ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

Design of multiplier using regular partial products.

Bipin[1], Ms. Sakshi[2]

M-Tech VLSI Design Thapar University, Patiala, India[1]

Assistant Professor, ECE Department, Thapar University, Patiala, India[2]

Abstract: The conventional Modified Booth Encoding (MBE) generates n/2+1 rows instead of n/2 rows and also irregular partial product array because of the extra partial product bit at the LSB position of each partial product row. In this brief, a simple approach is proposed to generate n/2 partial product rows along with a regular partial product arrays and negligible overhead, thereby lowering the complexity of partial product and reducing the area and power of MBE multipliers. The proposed approach can also be utilized to regularize the partial product array of MBE multipliers along with the issue of disposal of the negative partial products efficiently by computing the 2's complement thereby avoiding the additional adder for adding 1 and generation of long carry chain. The proposed mechanism also continues to support the concept of reducing the partial product from n/2 +1 partial products achieved via modified booths algorithm to n/2.In this direct two's complement method has been used to reduce partial product rows from n/2+1 to n/2.Implementation results demonstrate that the proposed MBE multipliers with a regular partial product array really achieve significant improvement in area and power consumption when compared with conventional MBE multipliers. Here different multipliers have been designed and compared. These multipliers have been designed with the help of Verilog, simulated on Modelsim SE 6.3f and synthesized on Xilinx FPGA Spartan 3E xc3s500E, that helps in comparing their area, power and delay.

Keywords: DSP, MBE, PP, VLSI

I. INTRODUCTION

Enhancing the processing performance and reducing the power dissipation of the systems are the most important design challenges for multimedia and digital signal processing (DSP) applications, in which multipliers frequently dominate the system's performance and power dissipation. Multipliers are widely used in DSP and multimedia applications. Hence modifications are made to the multiplier architecture to achieve all those requirements. Multipliers have large area, long latency and consume considerable power. Therefore high speed multiplier design has been an important part in high speed VLSI system design. Multiplication consists of three major steps:-

1) Generation of partial products.
2) Summing up all partial products until only two rows remain.
3) Adding the remaining two rows of partial products by using a carry propagation adder (eg. Ripple Carry Adder).

There are number of technique had been developed in the past for these three steps to enhance the performance of multiplier. In this brief we will concentrate on first step i.e. generation of partial product to improve speed, area and power. Booth's algorithm is considered to be best known algorithm for generation of the partial products [2], [3]. Booth's algorithm deals with both sign and unsigned number. In order to improve the speed and performance of the system a new technique was introduced known as Modified Booth's Algorithm. Modified Radix -4 Booth's Algorithm is made use for fast multiplication. The salient features of this algorithm are:

• Only n /2 clock cycles are needed for n-bit multiplication as compared to n clock cycles in Booth's algorithm.
• Isolated 0/ 1 are handled efficiently.

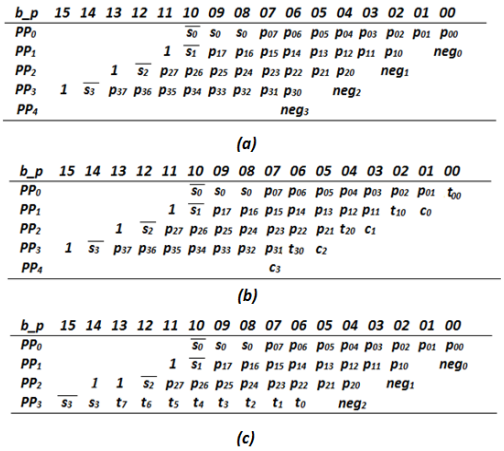

Fig 1 Conventional MBE partial product array for 8*8 bit multiplication

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

In order to generate more regular partial product some approaches [4], [5] have been proposed, as shown in Fig. 1(b) and 1(c), for the MBE multipliers. Thus, the area, delay, and power consumption of the reduction tree, as well as the whole MBE multiplier can be reduced.

The proposed work will combine the method proposed in [4] to generate a parallelogram-shaped partial product array and reduce n/2+1 partial product rows to n/2 by the method proposed in [1]. 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. Elimination of last partial product row will save an extra adder circuitry to add last two partial product rows results in improvement in area, power and delay.

II. MODIFIED BOOTH ALGORITHM

As discussed above MBE results into an extra partial product row, this additional partial product row is required to take care of the negative encoding ,which would hence, require more adder circuitry while implementing and also affect the speed efficiency of the system. The booth encoding algorithm is a bit-pair encoding algorithm that generates partial products which are multiples of the multiplicand. The booth algorithm shifts and/or complements the multiplicand (a operand) based on the bit patterns of the multiplier (b operand). Essentially, three multiplier bits [b (i+1), b (i) and b (i-1)] are encoded into nine bits that are used to select multiples of the multiplicand {-2a, -a, 0, +a, +2a}. The three multiplier bits consist of a new bit pair [b (i+1), b (i)] and the leftmost bit from the previously encoded bit pair [b (i-1)] as shown in figure 2.1.

