Volume 4, Issue 10, October 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Implementation of Modified Booth Recoded Wallace Tree Multiplier for fast Arithmetic Circuits

**P. S. Tulasiram*, D. Vaithiyanathan, R. Seshasayanan**
Department of Electronics and Communication Engineering, College of Engineering Guindy,
Anna University, Chennai, Tamil Nadu, India

***Abstract—Power consumption has become a critical concern in today's VLSI system design. The growing market for fast floating-point co-processors, digital signal processing chips, and graphics processor has created a demand for high speed, area-efficient multipliers. The Modified Booth Recoding method is widely used to generate the partial products for implementation of large parallel multipliers, which adopts the parallel encoding scheme. A Wallace tree multiplier is an improved version of tree based multiplier architecture and uses carry saveaddition algorithm to reduce the latency. This paper presents efficient design of multiplier that combines the features of Wallace tree and Modified Booth algorithm, and aimed at reduction of area and improvement in speed.An efficient VerilogHDL code has been written, successfully simulated and synthesized for Xilinx FPGA vertex-6 low power (Xc6vlx75tl-1Lff484) device, using Xilinx 12.2 ISE and XST. The analyses obtained from implementation show that architecture is 41% faster than the Wallace tree architecture with optimal area utilization.***

***Keywords—Multiplier, Adder, Modified booth recoding, Wallace tree, Xilinx.***

## I. INTRODUCTION

Multipliers are basic elements in several digital signal processing applications,such as filtering,convolution,fast fourier transform, discrete cosine transform,compression algorithms,hardware implementation of mathematical functions and so on. Thus multiplier optimization in terms of area, power and speed is a major concern for digital designers.Recent research activities in the field of arithmetic optimization have shown that the design of arithmetic optimization have shown [1-2] that the design of arithmetic components combining operations which share data, can lead to significant performance improvements. Based on the observation that an addition can often be subsequent to a multiplication, the Multiply-Accumulator (MAC) and Multiply-Add (MAD) units were introduced [3] leading to more efficient implementations of DSP algorithms compared to the conventional ones, which use only primitive resources [4].Several architectures have been proposed to optimize the performance of the MAC operation in terms of area occupation,critical path delay or power consumption [5-6]. As notedin[7], MAC components increase the flexibility of DSP data path synthesis as a large set of arithmetic operations can be efficiently mapped onto them. Except the MAC/MAD operations, many DSP applications are based on Add-Multiply (AM) operations [8]. The straightforward design of the AM unit, by first allocating an adder and then driving its output to the input of a multiplier, increases significantly both area and critical path delay of the circuit. For Booth arrays,typically radix-4,their truncation history followed a similar path to that of the AND arrays, first followingtruncation [18] and column promotion [19].Exhaustive simulation of the truncated part of the Booth array was used to design compensation circuitry based upon the conditional exception of the error [20], or in order to construct Karnaugh maps of the ideal correction.Truncated arrays also have been considered for squarers, radix-4 and 16 and Booth squarers, radix-4 and 16 and Booth squarer arrays [21-23].The structural optimization is performed on the conventional Wallace tree multiplier, in such a way that the latency of the total circuit reduces considerably. The conventional Wallace tree multiplier architecture[24-25] comprise of an AND array for computing the partial products, a carry save adder for adding the partial products so obtained and a carry propagate adder in the final stage. Recently, the technique of [10] has been used for the design of high performance flexible coprocessor architectures targeting the computationally intensive DSP application [12]. Zimmermann and Tran [11] present an optimized design of [9] which results in improvements in both area and critical path.

This paper is organized as follows architecture is discussed in Section II. In Section III FAM Implementation is discussed. In Section IV, the performances of the proposed multipliers are evaluated. Conclusions are drawn in Section V.

## II. ARCHITECTURE

The multiplier is composed of three blocks: the Modified Booth Encoder and multiplicand selector block for formation of partial products and Wallace tree section, which adds all of the partial products simultaneously to eventually produce two numbers; the third block is the adder section which adds the two numbers obtained from the Wallace tree section as fast as possible.

 

### *A. Modified Booth*

Modified Booth is a prevalent form used in multiplication [13-15]. It is a redundant signed-digit radix-4 encoding technique. Its main advantage is that it reduces by half the number of partial products in multiplication comparing to any other radix-2 representation.

