PROCESSADORES BIPARTIDOS ASPECTOS DE PROJETO

Felipe Maia Galvão França *

Edil Severiano Tavares Fernandes **

*M.Sc. (COPPE/UFRJ,1987); arquitetura e organização de computadores, projetos VLSI, microprogramação, sistemas operacionais; Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ, Caixa Postal 68511-Rio de Janeiro-RJ-CEP: 21945.

**Ph.D. (Imperial College, London,1981); arquitetura de computadores, processamento paralelo, microprogramação, sistemas operacionais; Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ, Caixa Postal 68511-Rio de Janeiro-RJ-CEP: 21945.

SUMARIO
As motivações e contextos do aparecimento de processadores bipartidos devem ser esclarecidos. A deteção de paralelismo potencial ao nível convencional de instruções durante a execução de uma tarefa sequencial permite que se dobre o alcance máximo do desempenho. RISC e VLSI caracterizam o meio-ambiente de dois exemplos que serão vistos aqui.

1. INTRODUÇAO

A busca por soluções que impliquem em um melhor desempenho em Computação, tem incentivado inúmeras pesquisas baseadas no emprego de paralelismo: ações dos diferentes níveis de implementação ("hardware", "firmware", convencional etc.) sendo levadas a cabo simultaneamente.

O novo distanciamento semântico (linguagens de alto nível versus instruções de máquina) provocado pela introdução e validação dos processadores do tipo RISC ("Reduced Instruction Set Computer"), tornou aparente um novo campo onde o paralelismo também pode ser explorado: decomposição de comandos das linguagens de alto nível em primitivas RISC que poderiam ser executadas em paralelo. Os processadores bipartidos que serão descritos neste trabalho resultaram da exploração de caminhos diferentes dentro deste campo:

-o processador PIPE (Parallel Instructions and Pipelined Execution) é um dos resultados de pesquisa do Departamento de Ciência da Computação da Universidade de Wisconsin-Madison [SMIT82] [SMIT83] [SMIT84] [CRAI83].

-a máquina FILA&PILHA foi idealizada sob o contexto de um dos projetos levados a cabo no Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ [FERN83] [FRAN86] [FRAN87]

2. BIPARTIÇÃO: MOTIVAÇÕES COMUNS

2.1. "GARGALOS"

A identificação de "gargalos" no fluxo de execução de programas, motivou o aparecimento das máquinas bipartidas. O trabalho de J.E. Smith e equipe deixa claro os pontos a serem abordados. O "gargalo de von Neumann" como proposto por J. Backus [BACK78], significa que em sua forma mais simples, um "computador von Neumann" é limitado na transmissão de apenas uma palavra por vez entre Unidade Central de Processamento (UCP) e memória principal.

Esta limitação não parece ser um problema difícil de contornar, embora normalmente caracterize arquiteturas VLSI. O real entrave até então estava no chamado "gargalo de Flynn" [FLYN66]: a possibilidade de se executar apenas uma (1) instrução por ciclo de decodificação em uma organização SISD (Single Instruction Stream Single Data Stream), que no melhor caso reduz-se a uma (1) unidade de tempo "atômica" do processador ("clock"), significava um limite de desempenho em vários processadores "pipelined".

As duas máquinas que serão adiante abordadas têm como principal característica comum burlar esta limitação de maneira a virtualmente duplicar o desempenho potencial. É assumido como domínio do problema a classe de programas que então traduziam-se em dados e instruções sequenciais.

A adoção de um conjunto reduzido de instruções primitivas permitiu tanto a identificação do paralelismo que deu forma a estas arquiteturas físicas bipartidas como também quebrar um outro gargalo: o escalonamento ótimo das instruções em tempo de compilação. A bipartição pode implicar em um melhor aproveitamento dos recursos disponíveis ("pipes") nesses processadores.

2.2. APLICAÇÕES

O uso de processadores bipartidos é equivalente aos "convencionais" (SISD). Em sistemas multiprocessadores (MIMD), o ganho "individual" poderia ser refletido no desempenho total respeitando-se as limitações inerentes à arquitetura do sistema.

