ANALISE DE DESEMPENHO E IMPLEMENTACAO DE ARQUITETURAS PARALELAS

Luciano da Fontoura Costa
Gonzalo Travieso
Jan Frans Willem Slaets

* Instituto de Fisica e Quimica de São Carlos - USP

Resumo

Este trabalho analisa três formas de implementação em "hardware" para o processamento de uma mesma operação indivisível sobre diversos valores considerando-se a eficiência na utilização das memórias, que é um indicativo da velocidade total de processamento. Entre as arquiteturas analisadas, uma consiste de um único processador, outra é o processador de "array" propriamente dito (veja, por exemplo, pg. 120 da referência [1]), e uma outra mostra-se a melhor das arquiteturas aqui analisadas quando algumas condições forem obedecidas.

1. Introdução

O desenvolvimento de novas tecnologias de circuitos integrados e o aprimoramento das tecnologias tradicionais vem aumentando o número de componentes por centímetro quadrado e diminuindo os tempos de resposta dos dispositivos em circuito integrado. Entretanto, esse avanço de velocidade ainda mantém os processadores convencionais, com arquitetura SISD incapazes de resolver diversos problemas onde uma imensa quantidade de dados deve ser processada em tempos muito pequenos e/ou repetidas vezes, caso típico de processamento de vetores. A solução para estes casos é o uso de mais de uma unidade de processamento (processadores programáveis ou dedicados) de modo a permitir processamento concorrente, implicando no uso de processamento paralelo e/ou "pipelined". Nosso trabalho busca a determinação de relações para a análise do desempenho de arquiteturas destinadas a aplicação de uma mesma operação indivisível sobre N valores de um vetor V, ou seja, de uma arquitetura SIMD considerando-se o número de unidades de processamento e memórias utilizados e o tempo de processamento. São feitas as seguintes hipóteses:

i - As memórias utilizadas permitem acesso a um único de seus elementos durante seu tempo de acesso Ta.
ii - Todas as memórias utilizadas em um processador tem tempos de acesso iguais entre si.
iii - Todos as unidades de processamento utilizadas possuem

tempo de processamento Tp iguais entre si

iv - A cada valor de V processado obtem-se, após Tp, um novo valor que deve ser armazenado.

Para cada arquitetura aqui analisada definiremos um índice de eficiência de uso da memória EM, calculado como a relação entre o mínimo tempo necessário para a utilização da memória pelo tempo total de processamento. Por exemplo, se N valores devem ser processados por uma unidade de processamento EM será dada por N Ta/(tempo total de processamento). EM variam entre zero e um.

Normalmente o que se busca é o menor tempo de processamento para dados Ta e Tp, que são reflexos do estado atual da tecnologia de circuitos integrados. O índice EM reflete a velocidade do sistema, de forma que quando se deseja máxima velocidade deve-se obter EM máxima, ou seja, EM igual a um. Quando esta condição é atingida os dados são processados em taxa máxima igual a 1/Ta.

Partiremos de uma arquitetura básica (tipo 1) com uma unidade de processamento e prosseguiremos com a análise de outros três arranjos.

Em nossas análises não consideraremos a distribuição dos dados pelas memórias de cada arquitetura.

2. Arquitetura Tipo 1

Utilizando-se uma única unidade de processamento UP com tempo de processamento Tp, e memórias com tempo de acesso Ta, a arquitetura mais eficiente considerando-se as hipótese feitas anteriormente, é conseguida empregando-se duas memórias, uma de leitura (ML) e outra de escrita (ME), de forma a permitir o paralelismo entre a entrada de um dado proveniente da ML na unidade de processamento UP e a atualização do último valor processado em ME. A figura 1 representa este tipo de arquitetura que denominaremos tipo 1.

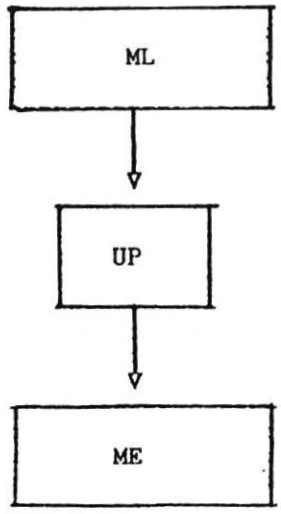

Figura 1 - Arquitetura tipo 1.

Assim, cada um dos N pontos de V armazenados na ML a serem processados deve ser acessado por UP, processado e atualizado em ME, onde após N processamentos de UP teremos o resultado. O tempo de determinação de cada endereço de busca e atualização são incluídos em Tp. A figura 2 apresenta o diagrama de tempo para a execução de N valores por este tipo de arquitetura.

