Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 11, Number 3 (2015), pp. 1647-1653

http://www.ripublication.com

Design of Low Power Reduced Delay Fixed-Width Modified Booth Multiplier

Balamurugan S, Ravi Kumar, Marimuthu R
School of Electrical Engineering, VIT University, Vellore, India
sbalamurugan@vit.ac.in

Abstract

This paper presents a design of low power reduced delay fixed-width booth multiplier for multimedia and communications systems. This multiplier architecture is based on radix-4 booth multiplier. We introduce decomposition logic in the proposed fixed-width booth multiplier architecture to improve the delay and power consumption. The design is developed by using Verilog-HDL and synthesized using a Cadence tool for Area, Power and Delay estimation of proposed architecture as well as existing architecture. This result shows that the proposed architecture consumes less power consumption and delay as compared to the existing architectures.

Keywords: fixed-width; booth multiplier; decomposition logic; low power; high speed

Introduction

The rapid advances in multimedia and communication systems, real-time signal processing like audio signal processing, video/image processing, or large-capacity data processing are increasingly being demanded. The multiplier is an essential element of the digital signal processing such as filtering, convolution, and Inner products.

Designing multipliers with high speed and low power consumption is one of the major challenges in any battery operated multimedia and communications systems. In many of these applications, it is desirable to maintain the same bit width in both input side as well as the output side with little accuracy loss of output data. The modified -Booth algorithm is widely used to compute the signed operand multiplier [1] due to its inherent advantage of almost half of the partial products storage can be saved and critical path delay also only half as compared to the Baugh-Wooley array multiplier [2]. The applications which required n-bits output, the full width multiplier is not the best choice because it takes two n-bit operands and produces 2n-bit product outputs which consumes more power and area. To avoid the output register widening, the result has to be of the same bit width as the inputs. A very simple way to do so is by

rounding the output to n-bits. This gives the most accurate result and is called the Post-Truncation (PT) multiplier. To achieve PT multiplication, add an extra '1' at the $(n-1)^{th}$ bit position of the true product. The PT multiplier, however, has more hardware complexity. The fixed-word size can be achieved by direct-truncation (DT) method.

In DT method least-significant n-bits (LSBs) are directly truncated, which leads low power and area consumption but produces more truncations error. In order to reduce this error appropriate bias values are introduced into the retained adder cells. Various fixed-width multiplier design schemes have been proposed in the literatures [3-10] using either constant correction or variable correction technique. In Constant correction technique (CCT) [3], the bias value is computed using simple bias generation circuit and added to the retained adder cells in order to reduce the truncation error. Using CCT the bias values are generated independent of the input signals. Variable correction technique (VCT) further reduces the total truncation error by considering the LSBs partial products columns [4-10].

Fundamental of Modified Booth Multiplier

Let us consider the multiplication operation of two n-bit signed numbers $X = x_{n-1}x_{n-2}...x_0$ (multiplicand) and $Y = y_{n-1}y_{n-2}....y_0$ (multiplier). The two's complement representation of X and Y can be expressed as follows:

$$X = -x_{n-1}2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i, \quad Y = -y_{n-1}2^{n-1} + \sum_{i=0}^{n-2} y_i 2^i \tag{1}$$

$$P = X.Y = \sum_{i=0}^{2n-1} P_i 2^i \tag{2}$$

P_i denotes the i^{th} output product bit. Assume n is even, we can rewrite Y as:

$$Y = \sum_{i=0}^{(n-2)/2} z_k 2^{2i} = \sum_{i=0}^{(n-2)/2} (y_{2i-1} + y_{2i} - 2y_{2i+1})2^{2i} \tag{3}$$

Where $y_{-1} = 0$ and $z_k \in \{-2,-1,0,1,2\}$.

$$P = XY = \sum_{i=0}^{(n-2)/2} (y_{2i-1} + y_{2i} - 2y_{2i+1})2^{2i} \cdot X \tag{4}$$

$$= \sum_{i=0}^{(n-2)/2} S_i \tag{5}$$

Where $S_i = (y_{2i-1} + y_{2i} - 2y_{2i+1})2^{2i} \cdot X$ and it is known that the scanning of triplets begin y_{-1} to the MSB with one-bit overlapping. Based on the encoded result shown in Table I, the Booth encoder and partial product generation proposed in [5] is adopted and shown fig. 1 and fig. 2 respectively.

