VOL. 10, NO. 10, JUNE 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

DESIGN OF MODIFIED 32 BIT BOOTH MULTIPLIER FOR HIGH SPEED DIGITAL CIRCUITS

P. Nithiyanandham[1] and V. Balamurgan[2]
[1]M.Tech VLSI design, Sathyabama University, Chennai, India
[2]Department of Electronics and Communication Engineering, Sathyabama University, Chennai, India
E-Mail: nithiinnm@gmail.com

ABSTRACT

This paper presents the design and implementation of Advanced Modified Booth Encoding (AMBE) multiplier for both signed and unsigned 32 - bit numbers multiplication the array multiplier and Braun array multipliers perform multiplication operation on unsigned numbers only. Thus, the requirement of the modern computer system is a dedicated and very high speed unique multiplier unit for signed and unsigned numbers. Therefore, this paper presents the design and implementation of AMBE multiplier that can be suitable for the high speed digital logic circuits. The Carry looks ahead Adder (CLA) tree and the final Carry Look ahead (CLA) adder used in combination of CBEL (Common Boolean Enable Logic) to speed up the multiplier operation. Since signed and unsigned multiplication operation is performed by the same multiplier unit the required hardware and the chip area reduces and this in turn reduces power dissipation and cost of a system.

Keywords: modified booth encoding multiplier, CLA, CSA, signed-unsigned.

1. INTRODUCTION

Multiplication is a most commonly used operation in many computing systems. Infact multiplication is nothing but addition since, multiplicand adds to itself multiplier no. of times gives the multiplication value between multiplier and multiplicand. But considering the fact that this kind of implementation really takes huge hardware resources and the circuit operates at utterly low speed. In order to address this so many ideas have been presented so far for the last three decades. Each one is aimed at particular improvement according to the requirement. One may be aimed at high clock speeds and another maybe aimed for low power or less area occupation. Either way ultimate job is to come up with an efficient architecture which can address three constraints of VLSI speed, area, and power. Among these three speeds is the one which requires special attention. If we observe closely multiplication operation involves two steps one is producing partial products and adding these partial products [3]. Thus, the speed of a multiplier hardly depends on how fast generate the partial products and how fast we can add them together. If the numbers of partial products to be generated are of less than it is indirectly means that we have achieved the speed in generating partial products. Booth's algorithms are meant for this only. To speed up the addition among the partial products we need fast adder architectures. Since the multipliers have a significant impact on the performance of the entire system, many high performance algorithms and architectures have been proposed [1-12]. The very high speed and dedicated multipliers are used in pipeline and vector computers. The high speed Booth multipliers and pipelined Booth multipliers are used for digital signal processing (DSP) applications such as for multimedia and communication systems. High speed DSP computation applications such as Fast Fourier transform (FFT) require additions and multiplications. The conventional modified Booth encoding (MBE) generates an irregular partial product array because of the extra partial product bit at the least significant bit position of each partial product row. Therefore papers [4] presents a simple approach to generate a regular partial product array with fewer partial product rows and negligible overhead, thereby lowering the complexity of partial product reduction and reducing the area, delay, and power of MBE multipliers. But the drawback of this multiplier is that it functions only for signed number operands.

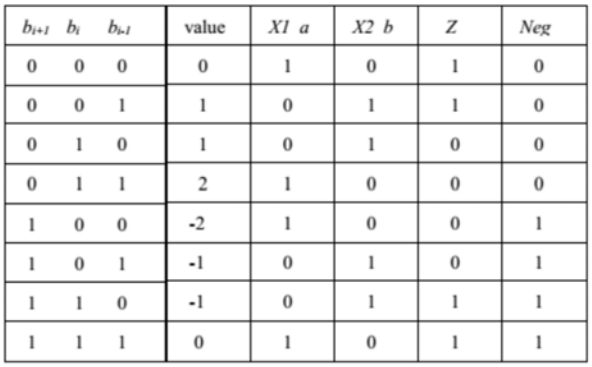
The modified-Booth algorithm is extensively used for high-speed multiplier circuits. Once, when array multipliers were used, the reduced number of generated partial products significantly improved multiplier performance. In designs based on reduction trees with logarithmic logic depth, however, the reduced number of partial products has a limited impact on overall performance. The Baugh-Wooley algorithm [7, 8, 9] is a different scheme for signed multiplication, but is not so widely adopted because it may be complicated to deploy on irregular reduction trees. Again the Baugh-Wooley algorithm is for only signed number multiplication. The array multipliers and Braun array multipliers [10] operates only on the unsigned numbers. Thus, the requirement of the modern computer system is a dedicated and very high speed multiplier unit that can perform multiplication operation on signed as well as unsigned numbers. In this paper we designed and implemented a dedicated multiplier unit that can perform multiplication operation on both signed and unsigned numbers, and this multiplier is called as AMBE multiplier.

