Informática UFRGS

UFRGS UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

GME Grupo de Microeletrônica

FAURGS

CNPq

Uma ferramenta para geração automática de VHDL para filtros FIR otimizados

Fábio Fabian Daitx [1], Eduardo Costa [1, 2] - ffdaitx@inf.ufrgs.br, ecosta@ucpel.tche.br

Paulo Flores [3], Sergio Bampi [1] - pff@inesc-id.pt, bampi@inf.ufrgs.br

[1] *Grupo de Microeletrônica (GME) – Instituto de Informática – UFRGS – Porto Alegre/RS – Brasil*

[2] *Grupo de Microeletrônica e Processamento de Sinais da Escola de Engenharia e Arquitetura da UCPel – Pelotas/RS – Brasil*

[3] *Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento em Lisboa (INESC-ID) – TULisbon – Portugal*

1. Introdução

Os módulos de filtros FIR são importantes em sistemas de comunicação digital. Estes filtros podem ser realizados em hardware para aplicações de alta performance, ao invés de utilizar implementação "soft" em DSP. Na sua forma transposta os filtros FIR fazem a multiplicação de uma palavra digital por múltiplas constantes. Na implementação ASIC, a arquitetura paralela dedicada pode ser implementada fazendo as multiplicações por uma série de operações de somas e deslocamentos. Este trabalho propõe um padrão para representar as soluções para o problema da Multiplicação por Múltiplas Constantes (MCM -"Multiple Constants Multiplication") e implementa uma ferramenta capaz de gerar código VHDL de alto desempenho para filtros FIR, atuando como um compilador otimizador. Foram sintetizados diversos filtros para FPGAs da Xilinx. Assim, foi possível comparar diferentes filtros gerados nesse trabalho com os filtros gerados por uma ferramenta profissional com os mesmos propósitos, o Matlab.

2. Objetivos

Nesse trabalho foi dado enfoque ao interfaceamento necessário para adaptar os algoritmos propostos nos últimos anos (visando minimizar o número de operações de adição/subtração no bloco multiplicador dos filtros FIR) para implementações finais do hardware. Foi utilizanda uma sistemática automatizada para o desenvolvimento dos filtros. O objetivo final é a geração de uma descrição VHDL para os filtros, otimizando área, freqüência de operação e potência dissipada de forma a viabilizar a utilização desses circuitos como blocos constituintes de arquiteturas mais complexas.

3. Metodologia

O processo de geração de filtros adotado se baseia em um fluxo a partir de um conjunto de coeficientes característicos de um filtro FIR específico, previamente gerados segundo alguma das técnicas matemáticas de "janelamento" disponíveis para o cálculo da Transformada Discreta de Fourier (TDF). A metodologia adotada segue o fluxo abaixo ilustrado:

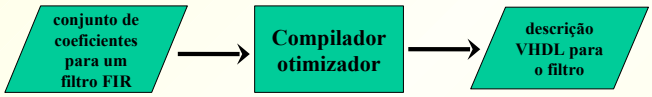

Figura 1 – Fluxo automatizado desenvolvido para o projeto de filtros FIR

Foi definido um padrão de representação para os arquivos de entrada, contendo o conjunto de coeficientes de um filtro, e para os arquivos internos, contendo as otimizações (realizadas segundo algum dos algoritmos propostos) utilizadas pelo compilador para gerar uma descrição VHDL de saída. O compilador completo foi emulado por uma ferramenta desenvolvida que simula sua funcionalidade. Assim, os códigos VHDL gerados são simplificados e legíveis, o que também ajuda a reduzir a complexidade do hardware final.

4. Arquitetura de Filtros FIR Otimizados

Em circuitos digitais, um filtro FIR ("Finite Impulse Response") pode ser visto como um bloco funcional que implementa a equação

$$y[n] = \sum_{i=0}^{N-1} h[i]x[n-i]$$

Nesse trabalho foi explorada a forma transposta de implementação de filtros FIR, que é a que permite maiores otimizações no bloco multiplicador, devido ao paralelismo existente nas operações. Utilizando o algoritmo proposto em [COS06], foi possível gerar, automaticamente, otimizações arquiteturais por somas e deslocamentos que simplificam significativamente a implementação das multiplicações, conforme mostrado abaixo.

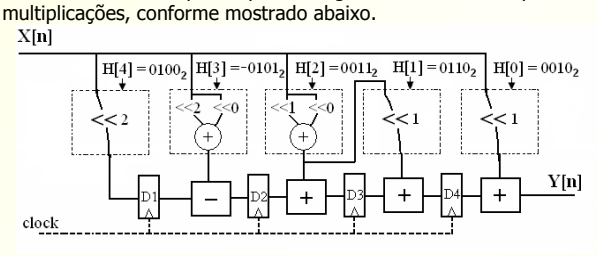

Figura 2 – Layout otimizado para um filtro FIR de 5 taps na forma transposta.

