FIR FILTER DESIGN USING LOW POWER ARITHMETIC OPERATORS

Eduardo Costa
ecosta@inf.ufrgs.br
UFRGS
P. Alegre, Brazil

Sergio Bampi
bampi@inf.ufrgs.br
UFRGS
P. Alegre, Brazil

José Monteiro
jcm@algos.inesc.pt
IST/INESC
Lisboa, Portugal

Abstract. *This paper addresses the use of architectural transformations techniques for the low power realization of FIR filters on dedicated architectures. We experiment a new encoding for the operators, called the Hybrid encoding, which is a compromise between the minimal input dependency offered by the Binary encoding and the low switching characteristic of the Gray encoding. The results show that with the use of Hybrid operators in FIR architectures power savings of up to 25% are possible, together with a 14% delay improvement, and an area penalty of 28%.*

1 Introduction

There has been intensive research on design methods for low power [1]. The trend nowadays is to investigate techniques at higher levels of design abstraction as it has been recognized that the potential for power savings is much larger at these levels [2].

This paper focuses on power optimization techniques at the architectural level applied to Finite Impulse Response (FIR) algorithms. The reduction of the switched capacitance component is addressed by presenting arithmetic operators which use coding as a method of decreasing the switching activity. In this paper, the low power arithmetic modules are experimented in different dedicated FIR filter architectures. Combinations of fully-parallel, fully-sequential and semi-parallel architectures with simple, transposed and pipelined versions are explored.

Initial work on restructuring an RTL circuit description for low power was proposed by Chandrakasan [3]. In this work, transformations that reduce the number of computations were proposed. Various of these transformation techniques have been used in FIR filter realizations we present.

In the FIR filter operation, the output is performed by a summation of data-coefficient products. Thus, some techniques have addressed the use of coefficient manipulation in order to reduce the switching activity in the multipliers inputs [4, 5] by finding the Hamming distance between consecutive coefficients. The work proposed in this paper will build on some of the transformation approaches mentioned, specially the techniques that target the increase in performance and switching activity reduction.

2 Low Power Techniques

We propose the use of a Hybrid encoding for the operators, which is a compromise between the minimal input dependency exhibited by the Binary encoding and the low switching characteristic of the Gray encoding [6].

2.1 Hybrid Adder and Multiplier Architectures

The idea of the Hybrid code is to split the operands in groups of m-bits, encode each group using the Gray code and use the Binary approach to propagate the carry between the groups. Table 1 exemplifies the Hybrid encoding for 4-bit numbers and for m=2.

Table 1: Hybrid code representation for m=2.

Dec	Hyb	Dec	Hyb	Dec	Hyb	Dec	Hyb	Dec	Hyb	Dec	Hyb
0	0000	2	0011	4	0100	6	0111	8	1100	10	1111
1	0001	3	0010	5	0101	7	0110	9	1101	11	1110

We have found that the most efficient implementation for a Hybrid adder is in fact to use a Binary adder and make the conversion at the inputs and outputs. This is shown in Figure 1 for a 4-bit adder operator. Translating words of W bits at the inputs and outputs of an Hybrid adder with m=2, all that is required are $\frac{3}{2}W$ EXOR gates.

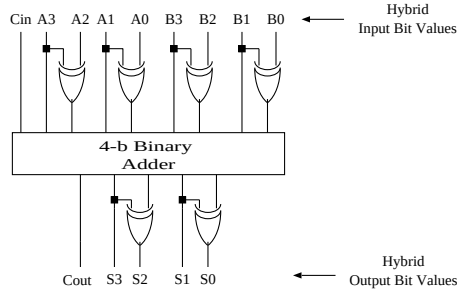

Figure 1: Hybrid Adder Architecture.

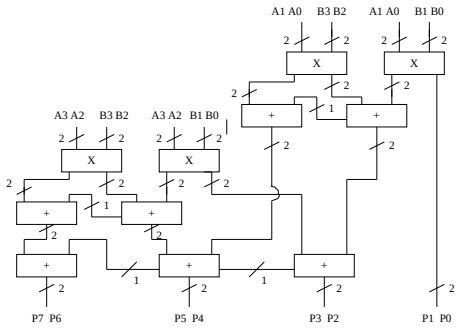

Figure 2: Hybrid Multiplier Architecture.

Contrary to the Binary multiplier where the operation is performed bit by bit, the Hybrid operation is performed for a m-bit group at a time. The full Hybrid architecture can be seen in Figure 2 for a 4-bit Hybrid array multiplier with m=2. As can be observed, it is necessary just one line composed by adders responsible for adding partial product terms. For a W-bit Hybrid architecture, ($W/2$)-1 of these adder lines will be used.

2.2 Coefficient Partitioning and Ordering

In our work we have experimented an extension of the coefficient ordering technique [4] in a Semi-Parallel architecture. Thus, the problem is related to finding the best partition for each coefficient by calculating the minimum Hamming distance between the coefficients into each group as shown in the pseudo-code presented in Figure 3.

3 Results

In this section we present the impact on power reduction through the application of low power techniques in dedicated pipelined FIR filter architectures. Area results are presented in terms of the number of literals. Delay results were obtained in SIS environment. Power results were obtained with the SLS tool [7] using the general delay model.

```
1.  for all permutations of coefficients H(0-7){
2.     partition1=Hamming((H[0],H[1]) + (H[1],H[2]) + (H[2],H[3]) + (H[3],H[0]));
4.     partition2=Hamming((H[4],H[5]) + (H[5],H[6]) + (H[6],H[7]) + (H[7],H[0]));
6.     cost function = partition1 + partition2;
7.     if cost function = minimum found{
8.        save current partition;
9.        minimum=cost;
10.    }
11. }
```

Figure 3: Algorithm for the generation of coefficient partitioning and ordering.

