CAPÍTULO VI

CONSIDERAÇÕES FINAIS E CONCLUSÕES

VI.1 - Análise das Propostas de Arquiteturas por Número de Transistores

A fim de melhor avaliar o custo de cada EP proposto no Capítulo III, sem a influência do tipo de dispositivo utilizado para a implementação, optou-se por estimar o número de transistores necessários à integração em VLSI (com tecnologia CMOS) de cada arquitetura. Trata-se apenas de uma estimativa e não de um valor exato, pois o número exato de transistores só poderia ser obtido através do projeto em VLSI de cada proposta de EP, o que não é uma tarefa trivial. A idéia de estimar o número de transistores tem por objetivo exatamente fazer uma avaliação simplificada e comparativa da área de silício necessária à implementação de cada arquitetura, que em última análise equivale a uma análise comparativa do custo de cada proposta.

Para avaliar o número de transistores, inicialmente dividiu-se cada arquitetura em blocos básicos: registradores, somadores, unidades lógicas, multiplexadores, portas "E", registradores de deslocamento, etc. Depois avaliou-se o número de transistores utilizados para implementar cada um destes blocos básicos. Por fim, totalizou-se o número de transistores de cada arquitetura de acordo com os seus blocos básicos e o número de transistores de cada um deles. Esta é uma avaliação apenas aproximada, pois nem sempre os blocos estão otimizados, além do que o controle dos EPs está sendo desconsiderado. Esta última simplificação é perfeitamente justificável, visto que num encapsulamento de vários EPs num único *chip* o controle não precisa ser replicado, mas apenas os blocos básicos. Sendo assim, o custo do controle não é preponderante.

A Tabela VI.1 apresenta a estimativa do número de transistores necessários à implementação de cada bloco básico. A maioria das funções básicas foi estimada a partir dos circuitos apresentados em [WeEs85] e [Texa76]. Outras funções mais específicas foram deduzidas a partir da teoria encontrada em [WeEs85] e das funções já estimadas.

Nas Tabelas VI.2, VI.3, VI.4, VI.5 e VI.6 são apresentadas as estimativas do número de transistores necessários à implementação de cada EP, a partir dos seus blocos básicos. Para que a comparação seja justa, foi considerado que os quatro primeiros processadores possuem a mesma estrutura de comunicação de dados entre EPs vizinhos apresentada no Capítulo V para

o EP4BR. Isto acarretou no acréscimo de um multiplexador 4 X 1 e de quatro *buffers tri-state* de 1 *bit* em cada arquitetura. É apresentada também uma estatística do número de transistores usados na síntese da lógica e da memória interna de cada EP.

Tabela VI.1 - Número de transistores para implementação CMOS de funções básicas

Função	Transistores
"NAND" com n entradas	2n
"NOR" com n entradas	2n
"OR" com n entradas	2n
"XOR" com 2 entradas	8
Buffer tri-state de 1 *bit*	6
Multiplexador 2 X 1	6
Multiplexador 8 X 4	18
Multiplexador 4 X 1	16
Multiplexador 16 X 4	52
Multiplexador 6 X 1	26
Multiplexador 8 X 1	34
Multiplexador 32 X 4	118
Flip-flop tipo D	26
Flip-flop tipo D (c/ *SET* e *RESET*)	36
Unidade lógica de 1 *bit*	16
Unidade lógica de 4 *bits*	64
Somador completo de 1 *bit*	24
Somador de 4 *bits*	96
Somador de 4 *bits* com entrada C0/C4	110
Reg. de deslocamento progr. (até 31 *bits*)	824
Reg. de deslocamento de 4 *bits*	128
Barrel shifter de 4 *bits*	70
FIFO de 32 *bits*	1350
Banco de registradores de 32 *bits*	1218

Tabela VI.2 - Número de transistores para implementação CMOS do EP1

Qtde	Tipo	Total de transistores
1	Multiplexador 2 X 1	6
2	Multiplexador 4 X 1	32
4	*Buffers tri-state* de 1 *bit*	24
3	*Flip-flop* tipo D	78
1	*Flip-flop* tipo D (c/*SET* e *RESET*)	36
1	Unidade lógica de 1 *bit*	16
1	Somador de 1 *bit*	24
	TOTAL	216

Memória: 114 transistores (53%) Lógica: 102 transistores (47%)

