ISSN : 2231-0711
IJCSET | June 2011 | Vol 1, Issue 5,234-237

Power Aware Reconfigurable Multiplier for DSP Applications

S. Sri Sakthi ,N. Kayalvizhi

Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham.

***Abstract*— DSP applications are rich in multiplication operations. Hence there is a growing need in improving the efficiency of multipliers. To improve the performance of multipliers, reconfiguration is introduced. In this paper, reconfiguration is introduced in the form of one level recursive architecture to the existing modified booth multiplier (MBM). It provides reconfigurable modes that satisfy multiple precision requirements. Power consumption of the multipliers can be reduced with the introduction of power efficient schemes namely Dynamic Operand Interchange and Spurious Power Suppression Technique to the reconfigurable booth architecture.**

***Keywords*— Booth multiplier, modified booth multiplier, reconfiguration, one level recursive architecture, power analysis.**

I. INTRODUCTION

Digital signal processing (DSP) requires flexible processing ability, low power consumption and high performance. Multipliers are widely used in DSP and multimedia applications. So modifications are made to the multiplier architecture to achieve all those requirements. The modification suggested is the introduction of reconfigurability to the multipliers. Run time reconfiguration ([1], [2]) utilizes hardware that is adapted at run-time to facilitate greater flexibility without compromising on performance. With dynamic reconfiguration, the same multiplier module is designed to perform the multiplication function for different applications at run time.

There is wide range of multipliers ([3], [4]). Based on the way the data is processed, they are classified as serial, parallel and serial-parallel multipliers as shown in figure 1. Among these multipliers, the basic add-and-shift multiplier [5] performs operations similar to the manual method of doing multiplication. The data are entered serially resulting in slower multiplication.

In parallel multipliers, there are two main classifications. They are array and tree multipliers. C.S. Wallace proposed a tree multiplier architecture [6] which performs high speed multiplication. But this has a high structural irregularity and is unsuitable for VLSI implementation as it demands regularity. Braun multiplier is an unsigned parallel array multiplier which requires n^2 and gates and n-1 adders for the multiplication of n bit operand. They are unsuitable for large size operands.

Baugh-Wooley multiplier ([7], [8]) is also an array multiplier but can perform signed multiplication. But, they also face the same problem as that of braun multiplier. They become inefficient for large size operands.

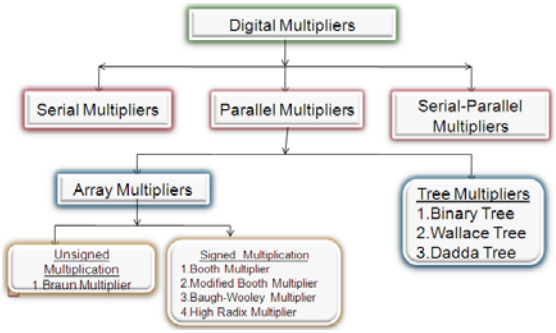

Figure 1.Classification of multipliers based on the way the data is processed.

A. D. Booth [9] introduced Booth multiplier (BM) for signed binary numbers. They are also called radix-2 multiplier. Their main advantage is that it involves no correction cycles for signed terms. But they become inefficient for alternate zeros and ones as it involves large number of adders and subtractors. This results in area and speed limitation. The problem is overcome with modified booth multiplier [10] or radix-4 multiplier which reduces the partial products by 50 percent. Thus it improves speed, reduces power consumption and also saves multiplier layout area. MBM has also a regular structure. So, it is much suited for reconfiguration.

Many types of reconfiguration are suggested to the modified booth multiplier. Modifications are suggested to carry save architecture [11] that compresses the partial products. They are reconfigured to form three level carry select adder. By this modification, they are capable of performing 22 arithmetic operations but the area overhead is very high. Wallace tree structure was suggested for faster compression of the partial products. Leap frog Wallace tree, ling adder [12] were incorporated for high speed multiplication. With 16 bit base multiplier, they are reconfigured to perform 16 bit, 32 bit and modulo multiplications. But the main problem is the irregularity associated with the Wallace tree structure. A scalable Booth architecture [13] is proposed such that smaller multipliers are integrated to perform larger multiplication thereby satisfying multiple precision requirements. The main advantage is that it is regular and scalable. But the main issue is the complexity involved in the design of the scalable architecture. Recursive architecture ([14]-[16]) is also suggested for reconfiguration.

