Capítulo 3

Exploração da mesogranularidade

A demanda por processadores de alta velocidade é sempre crescente, e força o arquiteto de computadores a a introduzir "pipelines" mais sofisticados e graus crescentes de paralelismo nas arquiteturas, desde os microprocessadores até os supercomputadores.

Conforme exposto anteriormente, o desempenho depende essencialmente do tempo de ciclo e do grau de paralelismo disponível. O tempo de ciclo dos processadores é em sua maior parte controlado pela tecnologia disponível; o arquiteto pode apenas tentar reduzir a complexidade do ciclo de controle, visando obter pequenas reduçes no valor do tempo de ciclo. Por outro lado, apenas a natureza dos programas limita o uso do paralelismo para elevar o desempenho.

Nesta tese, define-se o microgranularidade como o paralelismo potencial que pode ser explorado em um mesmo fluxo de controle, mesogranularidade como o paralelismo potencial que pode ser explorado através de fluxos de controle independentes, com alto grau de interligaçåo e sincronizaço. Este último caso é analisado neste capítulo. Define-se macrogranularidade como o paralelismo potencial que pode ser explorado através de fluxos de controle independentes, exigindo um baixo grau de interligaçåo; este tópico será objeto do próximo capítulo.

Em outras palavras, a exploração da mesogranularidade exige uma alta taxa de comunicação entre os processos, com operações de sincronização realizadas frequentemente (da ordem de uma operação de sicronização a cada 100-1000 ciclos), enquanto a macrogranularidade só exige estas operações apos a realização de milhares de ciclos.

Um caso particular importante de exploração de mesogranularidade é a execuço replicada de um mesmo bloco de código, com cada fluxo de controle controlando a execução de uma parte de uma malha. Este caso é muito mais simples de detetar automaticamente e de controlar que os corrrespondentes à macrogranularidade.

Este capítulo mostra que o desempenho pode ser substancialmente melhorado pela exploraço da mesogranularidade, mesmo que se mantenha o mesmo total de recursos de hardware que uma soluço equivalente que não a explore. Mais importante, apresenta ainda as principais características arquitetônicas necessárias para seu suporte.

Conclue-se que mediante a exploração da mesogranularidade é possível obter um desempenho superior com os mesmos recursos de hardware. Este resultado possui significado muito importante, uma vez que, como demonstrado em [Jou89a], existe uma equivalencia funcional entre as arquiteturas VLIW, superescalares e "superpipelined", e portanto é muito provável que o mesmo resultado seja válido para aquelas arquiteturas.

3.1 Determinação do paralelismo disponível

Este estudo foi conduzido com o objetivo de determinar o número ótimo de grupos a incluir em cada processador. A arquitetura básica já estava definida, as restriçes de implantaço determinadas e o objetivo foi determinar como distribuir as unidades funcionais em grupos, bem como determinar quais as caracterís-

ticas do projeto que teriam de ser modificadas para suportar a operaço independente dos grupos.

Os limites para o paralelismo disponível tem sido bastante estudados na literatura, com resultados extremamente discrepantes, como se pode ver na tabela abaixo, baseada parcialmente em [Smi89]:

Estudo	Ano	Aumento de desempenho
Weiss/Smith	1984	1.58
Tjaden/Flynn	1970	1.8
Sohi	1987	1.8
Acosta et al	1986	2.7
Kuck et al	1972	8
Reisman/Foster	1972	51
Nicolau/Fisher	1984	90
Smith et al	1989	~2
Jouppi/Wall	1989	~2 a ~5

Existem muitas razes para esta discrepância de resultados, mas a mais importante é a variaço das hipóteses adotadas pelos diversos autores. Afinal, é sempre possível obter qualquer grau de paralelismo desejado; como exemplo, a malha abaixo pode ter pelo menos o grau de paralelismo N, para qualquer N:

```
      DO 1 I = 1, N
1     X(I) = A(I)
```

Entretanto, é irrealista esperar que uma arquitetura horizontal (ou qualquer outra) possa ter um incremento de desempenho linear ("speedup") pela replicaço pura e simples de unidades funcionais, devido a restriçes de aceso à

memória, tanto por parte dos dados quanto por parte de acesso à instruço, que seria enorme, isto sem discutir a factibilidade dessa máquina. A concluso importante é que é muito fácil obter medidas irrealistas do grau de paralelismo.

O que é realmente importante é o grau de paralelismo que pode ser obtido em aplicaçes práticas, sob hipóteses razoáveis em relaço ao "hardware" subjacente.

Para uma arquitetura de memória compartilhada, no é realista esperar-se mais que alguns acessos por ciclo (por processador), e este fator tem de ser levado em consideração para a obtenção de resultados significativos.

Assim, torna-se mais importante a elvação de desempenho que se poder obter em uma arquitetura em estudo pelo acréscimo de novas unidades de um certo tipo do que uma medida abstrata do grau de paralelismo existente no código a compilar. Quando o aumento de desempenho obtido é pequeno, pode-se concluir que o grau útil de paralelismo foi atingido. Este critério permite levar em consideraço todas as restrições impostas pela realização física da arquitetura proposta.

A utilização deste critério exige alguns cuidados, uma vez que acrescentar algum tipo de recurso a uma arquitetura sempre trara benefícios para a execuço de um particular programa. Para o presente estudo, foram adotados os Lawrence Livermore Kernels [Mah86], considerado o benchmark mais significativo para avaliaçao de arquiteturas.