Um outro aspecto que deve ser levantado trata do custo imposto pelo uso de dois integrados ao invés de um somente. Embora não tenha sido feita nenhuma projeção nesse sentido, acreditamos que, a princípio haveria viabilidade pois um só tipo de integrado seria fabricado.

3. O PROCESSADOR PIPE

3.1. DESCRIÇÃO

Nesta proposta, dois processadores fisicamente idênticos podem trabalhar cooperativa e concorrentemente na execução de uma (1) tarefa (sequencial sob o ponto de vista do

programador de linguagens de alto nível), com filas implementadas em "hardware" para intercomunicação. A divisão do trabalho é feita pela definição de dois modos de operação: Acesso e Execução. A Figura 1 esquematiza esta máquina.

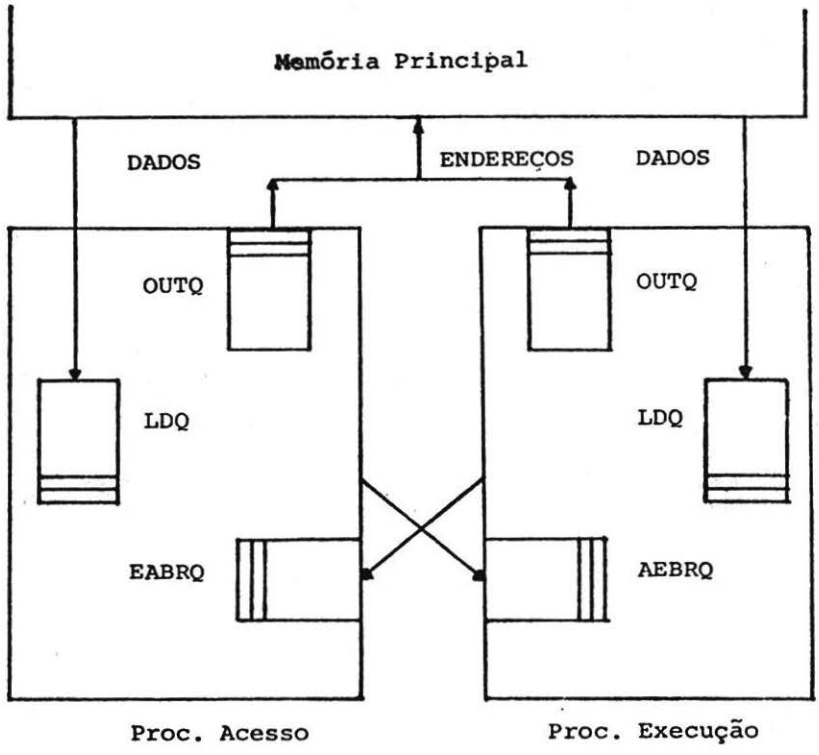

Figura 1. O Processador PIPE.

O processador de modo Acesso calcula endereços e realiza as operações sobre dados na memória principal necessárias aos dois processadores. O processador de modo Execução é responsável pelo processamento propriamente dito. Cada um obedece a uma (1) determinada sequência de instruções em tempo de execução, sendo esta divisão de trabalho feita em tempo de compilação. O processador de Acesso normalmente "roda" adiantado em relação ao de Execução visando minimizar conflitos em referências à memória principal.

3.2. CONSIDERAÇÕES DE PROJETO

Voltado para uma implementação VLSI onde o número de pinos pode ser um fator limitante, o uso de filas na arquitetura física PIPE procura descongestionar o "gargalo de von

Neumann". Para evitar o "gargalo de Flynn" dois contadores de programa e unidades de decodificação estão presentes nas duas unidades de processamento. Destaca-se novamente que esta bipartição pode permitir que se alcance mais facilmente uma geração de código otimizado visando o melhor aproveitamento dos estágios ("pipes") existentes.