In this paper, a proposal for efficient reconfigurable multiplier architecture is given to satisfy multiple precision requirements. The rest of the paper is organized as follows. The proposed architecture is explained in Section II. In Section III, analysis in terms of area, power and speed between

nonreconfigurable and reconfigurable multiplier is done. In section IV, conclusion and future work is presented.

II. Proposed Work

The architecture proposed for efficient reconfiguration is the One Level Recursive Architecture. Here an n bit multiplier is implemented using four n/2 bit base multipliers. These four n/2 bit multipliers executes in parallel and their results are added up. Thus a larger multiplication is performed using smaller multipliers. The recursive architecture is explained as follows,

Dividing the n-bit to n/2 bit word, they are represented as,

$$A = A_L + A_H \quad (1)$$

$$B = B_L + B_H \quad (2)$$

Therefore the overall multiplication of A with B is,

$$P = A.X. \quad (3)$$

$$P = (A_L + A_H).(B_L + B_H). \quad (4)$$

$$P = A_L B_L + A_L B_H + A_H B_L + A_H B_H. \quad (5)$$

There are four partial products A_LB_L, A_LB_H, A_HB_L, A_HB_H. After these products are formed they are given to the compressor architecture and finally given to adder. The reconfigurable recursive architecture is shown in figure 2.

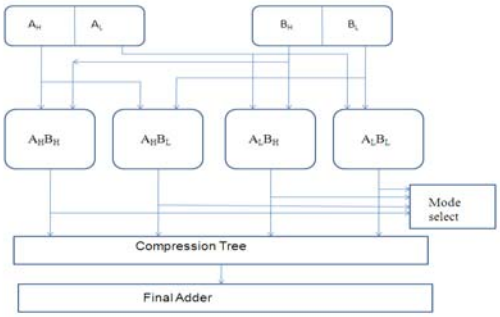

Figure 2. Reconfigurable Recursive Architecture

There are six modes of operation suggested. They are given in table I.

TABLE I
Modes of Configuration

Mode	Function
CM1	Single nxn full precision multiplier
CM2	Single n/2xn/2 full precision multiplier
CM3	Dual n/2xn/2 full precision multiplier
CM4	Single nxn fixed width multiplier
CM5	Single n/2xn/2 fixed width multiplier
CM6	Dual n/2xn/2 fixed width multiplier

Here n is the number of bits. The first three modes work on full precision multiplication. The next three modes work on fixed width multiplication. The first three modes are implemented. The area, speed and power analysis are given in section III.

CM1: All the four base multipliers are active. The intermediate partial products from the four base multipliers are compressed using carry save architecture and the final result is obtained. The schematic for this configuration mode is shown in figure 3.

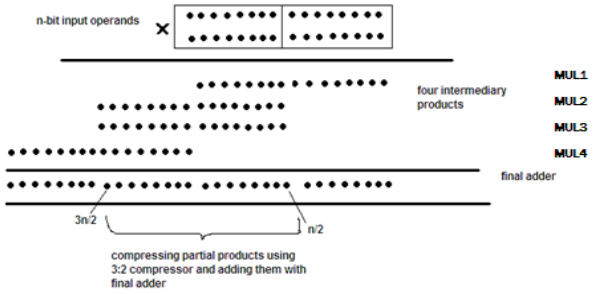

Figure3.Configuration Mode1

CM2: In this mode only one of the base multiplier is active and the other three base multipliers are shut down thereby saving power.

CM3: Two of the base multipliers are active. This improves the throughput of a n/2 bit multiplier without pipelining.

Modes CM4, CM5 and CM6 require the same base multipliers as used in CM1, CM2 and CM3 respectively. But here the width of the product must be equal to the width of the input. Hence the least significant bits (lsb) of the product are truncated.

The base multiplier in the one level recursive architecture can be implemented with any type of multiplier. Here modified booth multiplier is chosen since it halves the number of partial products. This improves the speed and also helps in reducing the power. Carry save architecture is used for the compression of partial products. A fast carry propagate adder is used to get the final result. To this architecture, power reduction techniques ([17], [18]) can be applied to improve the reliability and efficiency of the architecture.

