Eficiência em Processamento Pipelined-Paralelo

Luciano da Fontoura Costa
Jan Frans Willem Slaets

Instituto de Fisica e Quimica de São Carlos - USP

1. Sumário

Este artigo apresenta um estudo para a síntese de processadores pipelined-paralelos dedicados a algoritmos de fluxo invariável com eficiência máxima. Incluímos uma aplicação para o projeto de um módulo "butterfly" para FFT (Fast Fourier Transform) com elementos de processamento com taxas de execução diferentes.

2. Introdução

Este trabalho apresenta as condições necessárias para que um processador "pipelined"-paralelo dedicado a execução de um algoritmo de fluxo invariável (de um algoritmo que não inclua tomadas de decisão) tenha eficiência máxima, ou seja, que seus elementos de processamento não permanecem períodos de tempo desocupados. Embora a referência [1] tenha trabalhado este problema, as hipótese assumidas foram muito restritivas (como a suposição que os diversos elementos de processamento do sistema tenham tempos de execução iguais entre si) e as verificações das condições obtidas, incorretas, foram realizadas de maneira inconsistente. Além disto foram realizadas formalizações desnecessárias. Nosso trabalho buscou a determinação de condições mais gerais (permite-se tempos de execução diferentes entre os elementos de processamento) para a obtenção de processadores de máxima eficiência e a verificação destas condiçõoes de maneira consistente.

3. Suposições

Todo o desenvolvimento deste trabalho assume as seguintes hipóteses:

i - Um algoritmo constitui-se de um número finito de operações e pode ser dividido em n-niveis.

ii - Um nivel do algoritmo constitui-se por todas as operações que podem iniciar suas execuções em um mesmo instante.

iii - Uma implementação em hardware de tal algoritmo consiste em um processador pipelined-paralelo dedicado com n-estágios correspondentes aos n-niveis do algoritmo.

iv - Os elementos de hardware destinados ao processamento das operações do algoritmo, os operadores, atuam de forma continua durante cada execução do algoritmo e

possuem taxas de operação não necessariamente iguais. Permite-se operação em paralelo entre os operadores de um dado nivel, assim como a multiplexação de operadores (o número de operadores em cada estágio deve ser maior ou igual a um e menor ou igual ao número de operações do nivel do algoritmo correspondente).

v - Assumimos operação continuada do processador, ou seja, o número de execuções do algoritmo tende ao infinito.

4. Condição para Máxima Eficiência

A eficiência do processador, quando o número de suas execuções tende ao infinito, é um se e somente se para cada estágio i do processador (i=1,2,...,n), verificarmos:

$$\frac{F_{i,j}}{p_{i,j}} \; t_{i,j} = T \qquad (1)$$

para qualquer tipo j de operadores (soma, produto, etc.) no estágio ($j=1,2,\ldots,k_i$) e

$F_{i,j}$ = número de operações do tipo j no nivel i do algoritmo.

$p_{i,j}$ = número de operadores do tipo j no estáagio i do processador.

$r_{i,j}$ = taxa de processamento em operações por segundo - ops - dos operadores do tipo j no estágio i do processador.

k_i = número de tipos de operações no nivel i.

T = um número real qualquer (o ciclo basico do pipeline).

5. Verificação

A - Consideremos um processador pipelined o mais geral possivel de N operadores com respectivos tempos de execução Ti a cada execução do algoritmo. A figura 5.1 ilustra o diagrama de tempo para M execuções de um destes processadores.