Tabela VI.3 - Número de transistores para implementação CMOS do EP1SH

Qtde	Tipo	Total de transistores
4	Multiplexador 2 X 1	24
2	Multiplexador 4 X 1	32
4	*Buffers tri-state* de 1 *bit*	24
3	*Flip-flop* tipo D	78
1	*Flip-flop* tipo D (c/*SET* e *RESET*)	36
1	Unidade lógica de 1 *bit*	16
1	Somador de 1 *bit*	24
1	Reg. de deslocamento progr.	824
	TOTAL	1058

Memória: 938 transistores (89%) Lógica: 120 transistores (11%)

Tabela VI.4 - Número de transistores para implementação CMOS do EP1FIFO

Qtde	Tipo	Total de transistores
4	Multiplexador 2 X 1	24
2	Multiplexador 4 X 1	32
4	*Buffers tri-state* de 1 *bit*	24
2	*Flip-flop* tipo D	52
1	*Flip-flop* tipo D (c/*SET* e *RESET*)	36
1	Unidade lógica de 1 *bit*	16
1	Somador de 1 *bit*	24
1	FIFO de 32 *bits*	1350
	TOTAL	1558

Memória: 1438 transistores (92%) Lógica: 120 transistores (8%)

Tabela VI.5 - Número de transistores para implementação CMOS do EP4

Qtde	Tipo	Total de transistores
1	Multiplexador 16 X 4	52
2	Multiplexador 8 X 4	36
1	Multiplexador 4 X 1	16
2	Multiplexador 2 X 1	12
4	*Buffers tri-state* de 1 *bit*	24
9	*Flip-flop* tipo D	234
1	*Flip-flop* tipo D (c/*SET* e *RESET*)	36
1	Unidade lógica de 4 *bits*	64
1	*Barrel Shifter* de 4 *bits*	70
1	Somador de 4 *bits* c/entrada C0/C4	110
1	Reg. de deslocamento de 4 *bits*	128
	TOTAL	782

Memória: 398 transistores (51%) Lógica: 384 transistores (49%)

Tabela VI.6 - Número de transistores para implementação CMOS do EP4BR

Qtde	Tipo	Total de transistores
1	"OR" com 4 entradas	8
1	Multiplexador 16 X 4	52
4	Multiplexador 8 X 4	72
1	Multiplexador 6 X 1	26
1	Multiplexador 4 X 1	16
3	Multiplexador 2 X 1	18
4	*Buffers tri-state* de 1 *bit*	24
5	*Flip-flop* tipo D	130
1	*Flip-flop* tipo D (c/*SET* e *RESET*)	36
1	Unidade lógica de 1 *bit*	16
1	Unidade lógica de 4 *bits*	64
1	Somador de 4 *bits*	96
1	Reg. de deslocamento de 4 *bits*	128
1	Banco de registradores	1218
	TOTAL	1904

Memória: 1512 transistores (79%) Lógica: 392 transistores (21%)

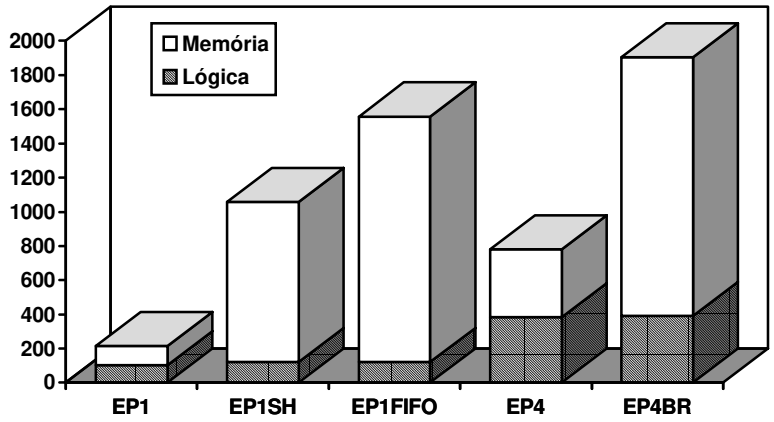

Figura VI.1 - Transistores necessários à implementação das arquiteturas

A Figura VI.1 apresenta um gráfico que resume as estimativas de custo para as diversas propostas de EP. Cada barra representa o número de transistores necessário à implementação de um processador, onde a área mais escura representa a lógica do EP e a área branca representa a memória do EP. Por este gráfico percebe-se que os dois grandes diferenciadores de custo são: a quantidade de memória interna do processador e o tamanho da palavra de trabalho do processador. Observa-se que a lógica depende basicamente do tamanho da palavra de trabalho. Já a memória interna não depende apenas do número de *bits* sintetizados. Observa-se que, apesar dos processadores EP1SH e EP1FIFO terem memórias internas de mesmo tamanho, o último tem um custo mais elevado, devido à complexidade da FIFO em