Dynamic Operand Interchange Technique: Dynamic operand interchange technique interchanges operands dynamically during the execution. The operand with the smallest dynamic range is chosen as the multiplier operand. This requires a Dynamic Range Detector (DRD) circuit to detect dynamic range of the operand and a switcher to exchange the operand dynamically. When the operand with smaller dynamic range is encoded with modified booth, there is increased probability of partial products becoming zero. This reduces the switching activities and thereby reduces power consumption. Around 20% power savings is expected when compared to the original modified booth multiplier.

Spurious Power Suppression Technique: Spurious Power Suppression Technique is used in the adder architecture. There is a detection logic which determines whether the most significant part is needed or not for the operation and

accordingly turns on or off that part during the execution. The overall architecture of the proposed multiplier is shown in the figure 4.

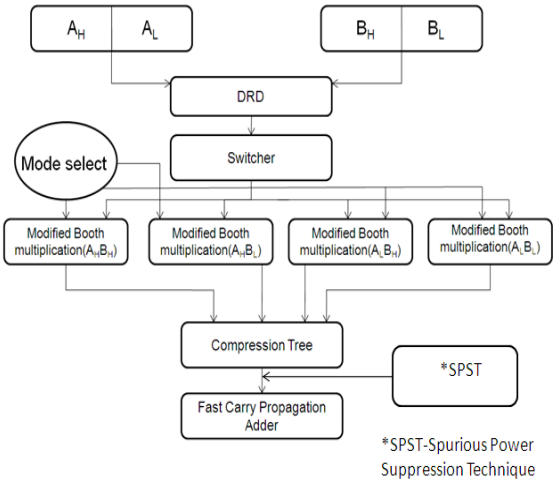

Figure 4. Overall Architecture of the Proposed Reconfigurable Multiplier

III. Results and Discussion

In this paper, scalable nonreconfigurable nxn (n is the number of bits) modified booth multiplier and reconfigurable nxn modified booth multiplier (only three modes are implemented, i.e CM1,CM2 and CM3) is implemented using Verilog HDL and functional simulation is done in MODELSIM. Area, power and timing analysis is done using Altera Quartus II tool (version 9.0) Area, Speed and Power Comparisons between nonreconfigurable and reconfigurable multipliers are shown in table II, table III and table IV respectively.

TABLE III
AREA COMPARISON BETWEEN NONRECONFIGURABLE AND RECONFIGURABLE MULTIPLIER

n (number of bits)	NonReconfigurable Modified Booth Multiplier(LE)	Reconfigurable Modified Booth Multiplier(LE)
16	404 (100%)	354 (87%)
32	737 (100%)	671 (84%)
64	1381 (100%)	1117(80%)

TABLE IIIII

SPEED COMPARISON BETWEEN NONRECONFIGURABLE AND RECONFIGURABLE MULTIPLIER

n (number of bits)	NonReconfigurable Modified Booth Multiplier(MHz)	Reconfigurable Modified Booth Multiplier(MHz)
16	150.06 (100%)	201.05 (133%)
32	116.29 (100%)	158.23 (136%)
64	93.52 (100%)	129.57 (138%)

TABLE IVV

POWER COMPARISON BETWEEN NONRECONFIGURABLE AND RECONFIGURABLE MULTIPLIER

n (number of bits)	NonReconfigurable Modified Booth Multiplier(mW)	Reconfigurable Modified Booth Multiplier(mW)
16	200.92 (100%)	216.15 (107%)
32	317.45 (100%)	315.67 (99.43%)
64	436.95 (100%)	366.25 (83.81%)

Table II shows that around 20% savings in area is obtained for 64-bit Reconfigurable multiplier when compared to 64-bit Nonreconfigurable multiplier and from the analysis much savings in area is achieved for a larger width multiplier. Table III shows that around 38% improvement in speed is achieved for 64-bit Reconfigurable multiplier when compared to 64-bit Nonreconfigurable multiplier. As the width of the multiplier increases, speed also increases. Table IV shows a power overhead of 7% for 8-bit Reconfigurable multiplier when compared to 64-bit Nonreconfigurable multiplier. But as bit size increases, there is savings in power and around 16% power is saved for 64-bit Reconfigurable multiplier when compared to 64-bit Nonreconfigurable multiplier.