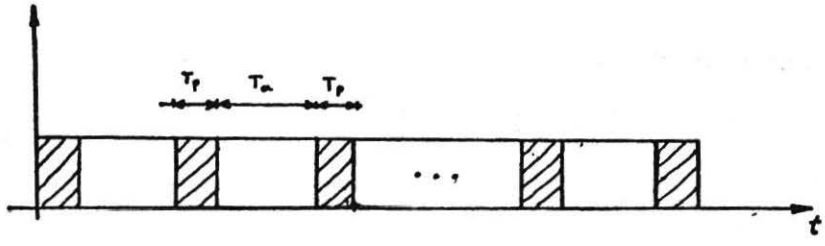

Figura 2 - Diagrama de tempo para a arquitetura tipo 1.

O tempo total de processamento deste tipo de arquitetura para os N valores de V (TTP1(N)) pode ser descrito por (2.1).

$$TTP1(N) = N\ (Ta + Tp) + Ta \qquad (2.1)$$

EM é dada pela razão entre o tempo ótimo, ou seja, N acessos em cada memória e o tempo total para o processamento dos N valores.

EM é igual para ML e ME e é dado por (2.2).

$$EM = \frac{N\ Ta}{N\ (Ta + Tp) + Ta} \qquad (2.2)$$

Quando N é muito grande EM é dada por (2.3).

$$EM = \frac{Ta}{Ta + Tp} \qquad (2.3)$$

Quando Ta >> Tp, ou Tp tende a zero, EM torna-se próximo de 1 indicando ótima utilização das memórias e o melhor tempo de processamento.

Por outro lado, se Ta << Tp EM tende a zero indicando péssima utilização das memórias.

Caso a operação a ser executada sobre os N valores pudesse ser dividida em níveis independentes podemos empregar a técnica de "pipelining" entre estes níveis a fim de diminuir Tp. Com N muito grande o tempo de processamento de um "pipeline" é dado pelo seu nível mais lento que é menor que Tp sem "pipeline", mas ainda diferente de zero. Este tipo de arquitetura não pode operar em taxa máxima.

3. Arquitetura Tipo 2

Uma outra forma de processarmos os N valores de V seria particionarmos seus N elementos entre k (k inteiro e maior que um) memórias MLi conectadas a k unidades de processamento UPi (i = 1,2,...,k) e utilizarmos k MEi formando k processadores tipo 1. A figura 3 ilustra este tipo de arquitetura denominada aqui como tipo 2.

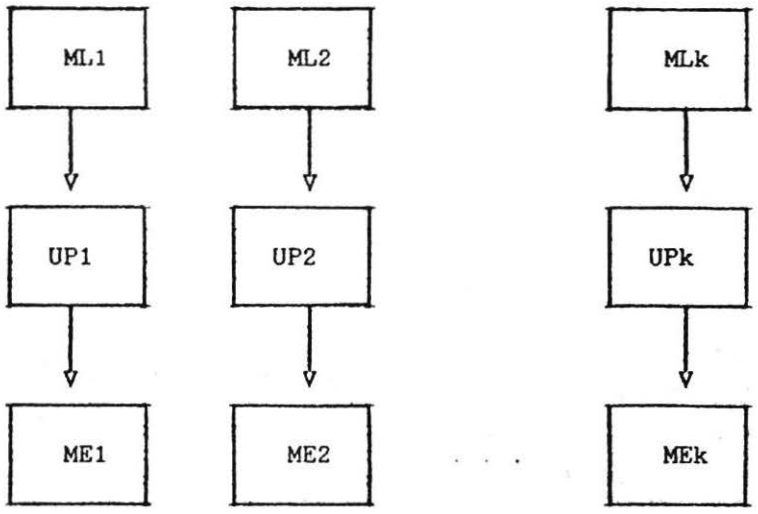

Figura 3 - Arquitetura tipo 2.

Considerando-se que cada unidade memória MEi possua ni valores, e nmax seja o maior entre os k ni, o diagrama de tempo deste tipo de arquitetura pode ser ilustrado como na figura 4.