Let us consider the multiplication of 2's complement numbers X and Y with each number consisting of n=2k bits. The multiplicand Y can be represented in Modified Booth form as:

$$Y = (y_{n-1}y_{n-2}\dots y_1y_0) = -y_{2k-1}.2^{2k-1} + \sum_{i=0}^{2k-2} y_i.2^i \quad (1)$$

$$= (y_{k-1}^{MB}y_{k-2}^{MB}\dots y_1^{MB}y_0^{MB}) = \sum_{j=0}^{k-1} y_j^{MB}.2^{2j} \quad (2)$$

$$y_j^{MB} = -2y_{2j+1}+y_{2j}+y_{2j-1}. \quad (3)$$

Digits $y_j^{MB} \in \{-2,-1,0,+1,+2\}, 0 \le j \le k-1$correspond to the three consecutive bits $y_{2j+1}, y_{2j}, y_{2j-1}$ with one bit overlapped and considering that$y_{-1} = 0$. Table I shows how they are formed by summarizing the Modified Booth encoding technique.

Table I modified booth encoding table

| **Binary** | | | $y_j^{MB}$ |
|---|---|---|---|
| $Y_{2j+1}$ | $y_{2j}$ | $Y_{2j-1}$ | |
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **+1** |
| 0 | 1 | 0 | **+1** |
| 0 | 1 | 1 | **+2** |
| 1 | 0 | 0 | **-2** |
| 1 | 0 | 1 | **-1** |
| 1 | 1 | 0 | **-1** |
| 1 | 1 | 1 | **0** |

### *B. Wallace Tree*

The Wallace tree has three steps: Multiply each bit of one of the arguments by each bit of the arguments, by each bit of the other, yielding$n^2$results.Depending on position of the multiplied bits, the wires carry different weights for example wire of bit carrying result of $a_2b_3$ is 32.Reduce the number of partial products to two by layers of full and half adders group the wires in two numbers and add them with a conventional adder.The second phase works as follows. As long as there are three or more wires with the same weight and a following layer: Take any three wires with the same weights and input them into a full adder. This result will be an output wire of the same weight and an output wire with a higher weight for each three input wires. If there are two wires of the same weight left, input them into a half adder. If there is just one wire left connect it to the next layer. The advantages of using the Wallace tree architecture is, all the bits of all of partial products in each column are added together in parallel independent of other columns.In the conventional 8 bit Wallace tree multiplier design more number of addition operations is required. Using adder, three partial product terms can be added at a time to form the carry and sum. The sum signal is used by the full adder of next level. The carry signal is used by the adder involved in the generation of the next output bit, with a resulting delay proportional to $\log 3n/2$, for n number of rows [17]. A multiplier consists of various stages of fulladders, each higher stage adds up to the total delay of the system. In the first and second stages of the Wallace structure, the partial products do not depend upon any other values other than the inputs obtained from the AND array. However for the immediate higher stages, the final value (PP3) depends on the Cout value of previous stage. This operation is repeated for further stages. Hence, the major cause of delay is the propagation of the carry out from the total number of stages in the critical path.

## III. FAM IMPLEMENTATION

In this paper, we focus on AM units which implement the operation$Z = X * (A + B)$.The design of the AM operator requires that its inputs A and B are first driven to n adder and then the input X and the sum Y=A+B are driven to a multiplier in order to get output Z. For addition different adders are used if we use the carry look ahead adder it will reduced the delay but there will be increase in area occupation and power dissipation.In the FAM design, the multiplier is a parallel one based on the MB algorithm. Let us consider the product X*Y.The term Y= $(y_{n-1}y_{n-2}\dots y_1y_0)$ is encoded based on the MB algorithm and multiplied with X=$(x_{n-1}x_{n-2}\dots x_1x_0)$.Both X and Y consist of n=2k bits and are in 2's complement form. Equation (4) describes the generation of the K partial products

$$PP_j = X * y_j^{MB} = \overline{p_{j,n}}2^n + \sum_{i=0}^{n-1} p_{j,i} * 2^i \quad (4)$$

 

For the computation of the least and the most significant bits of the partial product we consider $x_{-1} = 0$ $and$ $x_n = x_{n-1}$ respectively. Note that in case that n=2k+1, the number of the resulting partial products is$\lfloor n/2 \rfloor + 1 = k + 1$ and the most significant MB digit is formed based on sign extension of the 2'scomplement number.