Também é realçado neste projeto o uso de "CACHE" somente para instruções. As justificativas apresentadas baseam-se na busca por uma solução que possua uma relação custo/benefício adequada ao conjunto bipartido. Temos como exemplo que, cuidando-se apenas de código, não é necessário tratar escritas que fatalmente implicariam na manutenção de coerência entre dois controladores, o que significaria um aumento considerável da complexidade do projeto como um todo.

3.3. AVALIAÇÕES REALIZADAS

Os resultados apresentados até o momento são relativos à arquitetura de um famoso computador "pipelined", o CRAY-1 [RUSS78]. O processamento escalar foi comparado considerando-se as mesmas durações no acesso à memória pricipal e execução das instruções em número de períodos de relógio. Além de ser verificado uma menor sensibilidade à acessos na memória principal, foi constatado um ganho global de mais de 50 por cento em cima do CRAY-1 [SMIT84] [SMIT86].

4. A MAQUINA FILA&PILHA

4.1. DESCRIÇÃO

Novamente, dois processadores fisicamente idênticos funcionam em paralelo de modos diferentes: modo Sequenciamento/Cálculo de Endereços Primários (S/CE) e modo Execução/Acessos (E/A). Está previsto um (1) pino para esta seleção de modo em uma implementação VLSI. Apenas uma sequência de código é gerada pelo compilador.

O processador S/CE tem por função buscar as instruções na memória de código, "desempacotá-las", executar as de sequenciamento e cálculo de endereços primários (instruções de classes 0 e 1) e enfileirar as demais (classes 2 e 3) que serão interpretadas pelo processador E/A. Este último recebe as instruções da FILA do processador S/CE (Expressões, Comparações e Acessos a dados) e as executa sequencialmente nunca interferindo diretamente no controle do programa em andamento. A divisão do trabalho é feita naturalmente em tempo de execução. Na Figura 2 temos um esquema simplificado.

Cada processador possui dois barramentos externos de 32 bits: um de intercomunicação e outro dedicado à memória principal. É possível então definir áreas lógica e fisicamente exclusivas de memória principal: o processador S/CE comunica-se com a área de código e o E/A com a de dados.

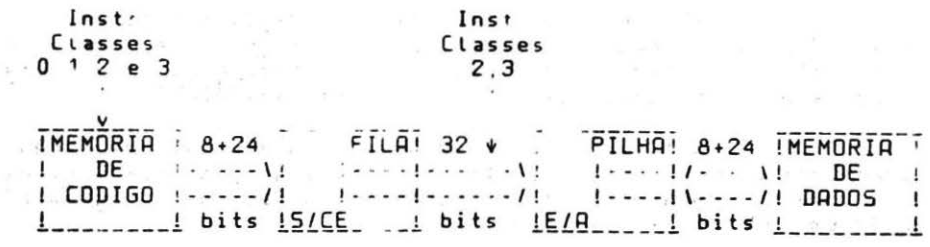

Figura 2 O processador FILA&PILHA.

4.2. CONSIDERAÇÕES DE PROJETO

A separação da memória principal na forma descrita permite acessos simultâneos a código e dados, amenizando a princípio o "gargalo von Neumann" da máquina Contribuindo no mesmo sentido, a representação das instruções em um (1) byte provoca na grande maioria dos acessos ao código, a busca de mais de uma instrução por vez. A FILA do processador S/CE que alimenta sequencialmente o processador E/A, também amortece os custos indesejáveis de comunicação entre os dois processadores.

Na definição do conjunto de instruções foram identificadas ações primitivas e representativas que mais ainda, eram potencialmente concorrentes Nos contornos desta definição estão as linguagens de alto nível que utilizam como base uma Máquina de Pilha. O uso de operandos implícitos converge com as novas relações semânticas trazidas pelos RISC's e o efeito imediato é uma atenuação na tendência de aumento exagerado do tamanho de código gerado.

