

P.R. FOINA (*)

SUMÁRIO

Nesse trabalho apresentamos uma proposta de arquitetura não convencional baseada nas idéias de Dennis (DENNIS). A arquitetura apresentada é uma arquitetura controlada pelo fluxo de dados ("data-flow architecture") onde o único tipo de dados existentes é a cadeia de caracteres.

ABSTRACT

This paper presents a non-conventional architecture based on Dennis idea (DENNIS). This architecture is a data-flow one where the character string is the only data structure.

(*) Bacharel em Física (UFSCar 76); Mestre em Ciência da Computação (ICMSC-USP 80); Doutor em Informática (PUC-RJ 85); Inteligência Artificial, Sistemas Especialistas, Processamento Gráfico, Computação de quinta geração; UFSCar-DCEs, caixa postal 676, CEP 13560 São Carlos SP, fone (0162) 71-8111 ramal 143.

O desenvolvimento da informática sempre leva a uma constante busca de máquinas mais rápidas e, eventualmente, mais adequadas a determinadas aplicações. Estudos levados a cabo em meados dos anos 70, mostraram que as arquiteturas baseadas na proposta original de von Neumann (vonNEUMANN) estavam chegando próximos aos seus limites práticos. Essas análises, juntamente com a cada vez maior dificuldade em melhorar o desempenho das tecnologias de semicondutores, fizeram aparecer propostas de arquiteturas alternativas (BACKUS), (DENNIS), (TRELEAVE).

A proposta a ser apresentada aqui corresponde a uma das arquiteturas em estudo pelo grupo de arquiteturas controladas pelos dados do Departamento de Computação e Estatística da Universidade Federal de São Carlos, com vistas a sua possível implementação na forma de protótipo.

2. Conceitos Básicos

Nas arquiteturas controladas pelos dados uma instrução é executada assim que os dados necessários à ela estejam disponíveis no computador, isto é, uma determinada instrução é executada tão logo os seus operandos estejam disponíveis, independentemente da sua posição no corpo do programa. Se, na nossa arquitetura, dispusermos de vários processadores então um grande número de instruções poderão ser executadas simultaneamente, desde que elas tenham todos os seus operandos disponíveis.

As arquiteturas controladas pelos dados são classificadas em duas categorias a) arquiteturas controladas pela disponibilidade de dados e b) arquiteturas controladas pela

demanda de dados.

As arquiteturas controladas pela disponibilidade de dados implementam, de forma completa, a filosofia de execução apresentada acima, ou seja, todas as instruções cujos dados estejam disponíveis são executadas assim que haja processadores livres.

As arquiteturas controlada pela demanda dos dados também implementam aquela filosofia de execução mas com uma pequena alteração, apenas são executadas as instruções realmente necessárias para a obtenção dos resultados do programa.

3. Características da Arquitetura Proposta

O desenvolvimento da nossa arquitetura balizou-se em função de certas premissas de projeto que discutiremos nessa seção.

A primeira premissa é a de que desenvolveremos uma arquitetura controlada pela disponibilidade de dados mas que minimizasse a execução de trechos desnecessários dos programas.

A segunda premissa foi adotar, como modelo básico de arquitetura controlada pelos dados, a proposta desenvolvida na Universidade de Manchester por Gurd e equipe (GURDa), (GURDb), (CATTO).

A terceira premissa é quanto ao tipo básico de dados a ser manipulado pela arquitetura. Foi adotado a cadeia de caracteres como estrutura de dados básica dessa proposta. Notem que um número é na verdade uma cadeia de dígitos e portanto não precisaremos nos preocupar com questão de precisão, representação, erros, etc. Da mesma forma uma texto é uma

A última premissa de projeto é quanto ao fato que, dentro de um programa qualquer, encontraremos certos trechos que serão executados obrigatoriamente em sequência. A nossa arquitetura deverá dispor de mecanismo para detectar esse fato e dar ao mesmo um tratamento adequado de forma a não carregar os recursos internos do computador.

4. Descrição da Arquitetura

A nossa arquitetura é formada por tres grandes subsistemas a) Memória de Operandos (MO), b) Memória de Instruções (MI) e c) Sistema de Operadores (SO) interligados entre si por um sistema de comunicação em anel (AN). A figura 1 mostra essa arquitetura de maneira esquemática.

