THE IMPACT OF DATA CHARACTERISTICS ON HARDWARE SELECTION FOR LOW-POWER DSP

G. Keane, J. R. Spanier, R. Woods School of Electrical Engineering and Computer Science The Queen's University of Belfast Ashby Building, Stranmillis Road, Belfast BT9 5AH, Northern Ireland e-mail: g.keane@ee.qub.ac.uk

ABSTRACT

Adders and multipliers are key operations in DSP systems. The power consumption of adders is well understood but there is no analysis based on detailed simulation of multipliers available. This paper considers the power consumption of a number of multiplier structures such as array and Wallace Tree multipliers and examines how the power varies with data wordlengths and different applications (e.g. image and speech). In all cases results were obtained from EPIC PowerMillTM simulations of synthesised circuit layouts, a process which is accepted to be within 5% of the actual silicon. Analysis of the results highlights the effects of routing and interconnect optimization for low power operation and gives clear indications on choice of multiplier structure and design flow for the rapid design of DSP systems. The application of the findings to system level design can result in savings of up to 40

1 INTRODUCTION

The need for lower power consumption has been prompted by cost of current packaging technologies [1] and proliferation of portable computing and communications [2]. Power can be reduced by considering the circuit development at the technology, circuit, architectural and algorithmic levels [3]. Whilst technologies such as Silicon on Insulator (SOI) and circuit level techniques like Complementary Pass-Transistor Logic can result in power savings [1], these options are not available in a semi-custom design flow using VHDL-based logic synthesis.

At the algorithmic or architectural level, the designer can either minimize the switched capacitance or reduce power by dropping the supply voltage. Transformations can be used to increase the throughput rate beyond what has been specified which is then traded off for low power operation by reducing the supply voltage. However, in semi-custom design, the supply voltage is preset by the silicon foundry.

In deep submicron designs, the percentage of both power and delay due to interconnect has been reported to be as high as 90% [4]. Thus, increasing regularity and locality at the silicon level should provide a means to reduce interconnect power. The use of a high-level approach in implementing locality, whereby operations are bound to adjacent hardware units, has been effective in power reduction [5][6], and the work presented here builds on this by forcing arithmetic structures to implement silicon locality at the layout level.

One of the major problems in designing for low power is that accurate power estimations are usually only possible when the circuit layout has been produced. This does not fit well into modern VHDL-based synthesis flow where design area and speed are evaluated early in the design flow. It becomes increasingly important in the Intellectual Property (IP) arena where the focus is to accelerate the design flow. In IP approaches for DSP, designs are synthesised using predefined VHDL cores ranging from multipliers and adders to more complex blocks such as ADPCM blocks [7]. If accurate models for these blocks were developed for the various specified parameters, e.g. wordlength, then it should be possible to perform power estimation at an early stage in the design flow. While analysis exists for the power consumption of adders [8,9,10,11], there are few power dissipation studies for multipliers, particularly from a logic synthesis viewpoint.

In this paper we have carried out extensive simulations of the power consumption of commonly used multipliers. The aim is to identify the optimal choice of multiplier structures for specific low power signal processing applications. Thus we have considered multipliers under different conditions: different wordlengths, locality (array multipliers where locality is preserved in the circuit layout), different data representations (e.g. two's complement) and different DSP data streams e.g. image, speech. In this study, the analysis has been carried out on synthesised circuit layout using the $\operatorname{PowerMill}^{TM}$ simulator. The paper also briefly describes the system that has been developed to work with Compass Design AutomationTM and EPIC tools. This includes the generation of the netlist files and data files necessary to run PowerMill. These results give clear indications on choice of multiplier structure and design flow for a rapid design methodology particularly aimed at DSP applications.

This paper is organized as follows: Section 2 presents the hardware structures being examined in this paper. Section 3 provides information on the design flow used in the research, and Section 4 considers the data streams which are being examined. Section 5 contains the results for the multiplier structures, and looks at some larger system-level issues in light of the results. Section 6 presents our conclusions.