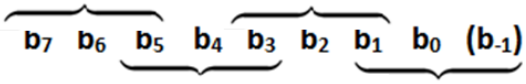

Fig. 2.1: Recoding in Radix 4

Recoding scheme of Modified Booth Algorithm can be easily explained by Table 1 proposed by booth in 1951.

Table 1
MBE Table

b_{2i+1}	b_{2i}	b_{2i+1}	operation	neg	2a	a
0	0	0	0	0	0	0
0	0	1	a	0	0	1
0	1	0	a	0	0	1
0	1	1	2a	0	1	0
1	0	0	-2a	1	1	0
1	0	1	-a	1	0	1
1	1	0	-a	1	0	1
1	1	1	0	0	0	0

Here neg bit represent the negative encoding. 2a can be achieved by left shifting multiplicand by one bit. There are, number of sign extension schemes [6]–[7] have been proposed to prevent extending up the sign bit of each row to the $(2n-1)^{th}$ bit position.

III. PROPOSED TECHNIQUE

Fig. 1(a) illustrates the MBE partial product array for an 8 × 8 multiplier with a sign extension prevention procedure, where s_i represents sign bit of the partial product row PPi, $\overline{s}_i$ is the complement of s_i, and b_p represents the bit position. As can be Seen in Fig. 1(a), MBE generates n/2+1 partial product rows where last partial product row represent neg_i bits which results in an irregular partial product array and one additional partial product row. To get more regular structure, least significant part of each partial product row PP_i i.e.pp_{i0} , is added with neg_i in advance and obtained a new least significant bit τ_{i0} and carry c_i [4] . In proposed multiplier addition of pp_{i0} and neg_i bit continue up to 3 partial product rows shown in fig. 3.1 and a direct method to find 2's complement [1] is implemented on last two partial product rows. Hardware realization of the 2's compliment becomes critical part due to the need of arithmetic addition of plus 1 which many times cause propagation of carry from LSB to MSB. So a new technique has been proposed in next section.

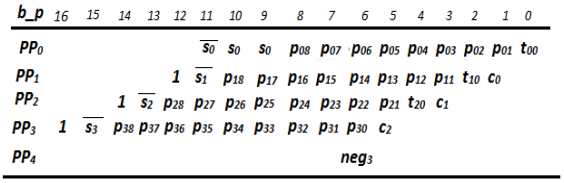

Fig. 3.1 Partial Product after adding pp_{i0} and neg_i

A. How to find 2's complement

Basic mechanism to find 2's complement is to firstly take the 1's complement and then adding 1, which is an inefficient operation because of generation of long carry chain. So in order to enhance the performance of multiplier efficient disposal system of adding 1 is needed. There are number of techniques available discussed below.
(i)The first and obvious method is to enhance the capability and performance of the adder used for the addition of 1 while calculating the 2's complement whenever needed. But this method is not efficient in terms of additional hardware requirement and extra time required.
(ii)One method is where all the bits after the rightmost "1" in the word are complemented but all the other bits are unchanged. An extended version of this algorithm was proposed in [5].
(iii)Another disposal method was suggested by Zheng and other co-authors in 2010 keeping the basic principle of not

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

deterring the value of the partial products. They partition the partial products into pp and C_{in} such that PP= pp + C-in, such that positive PPs will be PP=pp and C_{in}=0 [3].

(iv) Consider a 8 bit number A, to calculate 2's complement first is to invert all the bits of data , and denote them Abar. Next step is to perform "XOR" operation on Abar(0) with 1'b1, Abar(1) xor Abar(0), Abar(2) xor Abar(1) and so on, and denote them Axor. Now scan each bit of Axor until1'b1 arrives, copy those bits into results and remaining bits will be copied from Abar.

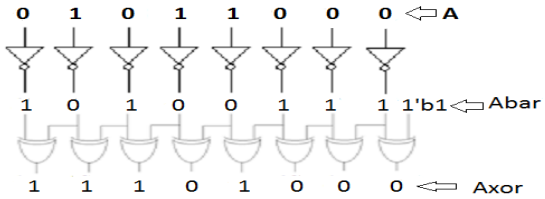

Fig. 3.2(a) shows xoring operation

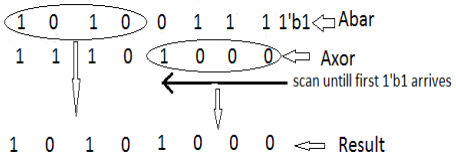

Fig. 3.2(b) shows method to find 2's complement of 8 bit data

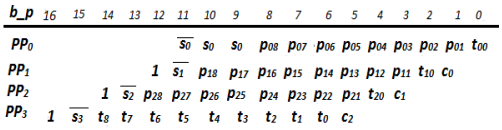

Fig.3.3 Final partial products after applying 2's complement

IV. EXPERIMENTAL RESULTS

Fig. 3.3 shows final four partial product rows after combining two above techniques. For comparisons several multiplier designed were implemented (i.e 8 bit and 16 bit) by using different approaches. These multipliers were modeled in verilog HDL and synthesis by using Xilinx 13.1 and also area and power were estimated by using Synopsys design compiler.