IV. Conclusion

An efficient reconfigurable multiplier for full precision applications has been implemented. The analysis between non-reconfigurable and reconfigurable multiplier shows that there is improvement in area, speed and power while using the reconfigurable multiplier. Implementation of reconfigurable modes for fixed width multiplication is the current work. Introduction of power efficient schemes to the reconfigurable multiplier is the future work.

References

[1] Bondalapati* and Viktor K. Prasanna, "Reconfigurable computing: Architectures, models and algorithms", *current science,* Vol. 78, no. 7, 10 , pp 828-837,April 2000.

[2] [white paper] ALTERA: FPGA run-time reconfiguration two approaches. http\\ www.altera.com.

[3] Kiat-Seng Yeo and Kaushik Roy,"Low-Voltage,Low-Power VLSI Subsystems" *Tata McGraw-Hill*,ISBN-13-978-0-07-067750-0, pp 119-146, 2009.

[4] Neil.H.Weste and Kamran Eshraghian," Principles of CMOS VLSI Design," *Addison-Wesley,* ISBN 0201733897,pp 361-380,1993.

[5] http:// Structure of Computer Systems[online]

[6] Wallace, C S. "A suggestion for a fast mulTiplier," *IEEE Transactions on Electronic Computers* ,vol EC-13 ,pp 14-17,Feb 1964.

[7] C.R Baugh and B.A Wooley," A Two's Complement Parallel Array Multiplication Algorithm," *IEEE Transactions on Computers*, Vol. 22, No 12, pp.1045-1047, December 1973.

[8] Aswathy Sudhakar , D. Gokila , "Proposal for an Efficient Reconfigurable Fixed-Width Multiplier", *ICNVS'10 Proceedings of the 12th international conference on Networking, VLSI and signal processing,* ISBN: 978-960-474-162-5,2010.

[9] A .D. Booth, "A signed Binary Multiplication Technique," *Quart. J. Mech. Appl. Math.v,4 part2,* pp 236-240, 1951.

[10] P.E.Madrid, B.Miller and E.E.Swartzlander," Modified Booth Algorithm for High Radix Fixed -Point Multiplication," *IEEE Transactions on Very Large Scale Integeration (VLSI)Systems*, Vol . 1, No. 2,pp 118-121 June 1993.

[11] Jer Min Jou, Yun-Lung Lee, Chen-Yen Lin, and Chien-Ming Sun" A Novel Reconfigurable Computation Unit for DSP Applications" in *IEEE Computer Society Annual Symposium,*, pp 183-188, March 2007.

[12] Wei Li ,Zi-Bin Dai ,Tao Meng Qiao Ren ," Design and Implementation of A High-speed Reconfigurable Multiplier" in *IEEE International conf.ASIC*,pp177-180, Oct.2007.

[13] Z. Shun, O. A. Pfander, H.-J. Pfleiderer, and A. Bermak, "A VLSI architecture for a run-time multi-precision reconfigurable Booth multiplier," in *Proc. 14th IEEE Int. Conf. Electron., Circuits, Syst.*, pp. 975–978, Dec. 2007.

[14] Shiann-Rong kuang , Jiun-Ping Wang ," Design of Power-Efficient Configurable Booth Multiplier," *IEEE transactions on circuits and systems regular papers*, vol. 57, no. 3, pp 568-580.march 2010.

[15] Danysh, A.N., Swartzlander, E.E., Jr., "A recursive fast multiplier", *Asilomar Conf. on Signals, Systems & Computers*, vol. 1, pp. 197 -201, Nov 1998.

[16] P. Mokrian, M. Ahmadi, G. Jullien, and W. C. Miller, "A reconfigurable digit multiplier architecture," in *Proc. IEEE Conf. Elect.Comput. Eng.*, pp. 125–128, May 2003.

[17] Park, S. Kim, and Y.-S. Lee, "A low-power Booth multiplier using novel data partition method," in *Proc. IEEE Asia-Pacific Conf. Adv. Syst. Integrated Circuits*, pp. 54–57, Aug. 2004.

[18] C. N. Marimuthu, P. Thangaraj , "Low Power High Performance Multiplier ", *ICGST-PDCS, Volume 8, Issue 1*, December 2008.