2 MULTIPLIER STRUCTURES

Three basic sign-magnitude multipliers were considered on the basis of their popularity and scope on synthesis. A Booth-encoded multiplier [12] and a Booth-encoded Wallace Tree [13][14] were chosen as they are often used in DSP systems. A Carry-Save array [15] was chosen as it allowed us to investigate the effects of synthesising a regular array multiplier structure. Regular circuit layout was developed to examine the effects of locality. The attraction of maximizing locality is that layout regularity can be preserved in conventional tool design flows. Both a flattened circuit layout and a hierarchical circuit layout for the Carry-Save structure were developed. To investigate the effect of different number representations, two's complement versions of the Carry-Save array and Booth-encoded Wallace Tree architectures were also prepared [http://www.iss-dsp.com]. VHDL models were either prepared by the researchers or, in the case of the Booth-encoded multiplier and Booth-encoded Wallace Tree, commercial models were adapted. In each case, the models were developed in a highly modular fashion allowing the circuits to be parameterised in terms of wordlength.

The multiplier structures presented all have different capabilities. Obviously, the Wallace-Tree structures can operate at much higher frequencies than the array structures. This factor was taken into consideration when comparing the hardware units, by employing parallel structures when necessary. The structures were evaluated operating at a speed of 20MHz, as this allows the designs to be evaluated on the basis of power consumption (the goal is to identify the most power-efficient implementations). Other metrics, such as the Area-Power product, have also been used.

3 DESIGN FLOW

In our work, we have developed a design flow based around commercial tools such as $Synopsys^{TM}$, Compass Design AutomationTM and EPIC Design Technology's PowerMillTM simulator. In order to accelerate the design, simulation and power estimation process, a framework has been developed in Perl which eases the passing of netlists and generation of data files. The framework has a broader range of use than the testing of multipliers used in this experiment, and can be used in amking high-level design decisions. Designs are described in VHDL (both structurally and behaviorally) and synthesized using $Synopsys^{TM}$. Synthesis is targeted towards particular silicon libraries, namely the 0.35μ standard cell CMOS library available in the Compass Design Automation TM toolset. The designs were "placed and routed" using the Compass Design AutomationTM toolset, and a physical netlist developed with characteristics extracted for both the standard cells and the interconnect.

SPICE characterizations of the standard cells (provided by the vendor) were used to prepare the final netlist for the multiplier designs for simulation using the PowerMillTM simulator. PowerMillTM has been reported to produce results within about 5-10% of silicon.

4 DATA CHARACTERISTICS

Whilst multiplier choice may be determined for a single application, it is important to determine the effects of using it in a wide range of applications. In image processing, the data is usually highly correlated, which is not the case in speech applications. In order to investigate the effects of different applications i.e. different data characteristics, on the multiplier selection, three separate data streams were considered. These were a representative speech stream - taken from the TIMIT database (which is widely used for speech recognition and synthesis research), a representative image stream (Lena) and a pseudo-random data stream, generated by the simulation framework.

Accurate representation of the bit stream being processed is essential if optimum hardware structures for the processing task are to be selected. It has been shown [1] that there is a direct relationship between the bit level probabilities and word-level statistics for a number of common data streams. This fact can be used to develop simple piece-wise linear models of the activity in a data stream.

Samples of the speech signal are assumed to be the output of a time-varying discrete-time system. The speech waveform displays a slowly time-varying structure, with significant anti-correlation between consecutive regions. This anticorrelation provides an insight into the best architecture for processing speech. When considering a two's complement implementation, using a preprocessing block that multiplexes data into two different datapaths depending on the sign of the data keeps activity in each of the processors to a minimum. However, for sign-magnitude representations, a single multiplier block is more efficient.

The image stream considered displays characteristics which are different to those of the speech stream. The image is represented as a stream of integers that possess a high degree of correlation between successive values as the image tends to be slow changing from pixel to pixel.

The data representation, namely two's complement, one's complement, sign-magnitude, etc. also has an impact on performance. The degree of correlation in the data stream has an influence (and vice versa) on the multipier chosen.

5 RESULTS

A wide range of experiments was carried out on each of the multipliers. This included synthesizing the multipliers using different wordlengths and simulating using numerous data streams. In each case, a circuit layout was generated which was optimized for the best timing and area within the constraints of the synthesis tools. The physical level and the library of spice-level netlists for the 0.35 μ CMOS standard cells were then used in PowerMillTM to obtain an accurate power estimation using the three different types of data stream.

5.1 Hardware Topology

Tables 1 and 2 provide some detail on the physical implementation of each of the multiplier structures which have been synthesized and brought to silicon layout. Of particular interest is the area comparison, which provides some idea of the cost inherent in implementing a regular silicon structure. The information for all the structures was generated from layout.