The implementation results show Dynamic Power, Hardware Area, and Delay.

Table 2
Experimental Results

	Simple MBE (8x8 bit)	**Proposed MBE (8x8 bit)**	**Simple MBE (16x16 bit)**	**Proposed MBE (16x16 bit)**
Maximum combination-al delay	21.579	21.104	38.852	38.491
Area (1 unit = 1 nand gate area)	5446.5	4065.35	18,845.71	16,164.59
Dynamic power (mW)	4.0150	2.842	20.255	17.06
No of slices	91/4656	108/4656	295/4656	334/4656
Maximum frequency	79.59 MHz	81.38 MHz	53.86 MHz	64.57 MHz

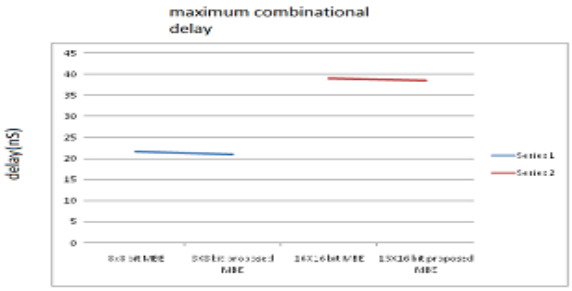

Fig 4.1 Combinational path delay comparison of 8*8, 16*16 MBE and proposed MBE

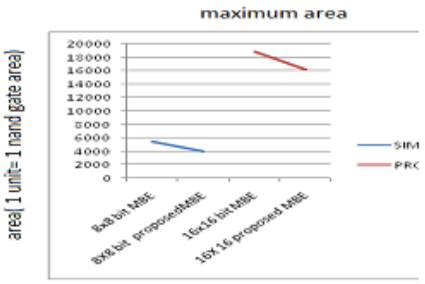

Fig 4.2 Area comparison of 8*8, 16*16 MBE and proposed MBE

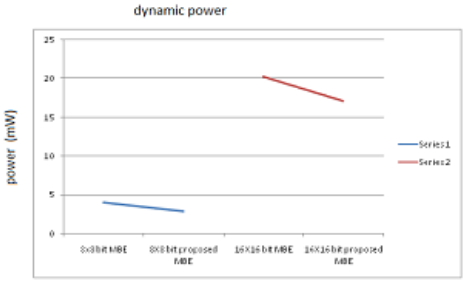

Fig 4.3 Dynamic power comparison of 8x8, 16x16 MBE and proposed MBE.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

V. CONCLUSION

This paper mainly deals with the first step of partial product generation. Here an efficient technique has been proposed to find 2's complement and to generate regular partial products. Experimental results have demonstrated that the proposed MBE with regular partial product arrays can achieve significant improvement in area, delay, and power consumption when compared with conventional multiplier.

ACKNOWLEDGMENT

First of all, I would like to express my gratitude to **Ms. Sakshi,** Assistant Professor, Electronics and Communication Engineering Department, Thapar University, Patiala for her patient guidance and support throughout this paper. I am truly very fortunate to have the opportunity to work with her. I found this guidance to be extremely valuable.

I am also thankful to our HEAD OF THE DEPARTMENT, **Dr. Rajesh Khanna** as well as PG Coordinator, **Dr. KulbirSingh,** Associate Professor, of Electronics and Communication Engineering Department. I would like to thank entire faculty and staff of Electronics and Communication Engineering Department and then friends who devoted their valuable time and helped me in all possible ways towards successful completion of this work. I thank all those who have contributed directly or indirectly to this work.

REFERENCES

[1] M. Kumar and R. Verma, "Disposition (reduction) of (negative) partial product for Radix 4 Booth's Algorithm," in *IEEE World Congress on Information And Communication Technologies*, Dec 2011, pp.1169-1174.

[2] S.R. Kuang, J.P. Wang, and C.Y. Guo, "Modified Booth Multipliers with a regular partial product array," in *IEEE Transaction on circuits and systems-II: Express Briefs*, vol. 56, no. 5, May 2009,pp.404-408.

[3] G. li, H.. Chen, X. Yang "Research on disposal of negative partial products for booth algorithm," in Proc. *IEEE Conference on Information Theory and Information Security (ICITIS)*, Dec 2010, pp 1115-1117.

[4] W.C. Yeh and C.W. Jen, "High-speed booth encoded parallel multiplier design," in *IEEE Trans. Computer society*, vol. 49, no. 7, Jul. 2000, pp. 692–701.

[5] J.Y. Kang and J.L. Gaudiot "A simple high-speed multiplier design ,"*IEEE Trans. Computer society*, vol. 55, no. 10, Oct. 2006 ,pp. 1253–1258.

[6] O. Salomon, J.M. Green, and H. Klar, "General algorithms for a simplified addition of 2's complement numbers," in *IEEE Solid-State Circuits*, vol. 30, no. 7, Jul. 1995,pp. 839–844.

[7] E. de Angel and E. E. Swartzlander, "Low power parallel multipliers," *IEEE Workshop on VLSI signal process. IX*, Oct 1996, pp. 199–208.