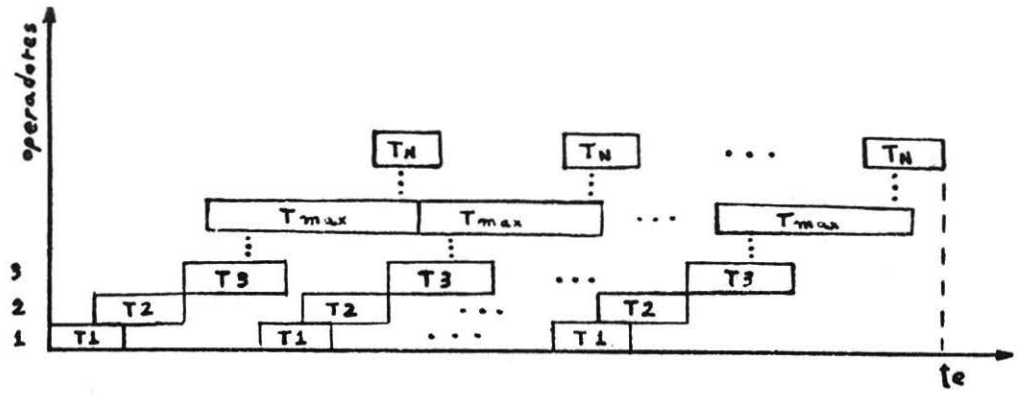

Fig. 5.1 - Diagrama de tempo para M execuções de um processador pipelined geral.

O tempo para M execuções do algoritmo pelo processador é te e Tmax é o tempo de execução do operador mais lento, ou seja:

$$Tmax \quad >= \quad Ti \qquad \text{para qualquer } i=1,2,\ldots,N$$

A eficiência (E) de tal processador pode ser descrita como:

$$E = \frac{Ra}{Rmax} = \frac{\text{taxa real de execução}}{\text{taxa máxima de execução}} \qquad (2)$$

Podemos escrever:

$$Ra = \frac{M\,N}{te} \qquad e \qquad Rmax = \sum_{i=1}^{N} \frac{1}{Ti}$$

Se Ta é o tempo entre o início da execução do processador até o início da operação do operador mais lento (com tempo de Tmax a cada execução do algori'tmo) e Td é o tempo entre o fim da execução deste operador até o término das M execuções do processador, podemos expressar o tempo total de execução como:

$$te = Ta + M\,Tmax + Td \qquad (3)$$

Deste modo:

$$\lim_{M\to\infty} E = \frac{M\,N}{(Ta + MTmax + Td)\sum_{i=1}^{N}\frac{1}{Ti}} = \frac{N}{Tmax\sum_{i=1}^{N}\frac{1}{Ti}} \qquad (4)$$

Verifiquemos as duas situações possiveis para os tempos de execução Ti:

i - Todos os tempos são iguais, ou seja:

$$Ti = Tmax \qquad \text{para qualquer } i = 1,2,\ldots,N$$

Para este caso a expresão para a eficiência do processador (4) fica:

$$\lim_{M\to\infty} E = \frac{N}{\frac{Tmax}{Tmax}\,N} = 1$$

ii - Existem K tempos menores que Tmax entre os N tempos Ti do processador. Denominando estes K tempos menores que Tmax de Qj, ou seja:

$$Qj = Ti < Tmax \qquad \text{para } j = 1,2,\ldots,K$$

a expressão (4) fica:

$$\lim_{M\to\infty} E = \frac{N}{(N-K) + Tmax/Q1 + Tmax/Q2 + \ldots + Tmax/QK}$$

Podemos escrever:

$$S = Tmax/Q1 + Tmax/Q2 + \ldots + Tmax/QK > K$$

ou

$$S = K + x, \qquad x > 0$$

desde que:

$$Qj < Tmax \quad \text{qualquer } j = 1,2,\ldots,K$$

e

$$Tmax/Qj > 1 \quad \text{qualquer } j = 1,2,\ldots,K$$

Desse modo

$$\lim_{M \to \infty} E = \frac{N}{N - K + K + x} = \frac{N}{N + x} < 1 \text{ , desde que } x > 0$$

Pelos resultados obtidos em i e ii podemos concluir que a eficiência do processador será um se e somente se todos os tempos de execução Ti de seus operadores forem iguais entre si e iguais a Tmax.