comparação com o registrador de deslocamento programável. Em compensação, a FIFO é mais versátil do que o registrador de deslocamento.

A Tabela VI.7 apresenta uma relação Custo X Benefício de cada processador para cada operação aritmética inteira. Foi considerado como custo a estimativa de transistores aqui apresentada e como benefício o inverso do número de ciclos de relógio gastos para a execução de cada operação (Tabela III.3). Para facilitar a comparação, a relação Custo X Benefício foi normalizada para o EP4BR, ou seja, esta relação foi considerada unitária para este processador em cada uma das operações. Sendo assim, para esta tabela, uma relação Custo X Benefício maior do que um representa um processador pior do que o EP4BR e vice-versa.

Tabela VI.7 - Relação custo X benefício normalizada para o EP4BR

Operação	nº de bits	ep1	ep1sh	ep1fifo	ep4	ep4br
Soma/Subtração	8	0,41	1,98	2,92	0,41	1,00
	16	0,43	2,09	3,08	0,41	1,00
	32	0,44	2,16	3,17	0,41	1,00
Multiplicação	8	0,71	1,53	2,25	0,89	1,00
	16	0,94	1,80	2,65	1,02	1,00
	32	1,07	1,90	2,79	1,06	1,00
Divisão	8	0,72	1,89	2,42	0,98	1,00
	16	0,95	2,09	2,76	1,10	1,00
	32	1,02	1,98	2,92	1,06	1,00

Para as operações de soma e subtração de inteiros, os processadores com pouca memória interna (EP1 e EP4) levam nítida vantagem sobre os processadores com maior quantidade de memória interna (EP1SH, EP1FIFO e EP4BR). Este resultado é previsível, visto que para estas operações não há utilidade para uma grande memória interna. Apenas três registradores (A, B e C) são suficientes.

Já nas operações de multiplicação e divisão de inteiros, os processadores de um *bit* com memória interna (EP1SH e EP1FIFO) perdem para todos os demais, pois o custo desta memória é muito elevado se considerado o benefício proporcionado pela mesma. Os outros processadores (EP1, EP4 e EP4BR) têm uma relação Custo X Benefício semelhante, variando ligeiramente de acordo com o número de *bits* dos operandos. Sendo assim e considerando-se que o desempenho do EP4BR é bem superior ao das quatro outras arquiteturas, conclui-se que a melhor proposta é o EP4BR, confirmando a escolha feita no Capítulo III.

VI.2 - Conclusões

Nas arquiteturas maciçamente paralelas, a maximização da relação Custo X Benefício do EP é de fundamental importância, pois estas arquiteturas baseiam seu desempenho na utilização de um grande número de processadores, o que só é possível se o custo individual de

cada processador for baixo. Por outro lado, um desempenho individual baixo pode comprometer a velocidade global da arquitetura. Portanto, estes dois fatores: custo e velocidade, devem ser bem pesados no projeto do elemento processador de uma arquitetura maciçamente paralela.

A implementação dos EPs em EPLDs torna ainda mais crítica a questão do custo do EP, dadas as limitações de um dispositivo programável. Estas limitações de custo levaram a opção por analisar apenas processadores de 1 e de 4 *bits*. Processadores de 8, 16 ou 32 *bits* são neste caso inviáveis, não só pelo custo da sua lógica, mas também pelo decorrente aumento do número de pinos para a comunicação com a memória externa. Esta última limitação não é uma característica apenas dos dispositivos programáveis, mas também de *chips* dedicados.