Verificado que não existiria a necessidade de um "CACHE" para código, um esquema de PILHA LOCAL foi adotado (CACHE de dados sem tratamento de consistência). A bipartição desta organização veio de encontro às limitações previstas em um (1) projeto VLSI.

4.3. AVALIAÇÕES REALIZADAS

Dois enfoques foram levados em consideração nos primeiros trabalhos de avaliação do processador FILA&PILHA:

a) a arquitetura sob o ponto de vista do programador;

b) a organização da máquina;

Os trabalhos de avaliação da máquina FILA&PILHA [OBRA87] e da linha INTEL IAPx 86 [SANT87] para execução de MODULA-2 [WIRT82] permitem comparações com o processador Lilith [WIRT81] Este processador foi projetado pelo próprio autor da linguagem MODULA-2 possui orientação específica e o conjunto de instruções é do tipo CISC ("Complex Instruction Set Computer")

O trabalho de D.B. Wortman [WORT72] foi adotado na avaliação das arquiteturas. Segundo este método determina-se o custo global das primitivas semânticas (fragmentos) de uma linguagem de programação de alto nivel em uma dada máquina.

A escolha de um "benchmark" para nossa avaliação, teve como orientação a possibilidade de analogia com outros processadores. Contando com grande aceitação, o "benchmark" utilizado em [GRAP81] foi adotado.

Os custos estáticos, ou sejam, os custos de representação dos programas, encontram-se resumidos na Tabela 1.

	Acker	Puzzle	Search	Sieve
Lilith	79	1254	97	111
8086	199	2964	263	342
80286	198	2936	262	341
Fi&Pi	401	4265	314	345

Tabela 1. Tamanho do Código em Bytes

A parte dinâmica encontra-se a seguir na Tabela 2.

	Acker	Puzzle	Search	Sieve
Lilith	683 +E04	3110+E04	2097+E04	704 +E04
8086	1254+E04	8617+E04	4338+E04	2033+E04
80286	1208+E04	8487+E04	4308+E04	2033+E04
Fi&Pi*	2362+E04	7158+E04	3556+E04	1489+E04
Fi&Pi**	2018+E04	6412+E04	3081+E04	1279+E04

* : taxa de falha na PILHA LOCAL= 100%
**: taxa de falha na PILHA LOCAL= 0%

Tabela 2. Volume de Referências à Memória Principal (Código e Dados)

O outro enfoque que deve ser abordado diz respeito à arquitetura fisica do processador. A Tabela 3 mostra o único resultado conseguido até o momento. Por não contarmos com um compilador, torna-se deveras árdua esta tarefa de avaliação.

	Acker	Puzzle	Search	Sieve
8086	55.5+E06	22.0+E07	36.8+E03	35.8+E05
68000	44.0+E06	14.7+E07	20.8+E03	31.4+E05
Fi&Pi	34.2+E06	-	-	-

obs1: sem tempos de espera por memória

Tabela 3. Ciclos gastos para execução.

Deve ser ressaltado que embora o método proposto

por D B Wortman não consiga refletir com fidelidade o desempenho esperado de uma máquina RISC escolhemos o programa Ackermann [WICH76] em primeiro lugar por apresentar o maior contraste na avaliação da arquitetura que foi realizada inicialmente.

5. COMENTARIOS E CONCLUSOES

A classificação de computadores proposta por M.J. Flynn [FLYN66] parece ter chegado a um ponto de reavaliação. A organização da máquina FILA&PILHA inclui dois processadores idênticos mas apenas uma sequência de instruções do nível convencional é utilizada. Embora o processador PIPE quebre o mesmo "gargalo de Flynn", trabalha com duas sequências distintas de instruções o que descaracteriza uma organização SISD. Na verdade, o processador PIPE tem arquitetura e organização bipartidas enquanto que no processador FILA&PILHA somente a organização o é.