The difference in power consumption between a Carry Save structure and a Wallace Tree multiplier can be seen from table 3. For example, a 16-bit Carry Save structure consumed

Name	8-Bit	16-Bit	24-Bit
	$(10^{-6} \mathrm{mm}^2)$	(10^{-6}mm^2)	$(10^{-6} \mathrm{mm^2})$
Carry Save (Regular)	0.20	0.81	1.81
Carry Save (flat)	0.10	0.48	1.15
Two's Complement	0.10	0.48	1.22
Carry Save			
Booth Encoded	0.12	0.48	1.32
Booth Encoded	0.12	0.56	1.42
Wallace Tree			
Two's Comp. Booth	0.10	0.49	1.19
Encoded Wallace Tree			

Table 1: Silicon area for multiplier structures

30% less power than a 16-bit Wallace Tree. This difference was investigated by examining the routing and low level construction details of the synthesized silicon layout. When the routing was examined, it was found that the netlist distribution in the two designs could be characterized as shown in figure 1. The distribution of nets can be approximated using

Name	Number of	Longest	Average Net
	Nets	$\operatorname{Net}(\lambda)$	$\operatorname{Length}(\lambda)$
Carry Save (Regular)	3584	332	130
Carry Save (flat)	1072	8488	472
Two's Complement	1087	6999	446
Carry Save			
Booth Encoded	979	7087	558
Booth Encoded	1053	10657	693
Wallace Tree			
Two's Comp. Booth	950	9786	654
Encoded Wallace Tree			

Table 2: Net information for 16-bit structures

the integral of the curves in figure 1, and this gives some idea of the relative power efficiency of various structures. However, considering just the power consumption does not give a true appreciation of the overall performance. In each of these structures the area varies.

Name	8-Bit	16-Bit	24-Bit
	(mW)	(mW)	(mW)
Carry Save (Regular)	5.99	23.19	56.92
Carry Save (flat)	3.23	25.94	67.96
Two's Complement	3.65	27.31	80.79
Carry Save			
Booth Encoded	5.09	27.95	88.58
Booth Encoded	5.50	37.10	93.60
Wallace Tree			
Two's Comp. Booth	4.00	32.24	86.12
Encoded Wallace Tree			

Table 3: Power consumption of multiplier structures processing random data at 20MHz

Given the contribution of interconnect to the overall power consumption of the circuit, it would be expected that the power performance of the regular array would be much lower than it actually is. The reason for this discrepancy may be the greater toggling which occurs as a direct result of the structure. The critical path of the array leads to unnecessary switching in the structure as data paths are not balanced. The performance of the array should be significantly improved if the input data is skewed and a wavefront array approach taken.

5.2 Data Characteristics

After these initial tests, the structures were tested using representative signal streams. From table 4 an obvious trend



Figure 1: Distribution of net lengths in 16-bit Wallace Tree and Carry-Save structures

in the power consumption of the arithmetic blocks can be seen (in all cases, the multipliers are operating at a speed of 20MHz). The best multiplier for all the data streams is one which exploits the inherently regular structure of the Carry Save array at the silicon level.

Name	Random	Image	Speech
	Power	Power	Power
	(mW)	(mW)	(\mathbf{mW})
Carry Save (Regular)	23.19	7.69	5.66
Carry Save (flat)	25.94	9.58	6.41
Two's Complement	27.31	10.04	9.12
Carry Save			
Booth Encoded	27.95	8.43	8.93
Booth Encoded	37.10	12.84	15.98
Wallace Tree			
Two's Comp. Booth	32.24	9.32	10.45
Encoded Wallace Tree			

Table 4: Power consumed by 16-bit multiplier structures for different data streams (at 20MHz)

As a comparison, the Area-Power product figures for 16 bit structures processing the different data streams are also given in table 5. It can be seen that the Carry Save array which has not been constrained at the silicon level provides the best performance across all the different data streams. The area overhead associated with the regular array structure makes the flat synthesised version more attractive when silicon area is of concern.

The application of these results to typical present-day DSP systems can produce large power savings. For an FIR based application operating at 20MHz, filters as large as 64 or 128 taps can be required. Savings in the region of 40% to 50% of the power could be achieved if a Carry-Save multiplier were used as compared to a Wallace Tree structure. This has

obvious implications for packaging costs and system power budgets.