Table 1: Modified Booth Encoding Table

y_{2i+1}	y_{2i}	y_{2i-1}	Operation	Crtl [2]	Crtl [1]	Crtl [0]
0	0	0	0	0	0	0
0	0	1	1	0	1	0
0	1	0	1	0	1	0
0	1	1	2	0	0	1
1	0	0	-2	1	0	1
1	0	1	-1	1	1	0
1	1	0	-1	1	1	0
1	1	1	0	1	0	0

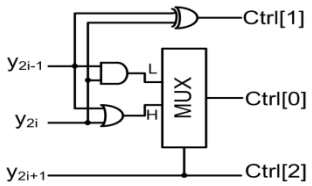

Figure 1: Modified Booth Encoder

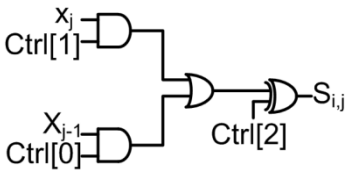

Figure 2: Partial Product generator

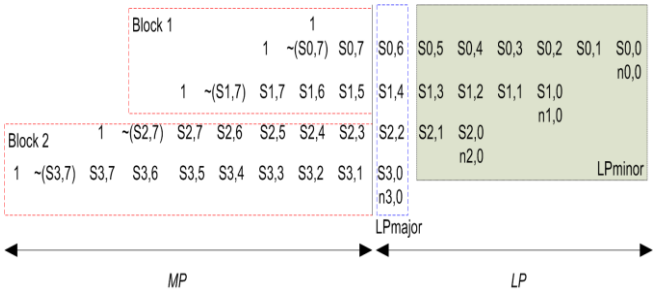

Figure 3: Partial products for n=8

Proposed Modified Fixed Width Multiplier Architecture

In the proposed architecture, the most significant columns of partial products are decomposed into two parts namely block 1 and block 2. It gives better results in terms

of low power consumption and less delay as compared to existing architecture available in the literature. It can be clearly seen that the two blocks are computed the partial sum bits in parallel and propagate it for final addition.

In our design the longest delay path produced by Song's Fixed-Width Multiplier architecture is reduced by decomposing the most significant column of 'n' bit as well as LPmajor column from LP part. The partitioning of blocks is shown in Figure 3. In Fig 5 and 6 we have shown the architecture and the longest delay path. As the structure here is divided into two blocks it has a shorter delay path than [4]. In Song *et al.* the last row adder which had to wait for the previous row carries caused the delay to increase as can be seen from the figure. This is because all the rows where adder in one block and thus the final row had to depend on previous block which in turn had to depend to its previous. Thus the final row had to depend on n/2 rows where 'n' is the size of the multiplier.

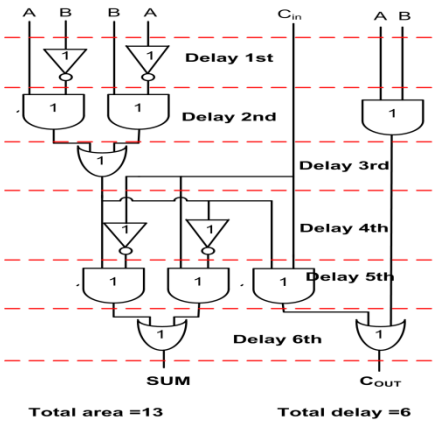

Figure 4: Delay and Area evaluation of a Full adder

The Full adder as shown in Figure 4 is implemented using AND, OR and INVERTER (AOI) for delay and Area evaluation. The number 1 on each gate indicates that it contributes 1 unit of delay and occupies 1unit of the area. The gates between the dotted lines perform their operation in parallel and thus each parallel execution contributes one delay. By adding up the number of stages we find the longest delay contributed by that logic block. Here the longest delay path contributes 6 units of delay as is clearly depicted. The total number of basic gates (AOI) adds up to determine the total area. Table II provides the delay and area count of proposed multiplier architecture (PMA) basic building blocks

In fig. 5, the critical path is shown in red dotted lines. As can be seen the path covers 11 full adders and 1 half adder. It contributes a delay of 69 gate times. In Proposed fixed-width multiplier design, shown in fig. 6, we can see that the critical path is as below: Each full adder will give a delay of 6 gate time and a half adder of 3 gate time. Only taking the adders i.e. 7 full adders and 4 half adders in the critical path, we have 54 gate delay.