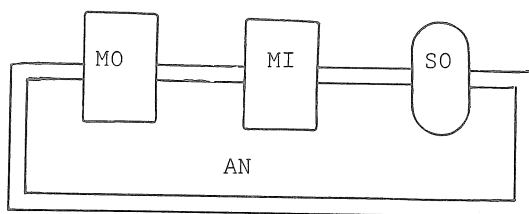


Figura 1. Esquema da Arquitetura Proposta

Um operando de uma operação binária, procura na MO o seu par. Caso ele não exista, esse operando é então armazenado (na própria MO) e fica esperando pelo par. Se o seu par já existir então ele é retirado da MO e ambos são montados juntos e

enviados para a MI.

Junto com os operandos está o indicador da instrução que os usará. Esse indicador não é o código da instrução mas apenas o número que essa instrução tem no grafo de fluxo correspondente ao programa em questão.

Na MI o indicador de instrução é substituído pelo código das instruções (pode haver mais do que uma) que irão efetivamente usar aqueles operandos. Além dos códigos de instruções, a MI anexa também os indicadores das instruções que usarão os resultados dessa operação. Ao chegar no Sistema de Operadores (SO), esse conjunto (operandos, código de instruções e indicador de destino) procura um operador que esteja disponível para a sua execução. O resultado dessa execução é um novo operando que será enviado para a MO afim de iniciar um novo ciclo.

5. Memória de Operandos

A memória de operandos armazena todos os operandos que são produzidos, ou introduzidos, durante a execução de uma programa. A MO é formada por um conjunto de Sistemas de Armazenamento de Listas (SAL) conforme mostra a figura 2. Em cada SAL temos uma estrutura de memória adequada ao armazenamento de listas encadeadas. Os operandos são formados por dois campos distintos a) indicador de instrução e b) informação. O campo indicador de instrução informa sobre qual a instrução que usará aquele operando. Essa indicação é feita através do número associado a cada instrução no grafo de fluxo

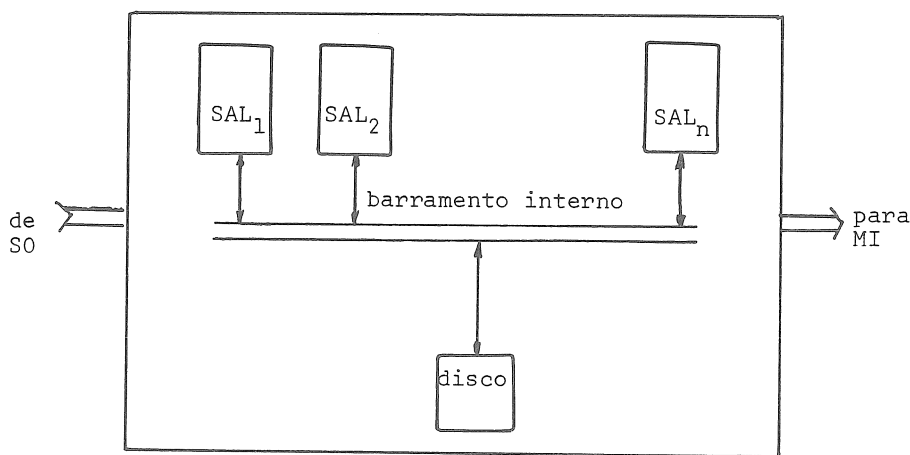


Figura 2. Sistema de Armazenamento de Listas

O compo de informação carrega o dado propriamente dito. Se o operando se destinar a uma operação aritmética teremos nesse campo uma lista de dígitos e, se o operando se destinar a uma operação sobre texto teremos então, uma cadeia de caracteres.

O indicador de instrução será usado para a alocação do operando dentro de um dos SAL existentes. Para cada SAL os indicadores de instrução formarão um diretório, acessado por rotinas de hash, que apontarão para o início do campo de informação correspondente.