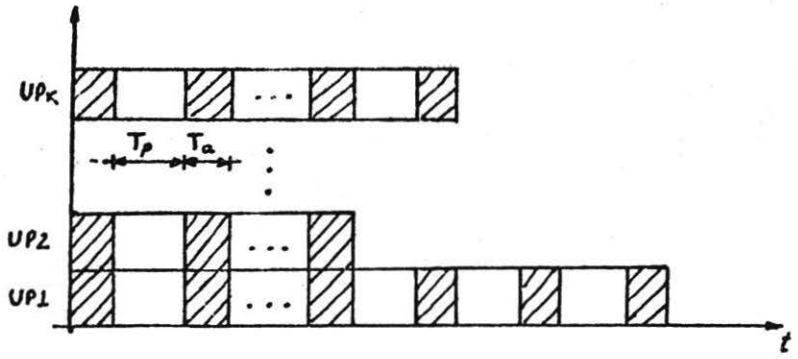

Figura 4 - Diagrama de tempos para arquitetura tipo 2.

O tempo total de processamento (TTP2(N,k)) será dado por (3.1).

$$TTP2(N,k) = nmax\ (Tp + Ta) + Ta \qquad (3.1)$$

Caso a partição dos N valores de V seja tal que N = nmax k indicando que a partição dos N valores de V é uniforme, (2.2) pode ser reescrita como (3.2).

$$TTP2(N,k) = \frac{N}{k}\ (Tp + Ta) + Ta \qquad (3.2)$$

Se tivermos partição uniforme, a eficiência EM desta arquitetura será dada (3.3).

$$EM = \frac{N/k\ Ta}{N/k\ (Tp + Ta) + Ta} \qquad (3.3)$$

Quando N é muito grande EM é dada pelas relação (3.4).

$$EM = \frac{Ta}{Tp + Ta} \qquad (3.4)$$

A máxima velocidade de processamento ocorre quando EM = 1, o que acontece apenas se Tp é zero, o que é obviamente impossivel.

O máximo número de processadores que podem ser utilizados, kmax, é igual a N e nmax = 1, ou seja, cada processador atua sobre apenas um dado, resultando distribuição uniforme e caracterizando processamento vetorial propriamente dito. Neste caso o tempo de processamento é dado por (3.5) e EM coincide com (2.3).

$$TTP2(N,k)\ minimo = 2\ Ta + Tp \qquad (3.5)$$

4. Arquitetura Tipo 3

Uma outra possivel arquitetura está mostrada na figura 5.

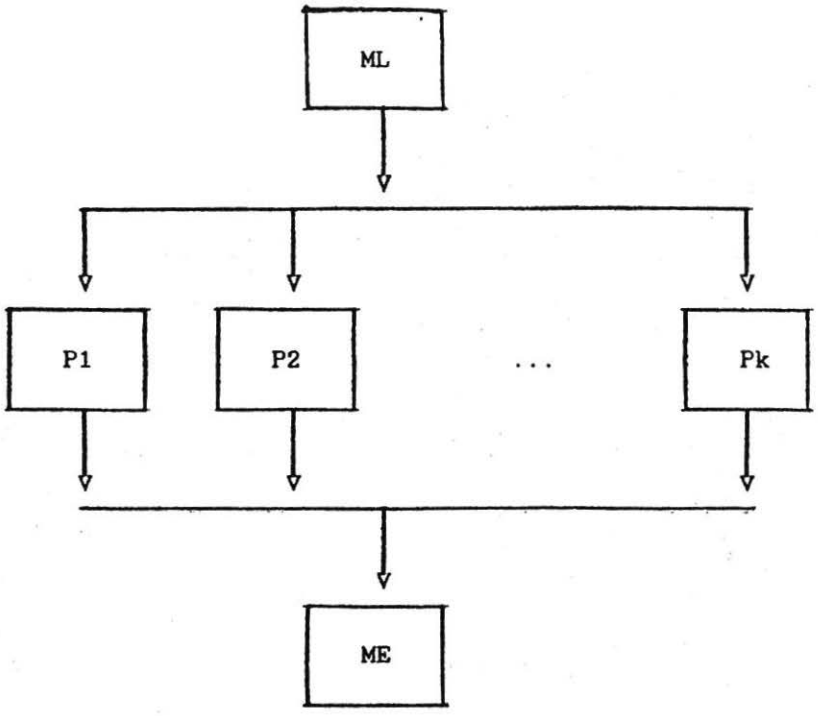

Figura 5 - Arquitetura tipo 3.