2. CONVENTIONAL MODIFIED BOOTH MULTIPLIER

A. Algorithm of the Modified Booth Multiplier

Multiplication consists of three steps: 1) the first step to generate the partial products; 2) the second step to add the generated partial products until the last two rows are remained; 3) the third step to compute the final multiplication results by adding the last two rows. The modified Booth algorithm reduces the number of partial products by half in the first step. We used the modified Booth encoding (MBE) scheme proposed in [2]. It is known as the most efficient Booth encoding and decoding scheme. To multiply X by Y using the modified Booth algorithm starts from grouping Y by three bits and encoding into one of {-2, -1, 0, 1, 2}. Table I shows the rules to generate the encoded signals by MBE scheme and Fig. 1 (a) shows the corresponding logic diagram. The Booth decoder generates the partial products using the encoded signals as shown in Figure-1(b).

Table-1. Truth table of MBE scheme.



b_{i+1}	b_i	b_{i-1}	value	X1 a	X2 b	Z	Neg
0	0	0	0	1	0	1	0
0	0	1	1	0	1	1	0
0	1	0	1	0	1	0	0
0	1	1	2	1	0	0	0
1	0	0	-2	1	0	0	1
1	0	1	-1	0	1	0	1
1	1	0	-1	0	1	1	1
1	1	1	0	1	0	1	1

The new MBE recorder [2] was designed according to the following analysis. Table-1 presents the truth table of the new encoding scheme. The Z signal makes the output zero to compensate the incorrect X2_b and Neg signals. Figure-1 presents the circuit diagram of the encoder and decoder. The encoder generates X1_b, X2_b, and Z signals by encoding the three x-signals. The y LSB signal is the LSB of the y signal and is combined with x-signals to determine the Row_LSB and the Neg_cin signals. Similarly, y-MSB is combined with x-signals to determine the sign extension signals.

The Figure-2(a) has widely been adopted in parallel multipliers since it can reduce the number of partial product rows to be added by half, thus reducing the size and enhancing the speed of the reduction tree. However, as shown in Figure-1(a), the conventional MBE algorithm generates n/2 + 1 partial product rows rather than n/2 due to the extra partial product bit (neg bit) at the least significant bit position of each partial product row for negative encoding, leading to an irregular partial product array and a complex reduction tree. Therefore, the Modified Booth multipliers with a regular partial product array [4] produce a very regular partial product array, as shown in Figure-3. Not only each neg is shifted left and replaced by ci but also the last neg bit is removed. This approach reduces the partial product rows from n/2 + 1 to n/2 by incorporating the last neg bit into the sign extension bits of the first partial product row, and almost no overhead is introduced to the partial product generator. More regular partial product array and fewer partial product rows result in a small and fast reduction tree, so that the area, delay, and power of MBE multipliers can further be reduced.

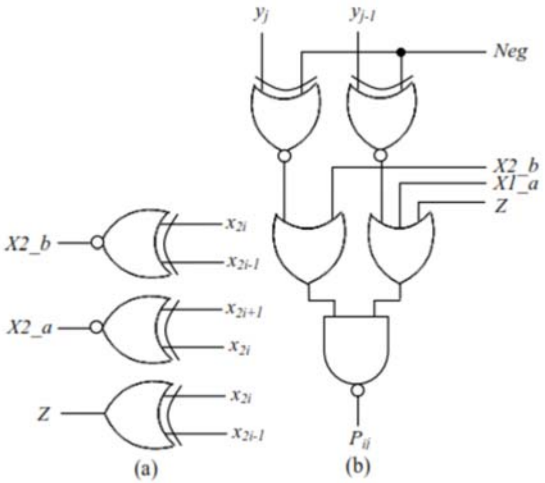


Figure-1. The Encoder and Decoder for the new MBE scheme. (a) Simple encoder (b) Decoder.