3.2 Metodologia e resultados

Para estudar o dimensionamento da arquitetura e investigar a conveniência da exploraço da mesogranularidade foram utilizados 22 dos 24 núcleos dos Lawrence Livermore Kernels; os núcleos 14 e 22 foram eliminados para se evitar consideraçes de preciso com as funçes envolvidas.

O subsistema de memória já havia sido dimensionado para suportar dois acesso para leitura e um para escrita a cada ciclo de 10 ns. Para obter resultados realistas, foi realizado um estudo das características do subsistema de memória, para avaliar a reduço em seu desempenho real pela existência de conflitos de acesso. A simulaço mostrou que a influência desses conflitos é bastante baixa, da ordem de 10% para o caso máximo de duas leituras e um escrita por ciclo.

Foram adotadas as seguintes caracteríticas para o comportamento do subsistema aritmético:

- todas as operaçes (exceto diviso e raiz quadrada) so pipelined, com tempo de latência 5, e ciclo de 10 ns;
- as operaçes de diviso e raíz quadrada no sero pipelined, e tero tempos de latência de 25 e 45 ciclos, respectivamente;
- existe um registrador interno, que pode ser utilizado com um tempo de latência de 2 ciclos;
- as operaçes de acesso individual à memória tem um tempo de latência 15.

Em um primeiro estágio, o estudo foi conduzido para o caso de um único grupo, com 1 a 8 unidades aritméticas, explorando portanto apenas o microparalelismo.

As tabelas 3.1, 3.2 e 3.3, construidas a partir de [Alv90], apresentam os resultados obtidos. A tabela 1 mostra o desempenho individual dos núcleos para as diversas alternativas estudas. As tabelas 3.2 e 3.3 apresentam resultados agregados.

A tabela 3.2 mostra que há um aumento significativo no desempenho ao se passar de uma para duas unidades aritméticas, e pequenos ganhos a partir

Figura 3.1 - Desempenho dos núcleos individuais

Núcleo	1ua	2ua	4ua		8ua	
	1x 1ua	1x 2ua	1x 4ua	2x 2ua	1x 8ua	4x 2ua
1	83	166	242	311	431	431
2	66	133	193	249	345	345
3	50	187	187	187	187	187
4	50	187	187	187	187	187
5	50	50	50	50	50	50
6	66	187	187	187	187	187
7	88	177	320	355	528	674
8	76	180	348	348	621	621
9	94	185	313	313	313	313
10	40	84	84	84	84	84
11	50	50	50	93	50	93
12	50	93	93	93	93	93
13	48	132	221	236	237	248
14						
15	20	31	31	63	31	127
16	29	30	30	30	30	30
17	48	65	65	65	65	65
18	56	113	142	211	161	345
19	33	33	33	54	33	66
20	20	20	20	20	20	20
21	97	172	172	172	187	172
22						
23	52	64	64	64	64	64
24	16	20	20	20	20	20

A primeira linha do título indica o número total de unidades aritméticas utilizadas; a segunda, sua distribuição em número de grupos x unidades aritméticas por grupo. O desempenho é o assintótico e está corrigido para considerar os conflitos de acesso à memória.

Tabela 3.2 - Desempenho com exploração de microparalelismo

Unidades aritméticas	Desempenho assintótico em megaflops					
	Minimo	Média harmônica	Média Geométrica	Média aritmética	Máximo	Variação da média geométrica em relação ao valor anterior
1	16	42	48	49	97	
2	20	61	84	98	187	75
3	20	64	94	117	273	11
4	20	64	98	127	348	5
5	20	66	104	139	401	6
6	20	68	115	173	528	6
7	20	68	116	174	528	< 1
8	20	68	117	181	673	< 1

deste ponto. Resultados análogos foram obtidos em outros estudos realizados sob hipóteses semelhantes, para vários tipos de arquiteturas.

Esta concluso foi fundamental para a evoluço do projeto, por varias razões:

- tendo mostrado as limitaçes do microparalelismo, sugeriu ao grupo de projeto a necessidade de exploraço de outros tipos de paralelismo, motivando em parte o presente trabalho;
- permitiu fixar em dois o número de unidades aritméticas por grupo, o que leva a uma configuraço de register file com um número realista de portas e um instruço de tamanho pequeno (64 bits).

A tabela 3.3 mostra os mesmos resultados para configuraçes com 2, 4 e 8 grupos de duas unidades funcionais. Estas configuraçes exploram o micro e o mesoparalelismo existente nos programas, e mostra um aumento de até 1.54 no

Tabela 3.3 - Desempenho com micro e mesoparalelismo

Número de grupos com duas unidades aritméticas	Desempenho assintótico em megaflops					Aumento de desempenho em relação à configuração anterior (em %)
	Minimo	Média harmônica	Média geométrica	Média aritmética	Máximo	
1	20	61	84	98	187	-
2	20	74	111	142	364	33
4	20	78	129	184	674	16

desempenho, para os mesmos números máximo de acessos simultâneos à memória e número de portas no register file, quando comparado à configuraço com um único grupo, considerada como sendo a configuraço base. Tudo isto sem sacrificar o desempenho máximo nos casos em que ocorre, e com uma execução física simplificada.

3.3 Implementação do aproveitamento da mesogranularidade

3.3.1– Desacoplamento dos acessos à memória

3.3.2– Interrupções retardadas

3.3.3– Comunicação entre grupos