B - Utilizando o resultado obtido na seção A podemos, agora, distribuir as operações de cada nivel do algoritmo entre operadores que constituirão os n estágios do pipeline de modo que os tempos de execução dos operadores em cada processamento sejam iguais entre si. E claro que o tempo de execução de cada estágio do pipeline, que denominaremos T, deve ser um múltiplo inteiro de t_{max}, o tempo de execução do operador mais lento, ou seja:

$$T = I\, t_{max} \qquad \text{onde } I \in \{1,2,\ldots\}$$

Deste modo, os demais operadores de cada estágio devem ser alocados de forma tal que:

$$\frac{F_{i,j}}{P_{i,j}} \frac{t_{i,j}}{} = I\, t_{max} = T$$

onde as variáveis possuem o mesmo significado descrito anteriormente. Essa condição pode ser reescrita de duas outras maneiras:

$$\frac{F_{i,j}}{P_{i,j}} \frac{1}{r_{i,j}} = I\, t_{max} = T \qquad (5)$$

ou

$$\frac{F_{i,j}}{P_{i,j}} \frac{t_{i,j}}{t_{max}} = I \qquad (6)$$

6. Aplicações

A relação (1) deve ser satisfeita para eficiência máxima, e depende do número de operadores de cada tipo em cada estágio e de suas correspondentes taxas de execução. Assim temos duas situações práticas:

i - A partir de um número pré-estabelecido de operadores de cada tipo em cada estágio, podemos determinar as respectivas taxas de execução para eficiência igual a um.

ii - A partir de operadores com taxa pré-estabelecidas, podemos buscar um determinado número de operadores de cada tipo. Este problema nem sempre tem solução para eficiência um, entretanto, podemos buscar a minimização da seguinte função de mérito FM:

$$FM = \left| \frac{\sum_{i=1}^{n} \frac{F_{i,j}}{p_{i,j}} \frac{t_{i,j}}{I\ t_{max}}}{n} - 1 \right| \qquad (7)$$

para qualquer $i=1,2,\ldots,n$ e qualquer $j=1,2,\ldots,k_i$

sendo I um número inteiro qualquer.

A seguir veremos um exemplo de aplicação ao projeto do butterfly tipo decimação no tempo (DIT) para FFT.

Aplicação ao butterfly DIT

Consideremos o algoritmo para o butterfly DIT a seguir:

```
Tr = a * cos + b * sin
Ti = b * cos - a * sin
a = c - Tr
b = d - Ti
c = c + Tr
d = d + Ti
```

Os quatro valores de entrada a, b, c e d devem ser processados, gerando novos a, b, c e d. Os valores cos e sin são as partes reais e imaginárias da função exponencial complexa.

Temos cinco níveis no algoritmo e a distribuição das operações por nível é:

nivel 1 - 4 transferências de dados (entrada de a,b,c e d)
nivel 2 - 4 produtos
nivel 3 - 2 somas e 2 transferências de dados
nivel 4 - 4 somas
nivel 5 - 4 transferências de dados (saída de a,b,c e d)

de onde

$$F_{1,dt} = 4 \qquad F_{2,x} = 4$$

$$F_{3,+} = 2 \qquad F_{3,x} = 2$$

$$F_{4,+} = 4 \qquad F_{4,dt} = 4$$

Consideramos, para este exemplo, as seguintes taxas de execução para os operadores, válidas para qualquer nível:

soma -> $r_{+} = 10$ ops

produto -> $r_{x} = 40$ ops

transferência de dados -> $r_{dt} = 20$ ops

A aplicação da condição (5) nos dá:

$$\frac{F_{1,dt}}{p_{1,dt}} \cdot \frac{1}{r_{dt}} = \frac{F_{2,x}}{p_{2,x}} \cdot \frac{1}{r_{x}} = \frac{F_{3,+}}{p_{3,+}} \cdot \frac{1}{r_{+}} = \frac{F_{3,x}}{p_{3,x}} \cdot \frac{1}{r_{x}} =$$