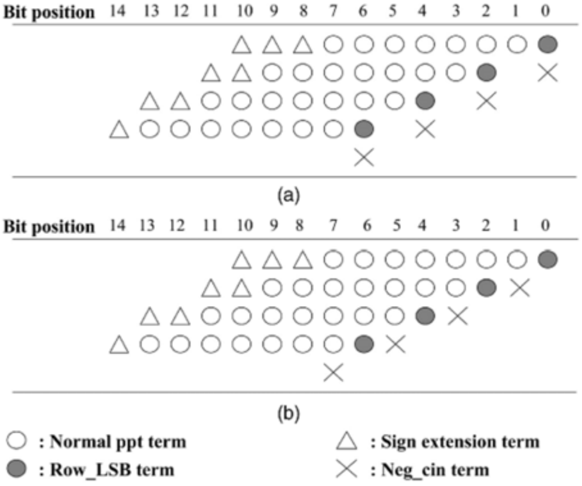

Bit position 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
(a)
Bit position 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
(b)
○ : Normal ppt term
● : Row_LSB term
△ : Sign extension term
✕ : Neg_cin term

Figure-2. 8× 8 MBE partial product array. (a) Traditional MBE partial product array. (b) New MBE partial product array.

Figure-3 shows the generated partial products and sign extension scheme of the 8-bit modified Booth multiplier. The partial products generated by the modified Booth algorithm are added in parallel using the Wallace tree until the last two rows are remained. The final multiplication results are generated by adding the last two rows. The carry propagation adder is usually used in this step.

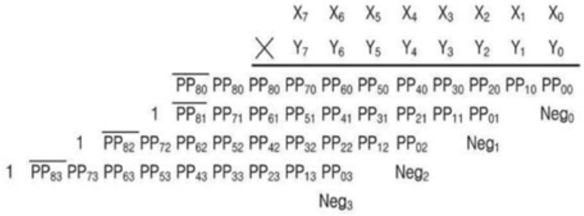

X_7 X_6 X_5 X_4 X_3 X_2 X_1 X_0
× Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0
$\overline{PP_{80}}$ PP_{80} PP_{80} PP_{70} PP_{60} PP_{50} PP_{40} PP_{30} PP_{20} PP_{10} PP_{00}
1 $\overline{PP_{81}}$ PP_{71} PP_{61} PP_{51} PP_{41} PP_{31} PP_{21} PP_{11} PP_{01} Neg_0
1 $\overline{PP_{82}}$ PP_{72} PP_{62} PP_{52} PP_{42} PP_{32} PP_{22} PP_{12} PP_{02} Neg_1
1 $\overline{PP_{83}}$ PP_{73} PP_{63} PP_{53} PP_{43} PP_{33} PP_{23} PP_{13} PP_{03} Neg_2
Neg_3

Figure-3. Generated partial products and sign extension scheme.

B. Architecture of the Modified Booth Multiplier

Figure-4 shows the architecture of the commonly used modified Booth multiplier. The inputs of the multiplier are multiplicand X and multiplier Y. The Booth encoder encodes input Y and derives the encoded signals as shown in Figure-1 (a). The Booth decoder generates the partial products according to the logic diagram in Figure-1(b) using the encoded signals and the other input X. The Wallace tree computes the last two rows by adding the generated partial products. The last two rows are added to generate the final multiplication results using the carry look-ahead adder (CLA).

3. PROPOSED AMBE MULTIPLIER WITH CLA TREE

The main goal of this paper is to design and implement 32×32 multiplier for signed and unsigned numbers using MBE technique. The output from the booth encoder is being sent to the Wallace tree and in the above mentioned systems we use the CSA tree and the final adder is being the CLA In order to consume the less power the replaced with the CLA tree has been proposed. After the calculation of the partial products, these partial products are made into the CLA Tree. The proposed system of this tree are calculated and carry will be propagated parallel mechanisms and finally then made into the CLA logics. CLA mechanisms can be integrated into the AMBE for the sharing the Common Boolean Logics.

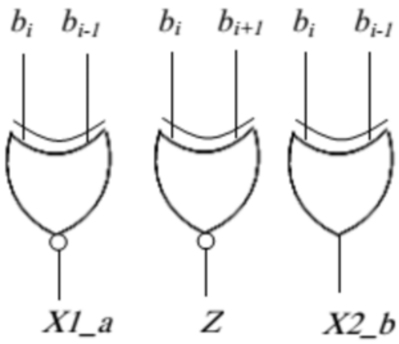


Figure-4. Logic diagram for AMBE.