Este trabalho apresentou cinco propostas de arquiteturas para o processador: três EPs de 1 *bit* e dois EPs de 4 *bits*. O EP1 é um processador de 1 *bit* extremamente simples, com mínima capacidade de armazenamento interno e lógica de custo extremamente baixo. O EP1SH e o EP1FIFO são processadores de 1 *bit* com uma memória interna maior e lógicas semelhantes. O que os diferencia é a forma de implementar esta memória: enquanto o primeiro utiliza um registrador de deslocamento de tamanho programável, o segundo utiliza uma fila FIFO (*First In First Out*). Este acréscimo de memória interna não é útil nas operações de soma e subtração de inteiros, onde não há ganho de desempenho em relação ao EP1. Já nas operações de multiplicação e divisão de inteiros, esta memória é responsável por um desempenho cerca de três vezes melhor do que o do EP1. Porém, esta melhor desempenho não justifica o aumento de custo dos processadores.

O EP4 é um processador de 4 *bits* com pequena capacidade de armazenamento interno. O seu custo é cerca de quatro vezes mais elevado que o do EP1, porém o seu desempenho também é cerca de quatro vezes maior, para todas as operações avaliadas. Por último, o EP4BR é um processador de 4 *bits* com memória interna implementada através de um banco de registradores composto por oito registradores de deslocamento de 4 *bits* ligados entre si. A flexibilidade deste banco de registradores torna o desempenho desta proposta de processador bem superior às demais. Nas operações de multiplicação e divisão de inteiros, o seu desempenho é da ordem de doze vezes superior ao do EP1 e três vezes superior ao do EP4. Além disto, o seu custo é perfeitamente justificado pelo seu maior desempenho.

Sendo assim, o resultado da análise do custo e do desempenho das diversas arquiteturas propostas levou à escolha do EP4BR (Elemento Processador de 4 *bits* com Banco de Registradores), como o melhor processador dentre os cinco apresentados. O uso das operações aritméticas básicas sobre inteiros como exemplo de demanda de processamento, serviu não só para avaliar o desempenho de cada proposta, mas também para melhor entender as limitações de cada arquitetura e desta forma encontrar soluções para as deficiências de cada

uma delas. Através da análise dos problemas e soluções encontrados no desenvolvimento de cada algoritmo aritmético, as arquiteturas sofreram um processo evolutivo apresentado ao longo do Capítulo III. Eleito o EP4BR como a melhor arquitetura, este foi aprimorado através da análise do desenvolvimento de algoritmos para execução de operações aritméticas em ponto-flutuante. A discussão sobre a implementação dos algoritmos de soma e multiplicação em ponto-flutuante foi apresentada no Capítulo IV.

Os dispositivos de comunicação entre EPs e entre os EPs e a UC foram discutidos no Capítulo V. Optou-se por uma comunicação entre cada EP e seus oito vizinhos mais próximos, através de uma malha em X, composta por canais de 1 *bit*. Esta proposta tem custo semelhante ao de uma comunicação apenas com os quatro vizinhos mais próximos e é mais flexível. A comunicação entre os EPs e a UC é feita através de uma árvore *SUM-OR*, como na maioria das arquiteturas SIMD. O formato da palavra de controle do EP, também discutido no Capítulo V e apresentado no Apêndice B procurou não limitar as possibilidades de execução de operações simultâneas dadas pela arquitetura, tendo atingido este objetivo.

A fim de comprovar o funcionamento tanto das arquiteturas quanto dos algoritmos, foi desenvolvido um simulador lógico (SLAP) e um editor gráfico (EGHA) para captura de esquemáticos. Ambos foram escritos em linguagem C, num ambiente DOS. Os módulos que compõem o programa fonte do editor totalizam cerca de 8000 linhas e os do simulador totalizam cerca de 7700 linhas. Um outro objetivo da simulação foi o de comprovar o desempenho calculado para os algoritmos. Todos os algoritmos e arquiteturas propostos foram simulados.