O campo de informação ficará armazenado no SAL na forma de uma lista encadeada. Os ponteiros dessa lista são gerados e gerenciados pelo próprio SAL sem que isso afete a informação

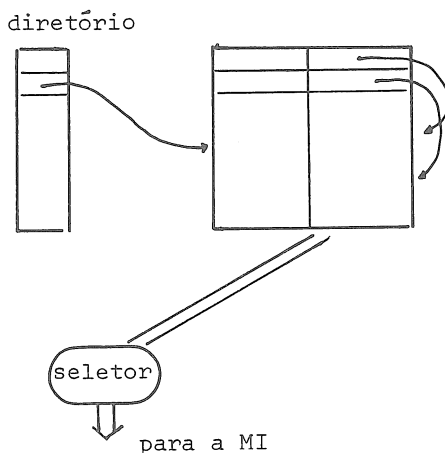


Figura 3. Estrutura da SAL

Em cada SAL existe uma unidade de seleção a qual se destina a extrair, de uma cadeia que esteja armazenada, uma sub-cadeia que será usada pela instrução que estara sendo executada. O recurso de equipar a SAL com uma unidade de extração de sub-estruturas torna-se necessário para evitar o trânsito de grandes estruturas de dados pela arquitetura.

6. Memória de Instrução

Similarmente à M0, a memória de instruções é formada por um conjunto de subsistemas de armazenamento denominados UAIs (Unidades de Armazenamento de Instruções). Cada UAI tem um diretório e uma memória estruturada em lista de forma similar à da SAL. A seleção da UAI correspondente e a busca da posição

onde está a instrução dentro da UAI é baseada no número da instrução que o operando carrega consigo.

O número da instrução corresponde a sua posição dentro do grafo de fluxo do programa. Esse identificador é substituído, na MI, pelo código da instrução correspondente. Um identificador pode corresponder a mais do que uma instrução e portanto pode-se executar um trem de instruções (que serão executadas necessariamente em sequência) de uma só vez minimizando a carga no anél. A figura 4 mostra a estrutura interna da MI e das UAIs.

diretório

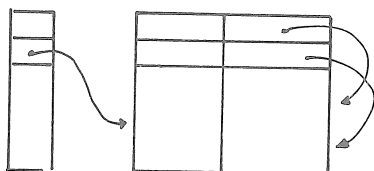


Figura 4. Estrutura das UAIs

7. Sistema de Operadores

O sistema de operadores é o órgão responsável pela efetiva execução das instruções. O sistema de operadores é formado por várias unidades de processamento cada qual capaz de executar qualquer uma das instruções do computador.

As unidades de processamento, denominadas de processadores (OP), estão interligadas entre si através de um barramento em anel. O anel tem duas portas de comunicação com o restante da arquitetura através das quais as operações são introduzidas e os resultados retirados do anel. A figura 5 mostra a estrutura desse anel.

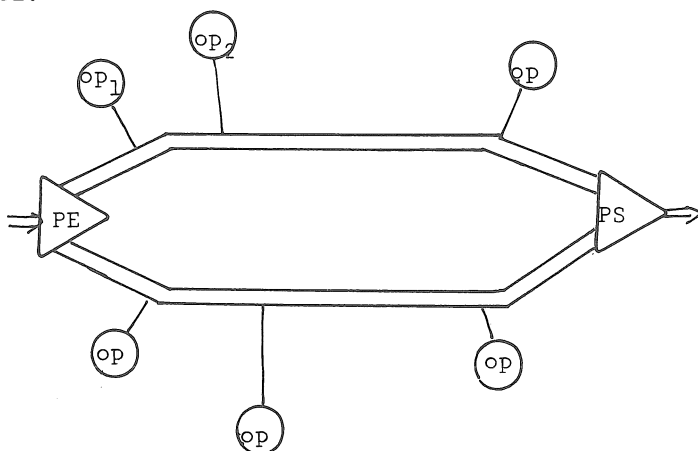


Figura 5. Estrutura do Sistema de Operadores (SO)

A rotina de operação do anel baseia-se na locação de todo o anel para permitir o envio de uma operação para um operador. Quando uma operação (instrução, operadores e destino) chega ao sistema de operadores ela é armazenada na porta de entrada (PE) do anel a espera da liberação de um operador. Quando um operador e o anel estiverem livres ele, o operador, faz

circular uma ficha com seu endereço. A porta de entrada do anel, ao detectar uma ficha de disponibilidade, retira essa ficha e começa a enviar, para o operador, a operação que tem armazenada.

A ligação entre a PE e o operador só é desfeita após a completa transmissão da operação. Enquanto uma ligação está feita nenhuma outra poderá ser conectada.