4. SIMULATION RESULT

The modified AMBE with the mechanisms have been included for the Xilinx ISE14, 5 and simulated with the ISIM for this 32-bit Multiplication logic.

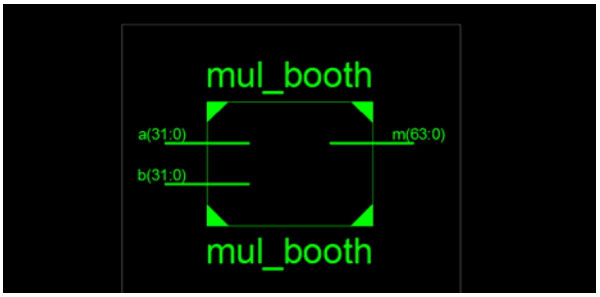
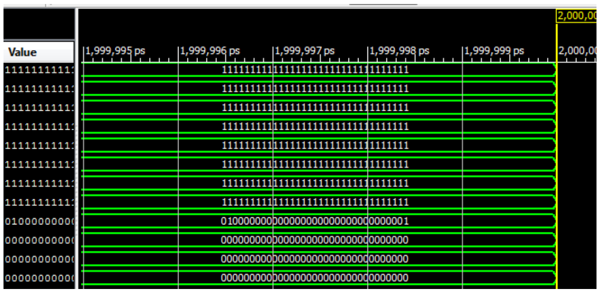


Figure-5. RTL Schematic for the AMBE with CLA.



```
11111111111  1111111111111111111111111111111111
11111111111  1111111111111111111111111111111111
11111111111  1111111111111111111111111111111111
11111111111  1111111111111111111111111111111111
11111111111  1111111111111111111111111111111111
11111111111  1111111111111111111111111111111111
11111111111  1111111111111111111111111111111111
11111111111  1111111111111111111111111111111111
01000000000  0100000000000000000000000000000001
00000000000  0000000000000000000000000000000000
00000000000  0000000000000000000000000000000000
00000000000  0000000000000000000000000000000000
```

Figure-6. Simulation result for Multiplier ISIM.

Comparative Result Analysis

S. No.	Tree structure	Area (No. of Slices)	Power (mW)
1	CSA Tree	2334	0.085
2	CLA Tree	163	0.081

5. CONCLUSIONS

The paper proposes the new CLA tree structure for the AMBE which provides the less power efficient and less area computations. The designed AMBE can be used for the high speed networks solutions.

REFERENCES

[1] W. -C. Yeh and C. -W. Jen. 2000. High Speed Booth encoded Parallel Multiplier Design. IEEE transactions on computers. 49(7): 692-701.

[2] Shiann-Rong Kuang, Jiun-Ping Wang and Cang-Yuan Guo. 2009. Modified Booth multipliers with a Regular Partial Product Array. IEEE Transactions on circuits and systems-II. 56(5).

[3] Li-Rong Wang, Shyh-Jye Jou and Chung-Len Lee. 2008. A well-tructured Modified Booth Multiplier Design. 978-1-4244-1617-2/08/$25.00 ©2008 IEEE.

[4] Soojin Kim and Kyeongsoon Cho. 2010. Design of High-speed Modified Booth Multipliers Operating at GHz Ranges. World academy of Science, Engineering and Technology. 61

[5] Magnus Sjalander and Per Larson-Edefors. 2008. The Case for HPM-Based Baugh-Wooley Multipliers. Chalmers University of Technology, Sweden.

[6] J. Fadavi-Ardekani. 1993. ªM×N Booth Encoded Multiplier Generator Using Optimized Wallace Trees,º IEEE Trans. VLSI Systems. 1(2).

[7] Wang G. 2004. A unified unsigned/signed binary multiplier. The Thirty-Eighth Asilomar Conference on Signals, Systems and Computers. 1: 513-516.

[8] Kim J. Y. 1999. Multiplier to selectively perform unsigned magnitude multiplication or signed magnitude multiplication. USpatent. 5, 870, 322.

[9] Hwang-Cherng Chow and I-Chyn Wey. 2002. A 3.3V 1GHz high speed pipelined Booth multiplier. Proc. of IEEE ISCAS. 1: 457-460.

[10] M. Aguirre-Hernandez and M. Linarse-Aranda. Energy-efficient high-speed CMOS pipelined multiplier. Proc. of IEEE.