Aqui, cada um os N dados da ML são distribuídos pelos k processadores de acordo com sua disponibilidade em taxa máxima, ou seja, a cada Ta. Caso haja processador livre, um dado será liberado por ML, caso não hajam processadores livres não se libera dado de ML, esperando-se o próximo ciclo Ta. Este tipo de processamento só é permitido se Tp for um múltiplo inteiro de Ta garantindo a sincronização entre leitura em ML e escrita em ME a cada Ta (4.1).

$$Tp = q\ Ta \qquad \text{para algum } q = 1,2,\ldots \tag{4.1}$$

A figura 6 mostra um diagrama de tempo para este tipo de arquitetura com N = 12, Tp = 2 e Ta = 1.

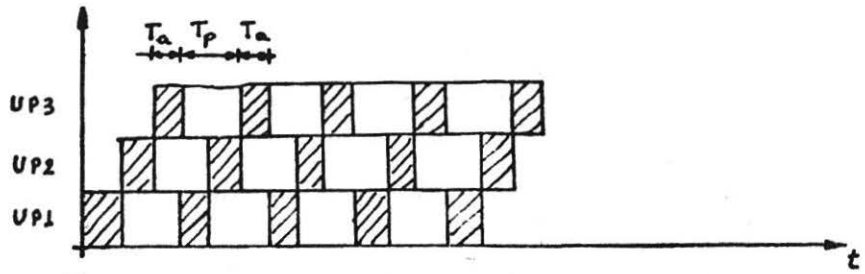

Figura 6 - Diagrama de tempo para arquitetura tipo 4 com N = 12, Tp = 2 e Ta = 1.

Se por simplicidade considerarmos N = j k (j = 1,2,...) o tempo total de processamento (TTP3(N,k)) será então dado por (4.2) ou alternativamente por (4.3).

$$TTP3(N,k) = \frac{N}{k} (Ta + Tp) + k\ Ta \qquad (4.2)$$

$$TTP3(N,k) = j\ Ta\ (1 + q) + k\ Ta \qquad (4.3)$$

A eficiência EM é dada por (4.4)

$$EM = \frac{N\ Ta}{N/k\ (Ta + Tp) + k\ Ta} \qquad (4.4)$$

Quando N é muito grande EM torna-se (4.5)

$$EM = \frac{k\ Ta}{(Ta + Tp)} \qquad (4.5)$$

A velocidade de processamento desta arquitetura é máximo quando tivermos EM igual a um o que implica (4.6).

$$k = \frac{Ta + Tp}{Ta} \qquad (4.6)$$

O valor de k de (4.6) é o limite máximo de processadores que podemos ter neste tipo de arquitetura e será denominado aqui como kmax.

Neste caso, (4.1) pode ser reescrita como (4.7)

$$TTP3(N,k) = (N+1)Ta + Tp \qquad (4.7)$$

5. Comparação entre as Arquiteturas Tipo 2 e 3

Nesta seção vamos comparar as arquiteturas tipo 2 e 3, considerando-se a condição (4.1) e que N = j k (para algum j = 1,2,...).

Para N muito grande a arquitetura tipo 3 com k processadores apresenta EM tendendo para k/(1 + q). Considerando-se a arquitetura tipo 2 com k processadores e N muito grande sua EM é. 1/(1 + q). Desta forma, EM da arquitetura tipo 3 será igual (quando k = 1) ou melhor (k = 2,3, ,kmax) que a EM da arquitetura tipo 2 para qualquer q · 1,2, Quando k = kmax EM da arquitetura tipo 3 é um.

A diferença (D23) entre o tempo total de processamento das arquitetura tipo 2 e 3 é obtida subtraindo-se (3.2) e (4.2) é dada por (5.1).

$$D23 = (k - 1)\ Ta \qquad (5.1)$$

Concluimos assim para um mesmo número k de unidades de processamento menor que kmax dado por (4.6) que quando (4.1) for verificada, a arquitetura tipo 3 apresenta em relação a arquitetura tipo 2 melhor EM e velocidades de processamento muito próximas quando D23 é pequena (caso bastante comum).

Ainda, caso não se verifique distribuição uniforme (N = k nmax) na arquitetura tipo 2, podemos ter, dependendo da variação entre os ni, uma ineficiência de processamento consideravelmente grande. Este problema pode ser resolvido com o uso de unidades de processamento UPi com tempos Tpi diferentes entre si de tal forma que verifiquemos a relação (5.2).

$$n1\ Tp1 = n2\ Tp2 = \ .. = nk\ Tpk \qquad (5.2)$$

Esta relação é facilmente determinada ao igualarmos todos os tempos totais de processamento de cada conjunto MLi - UPi- MEi dados pela relação (2.1) e só é viável quando a partição de N entre as k ML seja a mesma a cada processamento, caso contrário, a eficiência será reduzida, implicando também na utilização de processadores de tecnologias diferentes.