$$= \frac{F_{4,+}}{p_{4,+}} \cdot \frac{1}{r_{+}} = \frac{F_{5,dt}}{p_{5,dt}} \cdot \frac{1}{r_{dt}} = T$$

ou seja:

$$\frac{2}{p_{1,dt}} = \frac{1}{p_{2,x}} = \frac{2}{p_{3,+}} = \frac{1}{p_{3,x}} = \frac{4}{p_{4,+}} = \frac{2}{p_{5,dt}} = T$$

a única solução é T=1, com FM = 0, com a seguinte distribuição:

$$p_{1,dt} = 2 \qquad p_{2,x} = 1$$

$p_{3,+} = 2$ $p_{3,x} = 1$

$p_{4,+} = 4$ $p_{5,dt} = 2$

e o diagrama de tempo para uma execução desta configuração do processador fica como mostrado na figura 6.1.

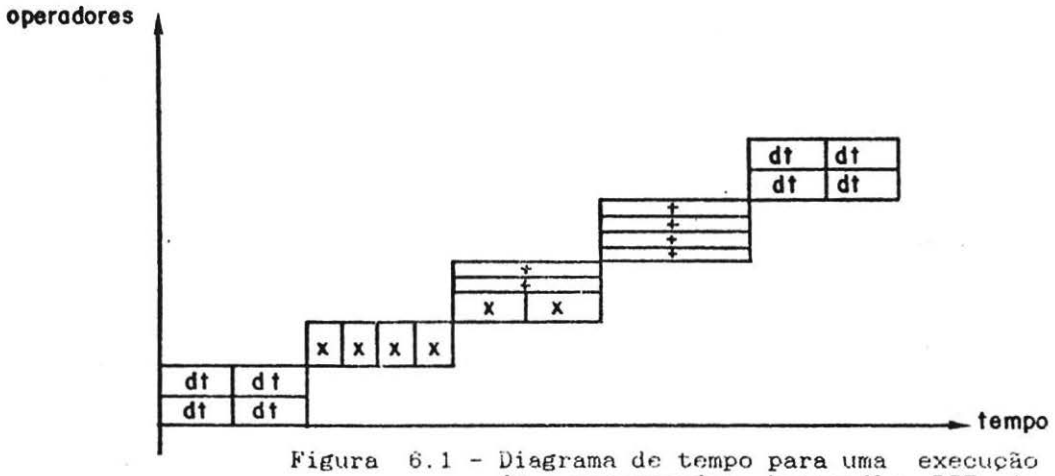

Figura 6.1 - Diagrama de tempo para uma execução do processador butterfly DIT do exemplo.

7. Conclusões

Foi apresentado um resultado que permite a busca da máxima eficiência de um processador pipelined-paralelo dedicado para o processamento de um algoritmo de fluxo invariável e um fator de mérito a ser otimizado para tanto. Estes resultados podem ser aplicados mesmo quando as taxas de execução dos operadores forem diferentes. Em relação ao trabalho da referência [1] podemos verificar que as hipóteses aqui adotadas são menos restritivas, nossos resultados apresentam-se propriamente normalizados e a verificação de nossa condição para máxima eficiência foi efetuada de maneira consistente. Foi ainda apresentado um exemplo de aplicação no projeto de um módulo de processamento destinado a efetuar o "butterfly" decimação no tempo para o algoritmo da Transformada Rápida de Fourier (FFT).

8. Referências Bibliográficas

[1] - SIOMALAS, K. O. & BOWEN, R. "Synthesis of Efficient Pipelined Architectures for Implementing DSP Operations", IEEE Transactions in Acoustics, Speech, and Signal Processing, vol. ASSP 33, número 6, dezembro de 1985.