Com relação às arquiteturas comerciais, as Tabelas III.4 e IV.7 mostram que o EP4BR é mais veloz que a maioria delas, perdendo apenas para o MasPar [Nick90]. Quanto ao MasPar, cada EP tem cerca de 14.000 transistores, sendo que 75% (10.500 transistores) são utilizados na implementação dos registradores internos e apenas 25% (3.500 transistores) são utilizados na implementação da lógica do EP. Fazendo uma análise superficial da relação Custo X Benefício do MasPar em comparação com a do EP4BR, percebe-se nítida vantagem deste último. A Tabela VI.8 apresenta uma relação Custo X Benefício dos dois processadores para algumas operações aritméticas. Foi considerado como custo do EP4BR a estimativa de transistores aqui apresentada (1.904 transistores) e do MasPar apenas os transistores associados à lógica do EP (3.500 transistores). Uma análise do custo total do EP do MasPar (14.000 transistores) levaria a uma vantagem ainda muito maior do EP4BR sobre o MasPar, mas poder-se-ia alegar que a grande memória interna do MasPar traz como vantagem uma menor dependência de comunicação com a memória externa. Sendo assim, resolveu-se dar esta vantagem ao MasPar na estimativa do seu custo.

Assim como na Tabela VI.7, o benefício é o inverso do número de ciclos de relógio gastos para a execução de cada operação e os valores estão normalizados para o EP4BR em cada operação. Deve-se considerar ainda, ao analisar esta tabela, que o desempenho estimado para o MasPar, pelo menos nas operações aritméticas inteiras, supõe que um dos operandos encontra-se num registrador especial e o resultado será armazenado neste mesmo registrador, o que é uma restrição importante. Este artifício equivale a supor que, no caso do EP4BR, um dos operandos encontra-se no banco de registradores A e o resultado também será colocado no banco de registradores. Isto acarretaria num aumento substancial da velocidade do EP4BR, principalmente na soma de inteiros, que teria um desempenho similar ao do MasPar a um custo bem inferior. Ou seja, a superioridade da velocidade do MasPar não justifica seu elevado custo, mesmo considerando-se o custo associado apenas à lógica do EP do MasPar contra o custo total do EP4BR. O MasPar só é bem superior ao EP4BR na operação de soma/subtração de inteiros e apenas devido ao artifício da medição, perdendo claramente nas operações de multiplicação tanto de inteiros quanto de ponto-flutuante.

Tabela VI.8 - Relação custo X benefício do MasPar e do EP4BR normalizada

Operação	MasPar	EP4BR
Soma/Subtração de inteiros de 8 *bits*	0,53	1,00
Multiplicação de inteiros de 8 *bits*	1,44	1,00
Soma/Subtração em ponto-flutuante com 32 *bits*	0,81	1,00
Multiplicação em ponto-flutuante com 32 *bits*	1,45	1,00

VI.3 - Perspectivas Futuras

A implementação em EPLDs é bastante interessante para o desenvolvimento de um protótipo. Porém, para uma versão definitiva, seria interessante a integração de vários EPs em um *chip*. Isto diminuiria o custo por EP e a área ocupada pela matriz de EP, permitindo a construção de uma matriz de EPs de maiores dimensões.

Numa análise superficial do número de transistores necessários à implementação de cada EP, poder-se-ia pensar que é possível integrar um número muito grande de transistores em um único *chip*. Porém, além das limitações de transistores, neste caso os *chips* possuem uma limitação maior que se refere ao número de pinos. Para o EP4BR, além dos *bits* de controle, que são comuns a todos os EPs, cada EP é responsável pela inclusão de pelo menos quatro pinos para a comunicação com a sua memória externa, fora os pinos gastos com a comunicação de dados entre EPs pertencentes a diferentes *chips*.

Uma solução para este problema é a utilização de uma memória interna maior. Por esta solução, todas as operações teriam operandos e resultados armazenados nesta memória. Além

disto, a memória externa seria compartilhada por vários ou todos os EPs de um *chip*, economizando pinos para comunicação com a memória. Para trocar dados entre as memórias interna e externa, passam a ser necessários vários ciclos, pois a memória externa tem que se comunicar com várias memórias internas. Sendo assim, seria melhor que a memória interna fosse dimensionada de forma a minimizar este tipo de comunicação e que esta comunicação fosse paralela a outros processamentos internos do EP.

Adotando um processador com uma memória interna grande, o custo da lógica passa a ter uma influência bem menor no custo total do processador. Sendo assim, poder-se-ia também cogitar a inclusão de circuitos que agilizassem algumas tarefas necessárias às operações de ponto-flutuante, como o alinhamento e o tratamento de exceções. Estas tarefas tem um fraco desempenho na versão atual do EP4BR, mas o uso de circuitos específicos foi descartado pois esta solução tem um custo muito alto relativamente ao custo total do processador.