5. Resultados e Comparações

Foram gerados diversos filtros FIR (descrição VHDL) utilizando a ferramenta implementada. Esses filtros são apresentados na tabela 1. Também foram gerados códigos VHDL utilizando a feature Filters Design and Analysis Tool do Matlab, para os mesmos filtros. Os códigos VHDL produzidos foram sintetizados utilizando a ferramenta Xilinx ISE 9.1.01i direcionada para FPGAs da família Virtex2P. As comparações entre os resultados de síntese para as descrições VHDL geradas pelo Matlab e pela Ferramenta proposta são apresentadas na tabela 2.

Filter	Filter Specification						Filter Parameters		
	Design Method	Response Type	Fs	Fpass	Fstop	Order	input	width	#coef
1	FIR Equiripple	Lowpass	48000	9600	12000	50	16	16	26
2	FIR Equiripple	Lowpass	48000	7200	9600	50	16	20	26
3	FIR Equiripple	Lowpass	48000	4800	6400	76	16	14	39
4	FIR Window (Kaiser)	Lowpass	48000	4800	6400	151	16	16	73
5	FIR Equiripple	Highpass	48000	12000	9600	56	16	18	29
6	FIR Equiripple	Highpass	44100	12000	9600	50	20	18	26
7	FIR Least-squares	Highpass	44100	6615	4410	50	16	12	22
8	FIR Window (Hamming)	Highpass	48000	N/A	10800 (cutoff)	79	16	13	34

Tabela 1 – Especificação e parâmetros dos filtros FIR testados.

Filter	Matlab		Proposed Tool		Proposed Tool Improvements	
	# Add./Sub.	Max. Freq. (MHz)	# Add./Sub.	Max. Freq. (MHz)	# Adders/ Subtracters	Maximum Frequency
1	251	37.663	77	59.149	-69,32%	+57,05%
2	392	30.687	92	43.195	-76,53%	+40,76%
3	350	34.409	114	74.343	-67,43%	+116,06%
4	668	32.372	212	60.932	-68,26%	+88,22%
5	317	36.613	92	47.621	-70,98%	+30,07%
6	349	30.267	87	44.710	-75,07%	+47,72%
7	138	43.091	60	90.610	-56,52%	+110,28%
8	237	33.622	96	89.617	-59,49%	+166,54%

Tabela 2 – Resultados experimentais comparativos.

6. Conclusões

A ferramenta proposta obteve **sempre** o melhor desempenho e consumo de recursos. Isso se deve às otimizações feitas pelos algoritmos utilizados para eliminar somadores e subtratores redundantes, reduzindo o caminho crítico (profundidade) da árvore de somadores que implementa as multiplicações dos filtros. Com isso obtivemos ganhos de até 76,53% em número de somadores e 166,54% em freqüência de operação.

7. Referências

[COS 06] COSTA, Eduardo; FLORES, Paulo; MONTEIRO, José. **Exploiting General Coefficient Representation for the Optimal Sharing of Partial Products in.** In: Symposium on Integrated Circuits and System Design, Ouro Preto(Minas Gerais):Brazil, 2006.

[POT 96] POTKONJAK, Miodrag; SRIVASTAVA, Mani B.; CHANDRAKASAN, Anantha P. **Multiple Constant Multiplications: Efficient and Versatile Framework and Algorithms for Exploring Common Subexpression.** In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 15(2):151-165, 1996.

[HAR 96] HARTLEY, Richard I. **Subexpression Sharing in Filters Using Canonic Signed Digit Multipliers.** In IEEE Transactions on Circuits and Systems II, 43(10):677-688, 1996.

[FLO 05] FLORES, Paulo; COSTA, Eduardo; MONTEIRO, José. **An Exact Algorithm for the Maximal Sharing of Partial Terms in Multiple Constant Multiplications.** In International Conference on Computing Aided Design, San Jose(California):USA, 2005.

[PAR 01] PARK, In-Cheol; KANG, Hyeong-Ju. **Digital Filter Synthesis Based on Minimal Signed Digit Representation.** In Design Automation Conference, pages 468-473, 2001.

[XIL 07] Xilinx, Inc. <http://www.xilinx.com>.

Av. Bento Gonçalves, 9500 - Bloco IV - Campus do Vale - Caixa Postal 15064 - 91501-970 - Porto Alegre/RS