Uma alternativa para estes casos seria mapearmos as k MLi da

arquitetura tipo 2 como uma única memória ML e as k ME como uma única ME, constituindo uma arquitetura tipo 3. Agora, mesmo com variações nas partições de N, podemos assegurar sempre a melhor eficiência em relação à arquitetura tipo 2 desde que a relação (4.1) seja verificada.

6. Exemplo de Aplicação

Seja o caso em que 1000 valores necessitem ser multiplicados por um real a. Consideremos também que disponhamos de multiplicadores com tempo de execução Tp de um microsegundo e memórias com tempo de acesso Ta igual a 100 nanosegundos. Se implementarmos um processador tipo 2 com k = 5 e tivermos os N valores uniformemente distribuidos entre as k ML (200 valores por ML) teremos o seguinte tempo de execução calculado segundo (3.2):

$$TTP2(100,5) = \frac{1000}{5} (1\ us + 100\ ns) + 100\ ns = 220,1\ us$$

Se utilizarmos um processador tipo 3 com o mesmo número de processadores (k = 5) teremos por (4.2):

$$TTP3(1000,5) = \frac{1000}{5} (1\ us + 100\ ns) + 500\ ns = 220,5\ us$$

Observa-se neste caso uma variação pequena do tempo total de processamento entre as implementações tipo 2 e 3, o que sempre se verifica para o caso de Ta << Tp (caso tipico).

Consideremos agora que os 1000 valores a serem processados sejam o resultado de um outro processador que os libera diversas vezes com distribuição não uniforme entre as 5 ML da arquitetura tipo 2. Seja a distribuição entre as 5 ML para um processamento dos 1000 valores dada como:

ML1 contém 100 valores
ML2 contém 300 valores
ML3 contém 500 valores
ML4 contém 50 valores
e
ML5 contém 50 valores

Agora nmax é igual a 500 e o tempo total para processamento de um vetor de 1000 pontos será, por (3.1):

TTP2(1000 5. · 500 1 us + 100 ns) + 100 ns 550,1 us

O mesmo processamento realizado pelo processador tipo 3 seria igual ao caso em que havia distribuição uniforme, ou seja, 220,5 microsegundos e o custo dos dois processadores seria o mesmo considerando-se memória e processadores.

7. Um Modo Eficiente de Endereçamento para a Arquitetura Tipo 3

O endereçamento dos dados em ML e ME pode ser implementado de duas maneiras. Na primeira, que denominaremos endereçamento interno, cada unidade de processamento UPi deve providenciar seus endereços de leitura e escrita e o tempo para isto deve ser incluído em Tp. Por exemplo, se a varredura de ML e a atualização em ME for sequencial, cada unidade de processamento UPi pode utilizar como endereço o seu índice, acrescentado o número de processadores a este a cada valor calculado. A segunda forma de endereçamento, aqui denominada de endereçamento externo, utiliza um circuito externo para geração dos endereços. Este circuito pode ser formado por dois geradores de endereço G1 e G2 sincronizados por um relógio CK responsáveis pelos endereços de leitura e escrita respectivamente. A figura 9 ilustra o esquema de endereçamento externo para a arquitetura tipo 3.

