envolvendo modelagem, simulação e avaliação de desempenho $|_{14}| |_{15}| |_{16}| |_{17}| |_{18}|$ e propostas de novas arquiteturas de PVPs $|_{14}| |_{19}|$ têm surgido na literatura de modo crescente na medida em que o acesso ao PVPs pela comunidade científica tem sido ampliado.

Embora observado que a maioria das soluções dos problemas científicos e de engenharia envolve a manipulação de dados na forma matricial, outros fatores também influenciam a eficácia do processamento vetorial como um todo $|_{20}| |_{21}|$. Entre estes fatores destacam-se os problemas associados com o processo de conversão do 'software' desenvolvido para máquinas escalares para os processadores vetoriais. As soluções encontradas frequentemente se enquadram em duas categorias:

- I - Desenvolvimento de extensões para a linguagem FORTRAN e de novas linguagens vetoriais de alto nível $|_{22}| |_{23}|$, que permitem ao usuário codificar diretamente sua aplicação sob a forma de sintaxe vetorial.
- II - Desenvolvimento de compiladores vetoriais e técnicas de vetorização $|_{24}|$ que procuram recuperar a estrutura vetorial codificada na sintaxe escalar.

Outro fator importante é o esforço que se requer para reformular a física, a aproximação numérica e algoritmos para explorar a capacidade de paralelismo nos cálculos a serem executados no 'hardware' vetorial. Uma solução rotineira tem sido a de se desenvolver algoritmos vetoriais e paralelos $|_{25}|$ para serem utilizados na aplicação.

8. CONCLUSÕES

A presente análise dos supercomputadores permite se ter uma visão mais global e crítica dessa classe de arquitetura em contraste com as colocações mais específicas e descritivas encontradas na literatura.

Na opinião do autor, é imperativo o desenvolvimento mais intenso da computação científica em larga escala no Brasil, na medida em que as principais indústrias nacionais atingem um estágio que necessita de tecnologias mais avançadas. Devido aos recursos econômicos exigidos é preciso uma política nacional que integre indústrias, laboratórios de pesquisa e universidades para a utilização do potencial dos processadores de alto desempenho.

REFERÊNCIAS

- [1] KOZDROWICKI, E. W., THEIS, D. J., "Second Generation of Vector Supercomputers", IEEE COMPUTER, nov 1980.
- [2] MENDEZ, R., ORSZAG, S., "The Japanese Supercomputing Challenge", DATAMATION, vol 30 no 7, may 15 1984.
- [3] DALLAIRE, G., "American Universities Need Greater Access to Supercomputers", CACM, vol 27, no 4, may 1984.
- [4] "Access to Supercomputers: an NSF Perspective", entrevista com E. T. MAYES., CACM, vol 27, no 4, may 1984.
- [5] RODRIGUE, G. et al., "Perspective on Large Scale Scientific Computations", COMPUTER, Tutorial Series, oct 1980.
- [6] JENSEN, C. "Taking Another Approach to Supercomputers", DATAMATION, february, 1978.
- [7] RAMAMOORTHY, C. V., LI, H. F., "Pipeline Architecture", ACM Computing Surveys, vol 9, no 1, march 1977.
- [8] KOGGE, P. M., "The Architecture of Pipelined Computers", McGraw hill Book CO., 1981.
- [9] SMITH, A. J., "Cache Memories", Computer Surveys, vol 14, no 3, sep 1982.
- [10] KELLER, R. M., "Lookahead Processors", Computing Surveys, vol 7, no 4, dec 1975.
- [11] HOCKNEY, R. W. & JESSOPE, C. R., "Parallel Computers", Adam Hilger Ltd., Bristol 1981.
- [12] RUDSINSKI, L., WORLTON, J., "The Impact of Scalar Performance on Vector and Parallel Processors", High Speed Computer and Algorithm Organization, Kuck, D. J. et al. (eds) Academic Press, NY, 1977.
- [13] WIJK, H. C. VAN DER, "A Comparison of the CRAY1 and CDC 7600 Computer Systems", Mphil Thesis, Computing Dept., Imperial College, 1981.
- [14] AMORIM, C. L., "Simulation Performance of a Class of Vector Processors", PhD Thesis, Computing Dept., Imperial College, 1984.
- [15] LANG, D. E. et al., "A Modelling Approach and Design Tool For Pipelined Central Processors", IEEE Proc. Symp. Computer Architecture, 1979.

- |₁₆| PAUL, G., "Studies in Vector Architecture", Research Report, rc 9245, Computer Sciences Dept, IBM T. J. Watson RC, 1982.
- |₁₇| BASKETT, F. & KELLER, T. W., "An Evaluation of the CRAY1 Computer", High Speed Computer and Organization, Kuck et al. (eds.), Academic Press. 1977.
- |₁₈| SRINI, V. P., ASENJO, J. T., "Analysis of CRAY1s Architecture", the 10th. Annual Intern Symp. on Computer Architecture, SIGARCH, vol 11, no 3, jun 1983.
- |₁₉| SMITH, J. E., "Decoupled Access/Execute Computer Architecture", IEEE, 9th Annual Symp. on Computer Architecture, SIGARCH New, vol 10, no 3, apr 1982.
- |₂₀| WIRSCHING, J. E., KIRSHI, T., "Matching Machines and Problems", High Speed Computer and Algorithm Organization, Kuck, D.J. et al. (eds), Academic Press NY, 1977.
- |₂₁| CALAHAN, D. A., "Algorithmic and Architectural Issues Related to Vector Processors" AP 1027-776, Michigan Univ., Ann Arbor, System Eng. Lab., 1976.
- |₂₂| BASILI, V. R., KNIGHT, J. C., "A Language Design for Vector Machine", SIGPLAN notices, vol 10, no 3, March 1975.
- |₂₃| PERROT, R. H., "A Language for Array and Vector Processors", ACM Transaction on Programming Language and Systems, vol 1, no 2, Oct. 1979.
- |₂₄| AMORIM, C. L., "An Evaluation of Vectorization Techniques", MSc Thesis, Dept. of Computing, Imperial College, 1980.
- |₂₅| FONG, K. & JORDAN, T. L., "Some Linear Algebraic Algorithms and their Performance on CRAY1", LA 6774, LASL, 1977.