Outro aspecto explorado foi a conveniência de um conjunto reduzido e primitivo de instruções em máquinas "pipelined". As interdependências entre instruções deste tipo são minimizadas e assim as "turbulências" ("pipeline interlocks") nos "pipes" ficam reduzidas de forma a agilizar ainda mais o funcionamento. A bipartição das organizações apresentadas são resultados independentes que surgiram da observação de que instruções primitivas provenientes de uma tarefa sequencial poderiam ser executadas em paralelo.

É possível que este seja um atalho no difícil caminho do processo de independência tecnológica nacional. RISC's não são difíceis de implementar uma vez que tenha sido definida uma arquitetura. A idéia da bipartição pode também permitir um melhor aproveitamento das pouquíssimas tecnologias que nos são disponíveis.

6. BIBLIOGRAFIA:

[BACK78] BACKUS,J., "Can programming be liberated from the von Neumann style? A functional style and its algebra of programs", Communications of the ACM, vol 21, n 8, pp 613-641, August (1978)

[CRAI83] CRAIG,G.L. et alli, "PIPE: A High-Performance VLSI Processor Implementation", Technical Report n 513, University of Wisconsin, September (1983)

[FERN83] FERNANDES,E.S.T., FRANÇA,F.M.G., OBRACZKA,K., SANTOS,M.S., "Em Direção a uma Máquina EDISON LSI", Anais do I Simpósio Brasileiro de Concepção de Circuitos Integrados, Novembro (1983)

[FLYN66] FLYNN,M.J., "Very-High Speed Computing Systems", Proceedings of the IEEE, vol 54, n 12, pp 1901-1909, December (1966)

[FRAN86] FRANÇA,F.M.G "Máquina de Pilha VLSI: Uma Arquitetura Bipartida" Anais do VI Congresso da Sociedade Brasileira de Computação Julho (1986)

[FRAN87] FRANÇA,F.M.G., "Um Processador Bipartido" Tese M Sc COPPE/UFRJ, Março (1987)

[GRAP81] GRAPPEL,R.G., HEMMENWAY,J.E., "A Tale of Four Microprocessors: Benchmarks Quantify Performance" Eletronic Design News, April (1981)

[OBRA87] OBRACZKA,K., "Avaliação de uma Arquitetura RISC para MODULA-2", Tese M.Sc., COPPE/UFRJ, Março (1987)

[PATT85] PATTERSON,D.A., "Reduced Instruction Set Computers", Communications of the ACM, vol 28, n 1, January (1985)

[RUSS78] RUSSEL,R.M., "The CRAY-1 Computer System", Communications of the ACM, vol 21, pp 63-72, January (1978)

[SANT87] SANTOS,M.S., "Avaliação da linha INTEL para MODULA-2" Tese M.Sc., COPPE/UFRJ, Março (1987)

[SMIT82] SMITH,J.E., "Decoupled Acess/Execute Computer Architectures", Proceedings of the 9th Annual Symposium on Computer Architecture, May (1982)

[SMIT83] SMITH,J.E. et alli, "PIPE: A High-Performance VLSI Architecture", Technical Report n 512, University of Wisconsin, September (1983)

[SMIT84] SMITH,J.E., "Decoupled Acess/Execute Computer Architectures", ACM Transactions on Computer Systems, vol 2, n 4, pp 280-308, November (1984)

[SMIT86] SMITH,J.E., WEISS,S., PANG,N.Y., "A Simulation Study of Decoupled Architecture Computers", IEEE Transactions on Computers, vol C-35, n 8, August (1986)

[WICH76] WICH,B.A., "Ackermann's Function: A Study in the Efficiency of Calling Procedures", BIT, vol 16, pp 103-110 (1976)

[WIRT81] WIRTH,N., "The Personal Computer Lilith", Institute für Informatik, ETH, Zurich (1981)

[WIRT82] WIRTH,N., "Programming in MODULA-2", Springer- Verlag (1982)

[WORT72] WORTMAN, D.B., "A Study of Language Directed Computer Design", Ph.D. Thesis, Stanford University, December (1972)