After the partial products are generated, they are added,properly weighted, through a Wallace carry-save adder(CSA) tree shown in Fig 1.

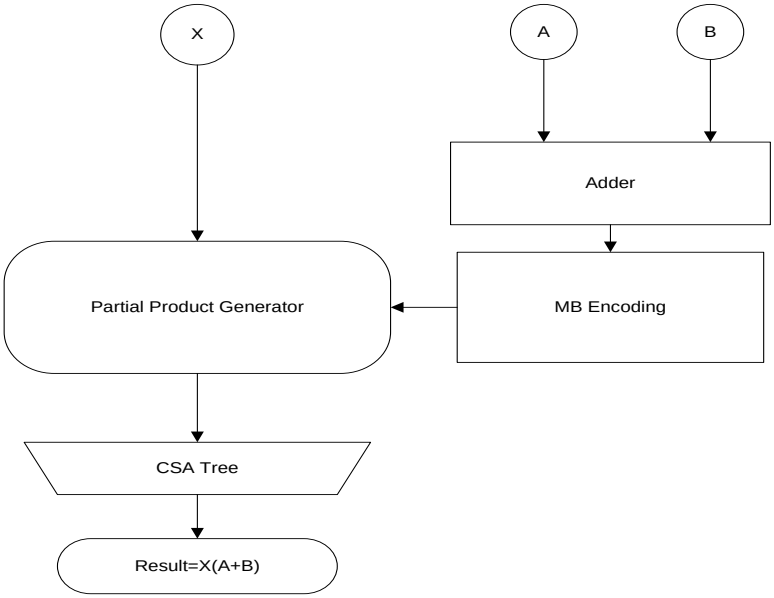


Fig. 1 Architecture of Add Multiply unit

## IV. RESULT

In Table shows the detailed report of the analysis done with benchmark circuits implement in Xilinx 6VLX240TFF784-3 and simulated and synthesized using Xilinx 12.2. The results were taken for both existing Wallace tree multiplier and proposed Wallace tree multiplier designed for 8 bit word.The proposed multiplier is optimized according to speed, generating partial products and sums. Upon comparison of the speed, we can conclude that Modified Booth recoded and Wallace tree multiplier proves to be a better option over conventional multipliers used in several complex VLSI circuits since its speed is 1.41 times faster than Wallace tree multiplier [17] and 1.15 times faster than booth recoded Wallace tree multiplier [8].The proposed architecture performs 13% faster when compared with architecture [18], and 41% faster compared with architecture [17]. The area remains same in proposed and reported architecture [18]. This variation is due to reduction in number of partial product i.e. by the factor of$\left\lfloor \frac{n}{2} + 1 \right\rfloor$in modified Booth recoded Wallace tree multiplier.

Table II comparison of different wallace tree multiplier

| S.No | Title | Delay(ns) | Slice Utilized | Cell Usage |
|---|---|---|---|---|
| 1. | Wallace tree multiplier [17] | 9.099 | 143 | 281 |
| 2. | Wallace tree multiplier(Carry look ahead adder )[8] | 8.933 | 179 | 162 |
| 3. | Booth recoded Wallace tree Multiplier[18] | 7.428 | 90 | 90 |
| 4. | Modified Booth recoded Wallace tree Multiplier | 6.442 | 92 | 90 |