It is clearly shown in the Table III that the delay of the proposed architecture is improved by more than 10% as compared to [4].

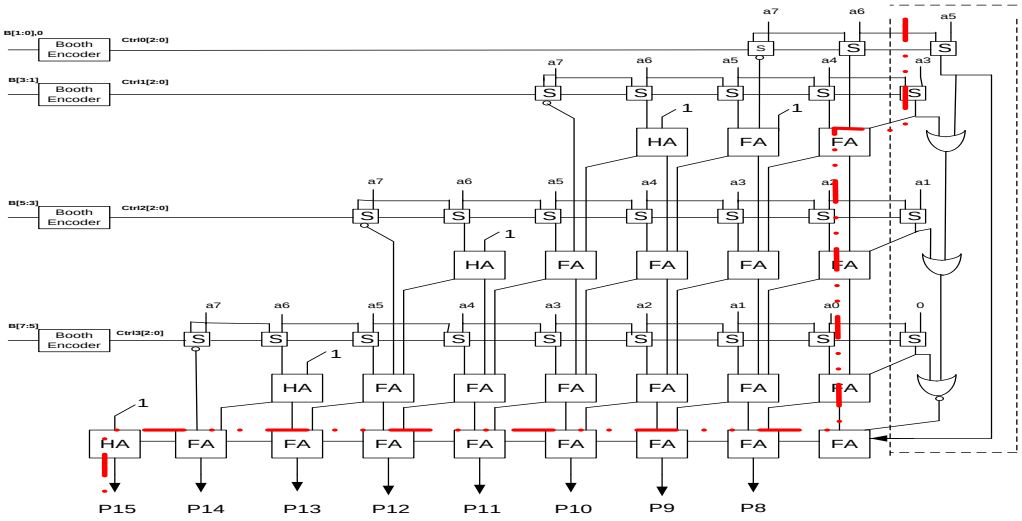

Figure 5: Song [4] Architecture and longest delay path

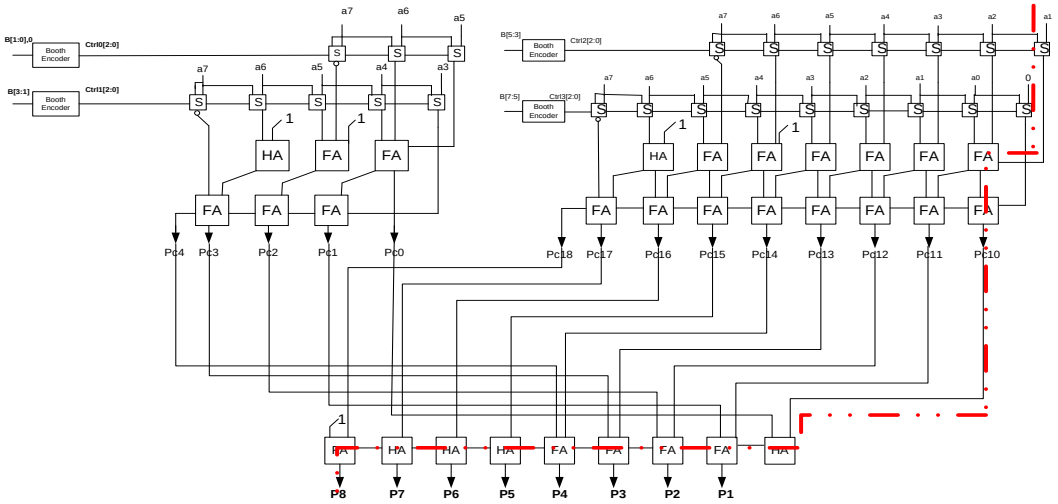

Figure 6: PMA and longest delay path

Table 2: Delay and Area Count of Basic PMA Building Blocks

	Delay	**Area**
XOR	3	5
2:1 MUX	3	4
Half Adder	3	6
Full Adder	6	13

Table 3: Delay Evaluation of longest path

	Song [4]		Proposed		Ratio of Proposed/[4]
	#FA+#HA	Delay	#FA+#HA	Delay	
8x8	11FA+1HA	69	7FA+4HA	54	1.278
12x12	17FA+1HA	105	12FA+4HA	84	1.25
16x16	23FA+1HA	141	16FA+6HA	114	1.24