Ao acabar de executar uma operação, o operador espera a passagem de uma ficha indicativa de disponibilidade da porta de saída (PS) e do anel. Quando essa ficha é encontrada então o operador bloqueia o anel e faz a transmissão, para a PS, do resultado da operação. Novamente aqui, essa conexão só é desfeita após o término da transmissão.

Uma operação pode ter seu resultado enviado para várias operações destino, e nesse caso, o resultado é duplicado pelo próprio operador e enviado, um de cada vez, para cada uma das operações destinos. Para não bloquear demasiadamente o anel, nos casos de duplicação de resultados, após o envio de um resultado a conexão entre o operador e a PS é desfeita para permitir que outro operador assuma o controle do anel.

Existem dois operadores especiais dedicados à aplicação específicas, eles são os operadores de entrada e de saída de informação. É através desses operadores que os dados de entrada são introduzidos no fluxo de execução de cada programa assim como são retirados os resultados finais. Os operadores de entrada e saída estão ligados aos dispositivos padrões de entrada e saída de dados no meio externo. A existência de dois processadores distintos, um para entrada e outro para saída, deve-se à exigências de respostas rápidas feitas pelas aplicações em controle de processos e aquisição de dados. Com dois processadores independentes fica possível a entrada e

8. Formato dos Dados

A figura 6 mostra a estrutura dos dados que transitam dentro da arquiteturas. O campo de operando é formado por uma cadeia de caracteres (de 8 bits). O campo de código de operação é também uma cadeia de caracteres onde cada caracter corresponde a uma instrução. Notem que a arquitetura disporá de apenas 256 instruções diferentes. O indicador de instrução é formado por 3 caracteres os quais poderão fazer referência a programas com até 32M instruções (nós do grafo de fluxo). Com a possibilidade de se agregar várias instruções em um único nó podemos ter na verdade aquele número aumentado.

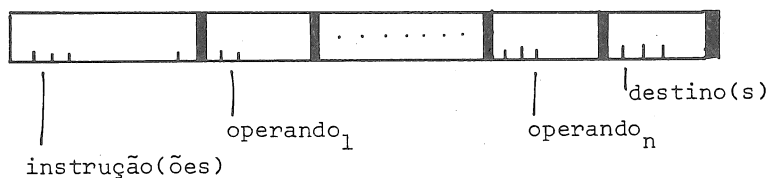


Figura 6. Formato dos Dados e Instruções

Cada operando carrega consigo, além dos valores propriamente ditos, um campo de descritores onde são indicados o nível de interação e a profundidade de chamada de procedimento. Outras informações contextuais podem ser anexadas

a esse campo de descritor. Certos subcampos são dedicados a aplicações específicas e são tratados diretamente pelos operadores (como é o caso dos indicadores de profundidade de chamada e de nível de interação). Os demais subcampos (tipicamente definidos pelos usuário) são ignorados pelos operadores e copiados na resposta.

REFERÊNCIAS BIBLIOGRÁFICAS

- CATTO, A.J., "Nondeterministic Programming in a Dataflow Environment", PhD thesis, University of Manchester, 1981.
- BACKUS, J., "Can Programming Be Liberated From the von Newmann Style? Functional Style and this Algebra of Programs", CACM, vol. 21, n. 8, agosto de 1978.
- DENNIS, J.B. et alii, "Data Flow Schema", in Lecture Notes in Computer Sciences, Springer Verlag, 1974.
- FOINA, P.R., "LINDA - Linguagem para Descrição de Algoritmos Paralelos", artigo técnico interno do departamento de Computação e Estatística da UFSCar, 1985.
-
- "Arquitetura e um Processador Funcional Usando uma Rede de Microprocessadores Dedicados", anais do VIII SEMISH, Florianópolis, julho de 1981. (b)
- GURD, J.R., MATSON, I., GLAUENT, J.R.W., "A Data Driven System for a High Speed Parallel Computing", Computer Design, vol. 19, n. 6 e 7, junho/julho de 1980.
- GURD, J. & WATSON, I., "A Multilayered Data Flow Computer Architecture", artigo técnico do Departamento de Computação da Universidade de Manchester, março de 1980.
- TRELEAVEN, P.C., "Data-Driven and Demand-Driven Computer Architecture", ACM Computing Surveys, vol. 14, n 1, março 1982.
- von NEUMANN, J. et alii, "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument", in TAUB, A.H., "Collected Works of John von Newmann", vol. 5, pp. 34-39, MacMillan, Company, NY, 1968