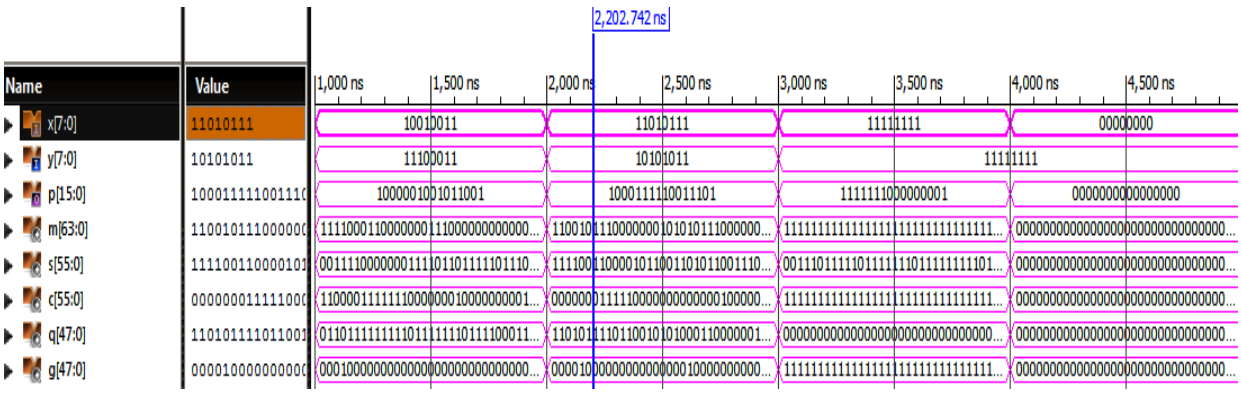


Fig.1 Output Waveform for Wallace Tree Multiplier.

 

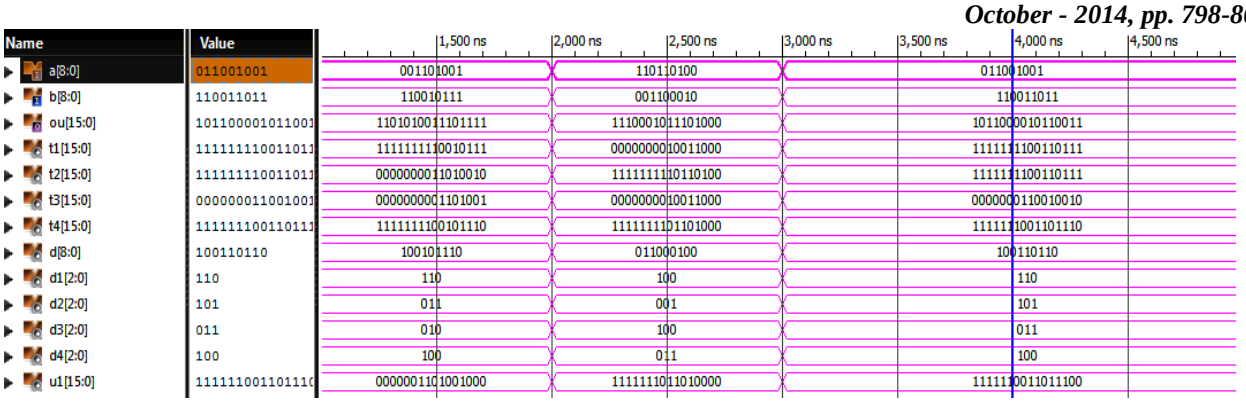


Fig. 3 Output Waveform for Modified Booth Recoded Wallace Tree Multiplier.

## V. CONCLUSION

In this paper, we proposednovel high speed architecture for multiplication of two 8 bit numbers, combining the advantages of Modified Booth algorithm and Wallace tree multiplier. Further the simulation results of various types of Wallace tree were compared and the proposed multiplier is optimized according to speed, generating partial products and sums.And speed is 1.41 times faster than Wallace tree multiplier [17] and 1.15 times faster than booth recoded Wallace tree multiplier [8]. Our multiplier can be used in high speed applications such as adaptive filter, phase locked loop and neural networks. As a future work, the multiplier's performance could be tested with Filters, ALU and several other multipliers.

## REFERENCES

[1] A. Amaricai, M. Vladutiu, and O. Boncalo, "Design issues and implementations for floating-point divide-add fused," *IEEE Trans. CircuitsSyst. II–Exp. Briefs*, vol. 57, no. 4, pp. 295–299, Apr. 2010. [2] E. E. Swartzlander and H. H. M. Saleh, "FFT implementation withfused floating-point operations," *IEEE Trans. Comput.*, vol. 61, no. 2, pp. 284–288, Feb. 2012.

[3] J. J. F. Cavanagh, *Digital Computer Arithmetic*. New York: McGrawHill, 1984. [4] S. Nikolaidis, E. Karaolis, and E. D. Kyriakis-Bitzaros, "Estimation of signal transition activity in FIR filters implemented by a MAC architecture," *IEEE Trans. Comput.-Aided Des.Integr. Circuits Syst.*, vol.19, no. 1, pp. 164–169, Jan. 2000.

[5] O. Kwon, K. Nowka, and E. E. Swartzlander, "A 16-bit by 16-bit MAC design using fast 5: 3 compressor cells," *J. VLSI Signal Process. Syst.*,vol. 31, no. 2, pp. 77–89, Jun. 2002.