3.1 Hybrid Arithmetic Operator Application

The most efficient implementation for a Fully-Parallel Hybrid FIR filter is to use Binary adders and EXOR gates to make the conversion at the input/output data buses. Thus, there is no significant area difference between the architectures with Binary and Hybrid operators as shown in Table 2. In the Fully-Sequential and Semi-Parallel architectures, where array multiplier operators are used, Hybrid architectures present slightly more area.

Table 2: Area, delay and energy per sample results for the pipelined architectures.

	Area		Diff.(%)	Delay(ns)		Diff.(%)	Energy(μJ)		Diff.(%)
	Bin	Hyb	H→B	Bin	Hyb	H→B	Bin	Hyb	H→B
Par	19482	19578	+0.5	216.6	217.6	+0.5	24.5	25.5	+4.4
Seq	5808	7240	+24.6	272.6	233.5	-14.3	174.6	130.4	-25.3
Semi-Par	10055	12919	+28.5	271.6	232.5	-14.4	155.2	115.3	-25.6

Although the architectures with Hybrid operators present higher area, these architectures can work at a higher clock frequencies. This is due to the fact that the Hybrid multiplier presents a lower critical path. The direct application of Hybrid operators in the FIR architectures produces a reduction of energy per sample consumption as shown in Table 2. This is observed in the Fully-Sequential and Semi-Parallel architectures where array multipliers are used. The main reason for this is that the Hybrid multiplier presents a smaller logic depth than the Binary architecture, which reduces significantly the amount of glitching in the circuit.

3.2 Coefficient Manipulation

In the Semi-Parallel architecture, the performance is improved since the hardware can be operated in half of the clock cycles. Thus, we have experimented the application of our ordering and partitioning algorithm on this architecture. Table 3 shows the energy per sample results for the Pipelined Semi-Parallel architecture with Binary and Hybrid operators.

Table 3: Coefficients ordering and partitioning and coefficients with higher correlation.

Operators	Original Coefficients	Ordering and Partitioning	Difference(%)	Coefficients with Correlation	Difference(%)
Binary	155.12μJ	142.18μJ	-8.34	126.64μJ	-18.36
Hybrid	115.32μJ	111.41μJ	-3.39	100.43μJ	-12.91

As can be observed in Table 3, the filter architecture with Hybrid operator presents the least value of power consumption due to the lower critical path shown by this operator. Moreover, it should be observed that the Semi-Parallel architecture can operate at half of clock cycles and thus, the supply voltage can be reduced by a 2 factor and thus its energy per sample can be reduced by a 4 factor. The use of a new set of coefficients with higher correlation produces a significant energy per sample reduction in the architectures. Once again, the architecture with Binary operator presents more energy per sample reduction as can be observed in Table 3. This result shows that although the Hybrid multiplier produces a significant power reduction in the filter architectures, the Hybrid signal propagated in data buses become less efficient. This occurs due to the fact that in general there is less correlation in data buses.

4 Conclusions

In this work different dedicated architectures for FIR filters were implemented. Arithmetic operators that operates with a different code, the Hybrid encoding, were experimented in the FIR filter architectures. Performance comparisons for pipelined architectures using Binary and Hybrid operators were investigated and the results showed that despite higher area shown by the architectures with Hybrid operators, these architectures can present less minimum clock period and energy per sample. As future work we hope to explore different values for m, the size of the groups that work as Gray codes in the Hybrid encoding scheme. We also hope to experiment our Hybrid code in signed multiplication scheme and the impact on power reduction by application of these operators in the FIR architectures.

Acknowledgments

This research was supported in part by the CAPES (Brasilia/Brazil) Institute, Universidade Católica de Pelotas (RS, Brazil) and by the portuguese FCT under program POCTI.

References

[1] Massoud Pedram. Power Minimization in IC Design: Principles and Applications. *ACM Transactions on Design Automation of Electronic Systems*, 1(1):3–56, January 1996.

[2] L. Benini and G. DeMicheli. System-Level Power Optimization: Techniques and Tools. In *Proceedings of the ISLPED*, pages 288–293, 1999.

[3] A. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and W. Brodersen. Optimizing Power Using Transformations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(1):12–31, January 1995.

[4] M. Mehendale, S. D. Sherlekar, and G. Venkatesh. Algorithmic and Architectural Transforms for Low Power Realization of FIR Filters. In *Eleventh Int. Conf. on VLSI Design*, pages 12–17, 1998.

[5] M. Mehendale, S. Sherlekar, and G. Venkatesh. Techniques for Low Power Realization of FIR Filters. *Design Automation Conference*, 3(3):404–416, September 1995.

[6] E. Costa, J. Monteiro, and S. Bampi. Power Optimization Using Coding Methods on Arithmetic Operators. In *IEEE Int. Symp. on Signals Circuits and Systems*, pages 505–508, July 2001.

[7] A. Genderen. SLS: An Efficient Switch-Level Timing Simulator Using Min-Max Voltage Waveforms. In *Proceedings of the International Conference on Very Large Scale Integration*, pages 79–88, 1989.