Result

We design the proposed architecture as well as the architecture in [4] using Verilog HDL code. Both the designs were synthesized using the Cadence RTL compiler with a commercial 90nm CMOS technology and standard cell library for the calculation of power, area and delay using 200 test vectors. The results of the Area, Power, and Delay were obtained using Cadence RTL Compiler. In Table IV, proposed multiplier architecture is compared with full-width and fixed-width multiplier for area, power and delay. The power and delay of the proposed design has reduced 31.48% and 21.92% respectively as compared to full-width multiplier [4]. We further compare the proposed design with truncated multiplier [4], we achieves 5.95% and 7.81% reduction on power and delay respectively.

Table 4: Area, Power, and Delay Results (for n=12)

12x12	Area(μm^2)		Power(µW)			Delay (ps)
	Cells	Area	Leakage	Dynamic	Total	Delay
Full Width [4]	268	2061	177	38455	38632	3704
Truncated [4]	175	1255	105	28038	28144	3137
Proposed	179	1311	110	26358	26468	2892

Conclusion

In this paper a new approach is proposed for array-based fixed-width Booth multiplier that achieves noticeable reduction of delay and power compared with existing designs. This design is more suitable for inherent error resilience applications like DSP and media processing. Compared with traditional fixed-width multiplier, our analysis demonstrates that proposed design shows sensible improvements when the high speed and low power is required.

References

[1] Yeh, W.C. & Jen, C.W. 2000, "High-speed booth encoded parallel multiplier design", *IEEE Transactions on Computers*, vol. 49, no. 7, pp. 692-701.

[2] Baugh, C.R. & Wooley, B.A., "Two's Complement Parallel Array Multiplication Algorithm.", *IEEE Transactions on Computers*, vol. C-22, no. 12, pp. 1045-1047, 1973.

[3] Schulte, M. J., and Swartzlander, E. E. 1993. Truncated multiplication with correction constant [for DSP]. In VLSI Signal Processing, VI, pp. 388-396, October 1993, IEEE.

[4] Song, M. -A., Van, L. -D., Huang, T. -C. & Kuo, S. -Y. 2004, "A generalized methodology for low-error and area-time efficient fixed-width booth multipliers", *Midwest Symposium on Circuits and Systems*, pp. I9.

[5] Cho, K.-J., Lee, K.-C., Chung, J.-G. & Parhi, K.K. 2004, "Design of low-error fixed-width modified booth multiplier", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 5, pp. 522-531.

[6] Chen, Y.-H., Chang, T.-Y. & Jou, R.-Y. 2010, "A statistical error-compensated Booth multipliers and its DCT applications", *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, pp. 1146.

[7] Wang, J.-P., Kuang, S.-R. & Liang, S.-C. 2011, "High-accuracy fixed-width modified booth multipliers for lossy applications", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 1, pp. 52-60.

[8] Chen, Y.-H. & Chang, T.-Y. 2012, "A high-accuracy adaptive conditional-probability estimator for fixed-width Booth multipliers", *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 3, pp. 594-603.

[9] Balamurugan, S., Srirangaswamy, B., Marimuthu, R. & Mallick, P.S. 2012, "FPGA design and implementation of truncated multipliers using bypassing technique",*ACM International Conference Proceeding Series*, pp. 1111.

[10] Balamurugan, S., Ghosh, S., Atul, Balakumaran, S., Marimuthu, R. & Mallick, P.S. 2012, "Design of low power fixed-width multiplier with row bypassing", *IEICE Electronics Express*, vol. 9, no. 20, pp. 1568-1575.