[6] Y.-H. Seo and D.-W. Kim, "A new VLSI architecture of parallelmultiplier–accumulator based on Radix-2 modified Booth algorithm,"*IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no.2,pp.201–208,Feb.2010.

[7] A. Peymandoust and G. de Micheli, "Using symbolic algebra in algorithmic level DSP synthesis," in *Proc. Design Automation Conf.*, LasVegas, NV,pp. 277–282.2001.

[8] W.-C. Yeh and C.-W. Jen, "High-speed and low-power split-radixFFT," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 864–874, Mar.2003.

[9] C. N. Lyu and D. W. Matula, "Redundant binary Booth recoding," in*Proc. 12th Symp. Comput. Arithmetic*,pp.50–57,1995.

[10] W.-C. Yeh, "Arithmetic Module Design and its Application to FFT,"Ph.D. dissertation, Dept. Electron.Eng., National Chiao-Tung University,Chiao-Tung, 2001.

[11] R. Zimmermann and D. Q. Tran, "Optimized synthesis of sum-of-products," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove,Washington, DC, pp. 867–872, 2003.\

[12] S. Xydis, I. Triantafyllou, G. Economakos, and K. Pekmestzi, "Flexible datapath synthesis through arithmetically optimized operation chaining," in *Proc. NASA/ESA Conf. Adaptive Hardware System*, pp. 407–414., 2009.

[13] Z. Huang, "High-Level Optimization Techniques for Low-Power Multiplier Design," Ph.D., University of California, Department of Computer Science, Los Angeles, CA, 2003.

[14] Z. Huang and M. D. Ercegovac, "High-performance low-power left-toright array multiplier design," *IEEE Trans. Comput.*, vol. 54, no. 3, pp.272–283, Mar. 2005.

[15] O. L. Macsorley, "High-speed arithmetic in binary computers," *Proc.IRE*, vol. 49, no. 1, pp. 67–91, Jan. 1961.

[16] N. H. E. Weste and D. M. Harris, "Datapath subsystems," in *CMOSVLSI Design: A Circuits and Systems Perspective*, 4th ed. Readington: Addison-Wesley, ch. 11.2010.

[17] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron. Compute.*,vol. EC-13, no. 1, pp. 14–17, 2004.

[18] A.A. Katkar and J.E. Stine, "Modified Booth Truncated Multipliers,"Proc. 14th ACM Great Lakes Symp. VLSI (GLSVLSI'04). pp. 444-447, 2004, http://doi.acm.org/10.1145/988952.989059, accessed on Mar. 11, 2013.

[19] H.-A. Huang, Y.-C.Liao, and H.-C. Chang, "A Self-Compensation Fixed-Width Booth Multiplier and Its 128-Point FFT Applications,"IEEE Int'l Symp. Circuits and Systems (ISCAS'06),pp. 4-3541, May 2006.

 

[20] Y.-H. Chen, T.-Y.Chang, and R.-Y. Jou,"A Statistical Error-Compensated Booth Multipliers and Its DCT Applications,"IEEE Region10Conf., pp. 1146-49,Nov.2010.

[21] V. Garofalo, M. Coppola, D. De Caro, E. Napoli, N. Petra, and A.Strollo,"A Novel Truncated Squarer with Linear Compensation Function," IEEE Int'l Symp. Circuits and Systems (ISCAS),pp. 4157-4160, May 30-June 2, 2010.

[22] S. Datla, M. Thornton, and D. Matula,"A Low Power High Performance Radix-4 Approximate Squaring Circuit,"Proc. 20th IEEE Int'lConf. Application-Specific Systems, Architectures and Processors(ASAP'09), pp. 91-97, July 2009.

[23] K.-J. Cho, W.-K.Kim, B.-K.Kim, and J.-G. Chung,"Design of Low Error Fixed-Width Squarer,"Proc. IEEE Workshop Signal ProcessingSystems (SIPS'03), pp. 213-218, Aug. 2003.

[24] List I. Abdellatif, E. Mohamed, "Low-Power Digital VLSI Design, Circuits and Systems," Kluwer Academic Publishers, 1995.

[25] H. Neil. Weste and Kamran Eshraghian, "Principles of CMOS VLSI design-A Systems Perspective," Pearson Edition Pvt Ltd. 3rd edition, 2005.

 