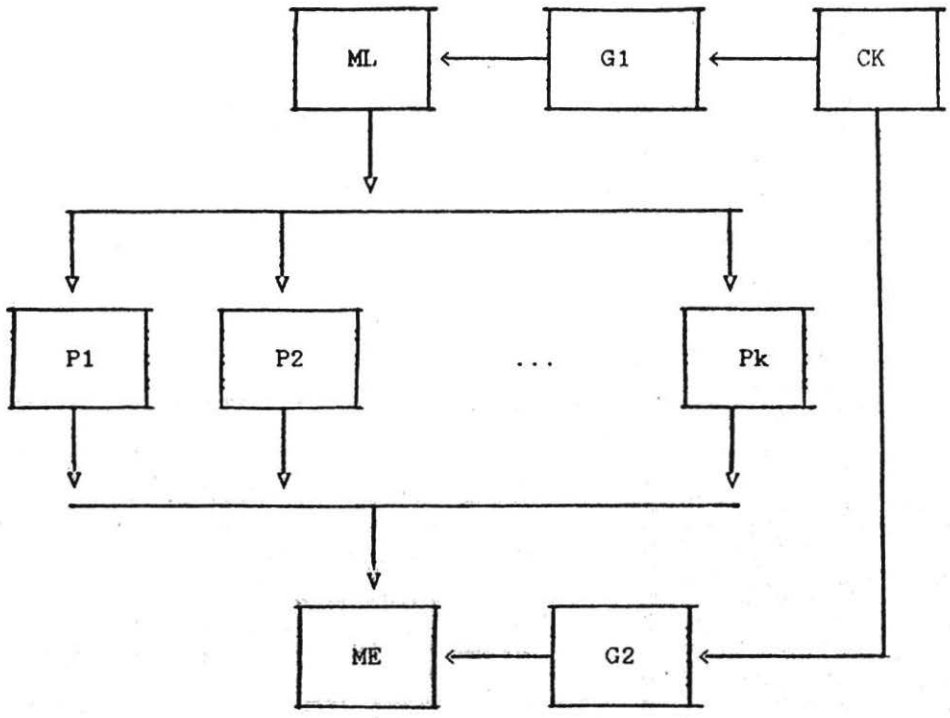

Figura 7 - Esquema de endereçamento externo para a arquitetura tipo 3.

Caso a varredura seja sequencial G1 e G2 podem ser contadores sendo que o início de contagem de G2 deve ser retardado por k ciclos Ta em relação a G1 (k é o número de processadores). O endereçamento externo possibilita a economia de espaço por utilizar apenas um circuito de endereçamento e se os tempo para geração do endereço de leitura, Tg1, e de geração de endereço de escrita, Tg2, puderem ser realizados num esquema como na figura 8, teremos um ganho de velocidade pois Tp deixa de incluir a parcela referente ao endereçamento.

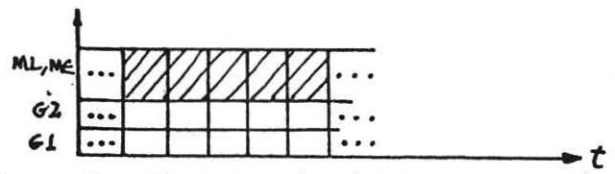

Figura 8 - diagrama de tempo para endereçamento na arquitetura tipo 3.

Observa-se que a arquitetura tipo 2 não se mostra adequada ao endereçamento externo quando não se verifica distribuição uniforme, pois devem existir k dutos de endereço, cada um exclusivo de seu processador

8. Conclusões

A implementação em "hardware" de processadores destinados á aplicação de uma determinada operação sobre N elementos segundo as hipóteses da seção 1 foi analisada segundo três opções. Em cada caso buscamos a velocidade máxima de processamento que era uma consequência de EM igual a um e que significa que os dados estão sendo processados em taxa máxima de liberação e atualização de dados nas memórias. A arquitetura tipo 1 é a célula básica das outras e para ela EM igual a um não é atingível pois implica em Tp zero. O mesmo acontece para a arquitetura tipo 2. Já a arquitetura tipo 3 possibilita EM igual a um. Esta arquitetura apresenta em relação á arquitetura tipo 2 tempo total de processamento um pouco maior quando os N valores a ser processados estiverem uniformemente particionados entre as k ML e apresenta velocidade muito superior para partição não uniforme.

A escolha entre as arquiteturas 2 e 3 para obtencão de velocidade máxima quando ocorre partição uniforme deve ser feita baseada na maneira que os dados a serem processados chegam ao sistema: se estes N valores chegam sequencialmente (por exemplo se estiverem armazenados numa memória única) a arquitetura tipo 3 deve ser escolhida, caso os N valores cheguem em paralelo ao sistema de processamento (especialmente em métodos iterativos vetoriais) devemos utilizar a arquitetura tipo 2.

Foram ainda determinadas neste trabalho fórmulas para determinarmos a velocidade de cada uma das arquiteturas estudadas e condições para máxima eficiência para a arquitetura tipo 3.

Esta análise aqui realizada pode ser estendida para processadores que possuam unidades de processamento com tempos diferentes de execução ou quando operações diferentes devam ser realizadas sobre o vetor V. Um próximo trabalho tratará destes casos.

9. Referências Bibliográficas

[1] - YOVITS, C. M. "Advances in Computers" volume 20. Academic Press (1981).

[2] - Special Issue on Performance of Multiple Processor Systems. IEEE Transactions on Computers, vol. C-32, nro. 1, janeiro de 1983.

[3] - Special Issue on Parallel Processing. IEEE transactions on Computers, vol C-34, nro. 10, outubro de 1985