Name	Random	Image	Speech
	A-P Prod.	A-P Prod.	A-P Prod.
	(mm^2mW)	(mm^2mW)	(mm^2mW)
Carry Save (Regular)	18.76	7.75	5.18
Carry Save (flat)	12.48	3.70	2.72
Two's Complement	13.19	4.85	4.41
Carry Save			
Booth Encoded	13.31	4.01	4.25
Booth Encoded	20.92	7.24	9.01
Wallace Tree			
Two's Comp. Booth	15.96	4.61	5.16
Encoded Wallace Tree			

Table 5: Area-Power product for 16-bit multiplier structures processing different data streams (at 20MHz)

6 CONCLUSION

The power consumption of a number of multiplier structures has been examined. The comparison has explored a large design space, considering different word lengths and data streams. The information gathered can be used in a highlevel design flow to allow the designer to choose the optimum multiplier structure for low power design resulting in savings of over 40%. This emphasises the advantages and implications of using regular array structures. The use of regular structures which minimize the interconnect will have an associated overhead in the form of increased area and reduced speed. This overhead should be considered as one of the design variables, and if a system's operating requirements can be met using regular structures they will provide low power operation. The use of a metric such as an Area-Power product will provide an insight into the relative merit of the various designs in cases where silicon area is considered important.

When the choice of hardware primitives is made with a particular application in mind even greater power savings are possible. In cases where a designer is constrained to certain number formats, the use of regular array architectures can again provide low power operation.

It must be re-emphasized that the performance provided by structures like the Booth Encoded Wallace Tree is far superior to that available from less exotic implementations. These results are intended to provide guidelines for hardware choice in signal processing applications, where the throughput required is known and need not be exceeded. In these cases, power consumption can be limited by choosing appropriate hardware solutions.

7 ACKNOWLEDGEMENTS

The authors would like to acknowledge the technical assistance provided by Integrated Silicon Systems and the Department of Computer Science at The University of Manchester. The financial assistance of the European Social Fund, the EC ESPRIT design experiment projects and the Engineering and Physical Sciences Research Council is also acknowledged.

8 REFERENCES

1. Chandrakasan, A. and Brodersen, R. Low Power Digital Design, Kluwer Academic Publishers, 1996.

2. Brodersen, R., Chandrakasan, A. and Sheng, S. "Low-Power Signal Processing Systems", VLSI Signal Processing V, pp 3-13, 1992.

3. Chandrakasan, A., Sheng, S. and Broderson, R. "Low Power CMOS Digital Design", IEEE Journal of Solid State Circuits, vol. 27, pp 473-484, 1992.

4. Lee, T. and Cong, J. "The new line in IC design", IEEE Spectrum, vol. 34, no. 3, pp 52-58, 1997.

5. Mehra, R., Guerra, L. and Rabaey, J. "Low-Power Architectural Synthesis and the Impact of Exploiting Locality", Journal of VLSI Signal Processing Systems, vol. 13, pp 239-258, 1996.

6. Mehra, R. and Rabaey, J. "Exploiting Regularity for Low-Power Design", Proceedings of the International Conference on Computer-Aided Design, 1996.

7. J.McCanny, D.Ridge, Y.Hu, J.Hunter "Hierarchical VHDL Libraries for DSP ASIC Design" IEEE Proceedings, ICASSP-97, Munich, vol.1, pp.675-679

8. Callaway, T. and Swartzlander, E. "Optimizing Arithmetic Elements for Signal Processing", VLSI Signal Processing V, pp 91-100, 1992.

9. Ko, U., Balsara, P. and Lee, W. "Low-Power Design Techniques for High-Performance CMOS Adders", IEEE Transactions on VLSI Systems, vol. 3, pp 327-333, 1995.

10. Nagendra, C., Owens, R. and Irwin, M. "Power-Delay Characteristics of CMOS Adders", IEEE Transactions on VLSI Systems, vol. 2, pp 377-381, 1994.

11. Nagendra, C., Owens, R. and Irwin, M. "Low Power Tradeoffs in Signal Processing Hardware Primitives", Proceedings of VLSI Signal Processing, pp 276-285, October 1994.

 Booth, A. "A Signed Binary Multiplication Algorithm", Quart. Journal Mech. Appl. Math., vol. 4, pp 236-240, 1951.

13. Wallace, C. "A suggestion for Parallel Multipliers", IEEE Transactions on Electronic Computers, vol EC-13, pp 14-17, 1964.

14. Fadavi-Ardekani, J. "M x N Booth Encoded Multiplier Generator using Optimized Wallace Trees", IEEE Transactions on VLSI Systems, vol. 1, pp 120-125, 1993.

15. Aggrawal, D. "Optimum Array-Like Structures for High-Speed Arithmetic", Proceedings of the 3rd Symposium on Computer Arithmetic, pp 208-219, 1975.