IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 2, Issue 2 (Mar. – Apr. 2013), PP 30-34
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Novel Booth Encoder and Decoder for Parallel Multiplier Design

Mamtha Prajapati[1], Saroj Kumar Lenka[2]
[1](*Department of Electronics and Communication Engineering, MITS (Deemed University), India)*
[2](*Department of Information Technology, MITS (Deemed University), India)*

Abstract : *Fast multipliers are essential components of most VLSI applications like digital signal processing systems, microprocessors, etc. The speed of multiplier operation is of fastidious importance within the general-purpose processors. The essential multiplication principle is twofold i.e., Evaluation of partial product and accumulation of the shifted partial products with the motivation to Booth's algorithm. In this pithy, an efficient design of modified Booth Encoder and Decoder scheme for high performance of parallel multiplier has been proposed. The proposed Booth encoder and Decoder logic are competitive with the present schemes and shows enhancements in delay. It additionally shows a substantial reduction in area.*

Keywords - *FPGA (Field Programmable Gate Array), Partial Product Generator (PPG), Radix-4 Modified Booth Encoder (MBE), Sign extension.*

I. INTRODUCTION

Most of the Very Large Scale Integration (VLSI) circuit applications use arithmetic operations widely. The performance in most cases, powerfully depend on the effectiveness of the hardware used for computing multiplications, since multiplication is massively utilized in these environments. The high interest in this application field is witnessed by the number of algorithms and implementations of the multiplication operation that are presented within the literature. More specifically, short-bit width multipliers became very important building blocks for high-performance embedded processors and DSP execution cores. In this case, short bit-width multipliers typically play the role of basic building blocks. Multipliers of moderate bit-width (less than 32 bits) are also being used massively in FPGAs [1]. All the above interprets into a high interest and motivation, for the design of high-performance short or moderate bit-width multipliers.

There are three major steps to any multiplication. In the first step, the partial products are generated, in the second step, the partial products are reduced to one row of final sums and one row of final carries and in the third step, the final sums and carries are added to generate the result. There has been abundant work on advanced multiplication algorithms and designs [1] -[14]. Most of the approaches utilize the Modified Booth Encoding approach [3] for the first step due to its ability to reduce the number of partial product rows in half. They then choose some variation of any one of partial product reduction schemes such as the Wallace tree [5], [6] or the compressor tree [7], [8] in the second step to rapidly reduce the number of partial product rows to the final two (sums and carries). In the third step, they use some kind of advanced adder approach such as carry-look ahead or carry-select adders to add the final two rows, resulting in the final product.

With advances in technology, several researchers have tried and are trying to design multipliers which provide either of the subsequent design intensions as high speed, low power utilization, regularity of layout and thus less area or perhaps a combination of them in one multiplier therefore making them appropriate for varied high speed, low power and compact VLSI implementation. To achieve high performance, the modified Booth encoding [3] that reduces the number of partial products by a factor of two through performing the multiplier recoding has been widely adopted in parallel multipliers. This algorithmic approach is applicable for each signed and unsigned multiplication.

This paper is organized as follows. After the introduction in Section I, a brief review of the parallel Booth multiplier design is presented in Section II. Section III discusses the related work. The proposed work is presented in Section IV. Section V deals with the VLSI implementation and Section VI include the performance analysis.

II. PARALLEL MULTIPLIER DESIGN

The block diagram of Modified Booth Multiplier is shown in Fig. 1. The basic building blocks of this multiplier are Modified Booth Encoder (MBE) and partial product generator (PPG) also called as Booth Decoder. In Booth's multiplication algorithm, negative encoding is performed for both signed and unsigned multiplication. Since it applies to 2's complement arithmetic, care must be taken to verify sign extensions are in place. This work is done by the sign extension corrector which is performed along with the decoder logic. The generated partial products along with the sign extension bits are accumulated in the accumulation phase.

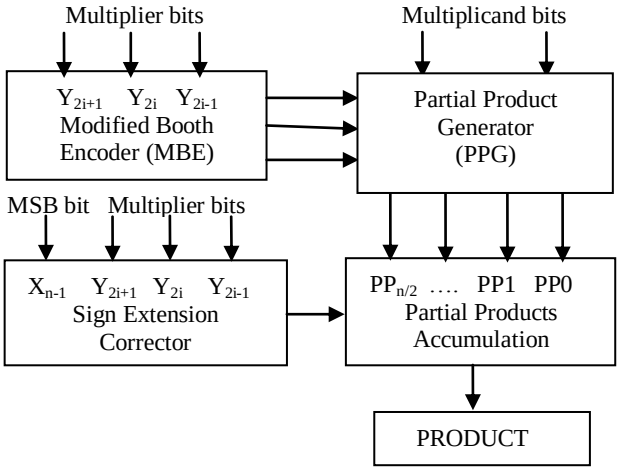

Fig. 1: Block Diagram of Modified Booth Multiplier.

The drawbacks of the conventional Booth algorithm [2] are overcome by processing 3 bits at a time during recoding in [3]. The modified Booth algorithm is also known as Booth 2 algorithm or Modified radix-4 Booth algorithm. It is a well-known algorithm as it reduces the number of partial products by about a factor of two. To Booth recode the multiplier term, we consider the bits in blocks of three, such that each block overlaps the previous block by one bit as shown in Fig. 2. Grouping starts from the LSB, and the first block use only two bits of the multiplier (since there is no previous block to overlap):

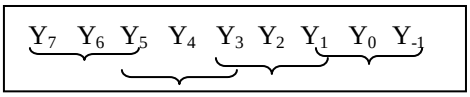

Fig. 2: Grouping bits of the multiplier term for modified Booth recoding.

The least significant block uses only two bits of the multiplier, and assumes a zero for the third bit. The overlap is necessary to know the sign of the last block, as the MSB bit acts like a sign bit. We then consult the Table 1, to decide what the encoding will be. Since the LSB of the first block is always assumed to be 0, there is never a negative partial product for the least significant block.

Table 1: Modified booth recoding for each of the possible values [3]

Y_{2i+1}	Y_{2i}	Y_{2i-1}	Operation
0	0	0	0
0	0	1	+X
0	1	0	+X
0	1	1	+2X
1	0	0	-2X
1	0	1	-X
1	1	0	-X
1	1	1	0

2.1 Booth Decoder

Booth Decoder is designed to produce the partial products by multiplying the multiplicand, X by 0, 1, -1, 2 or -2. The output of MBE acts as the selection inputs to the partial product generator. The partial products on each row are obtained as 1's complement numbers for negative encoding. To obtain the 2's complement number, '1' is to be added at the LSB of the obtained partial product. This operation is performed in the accumulation phase. Each partial product row is placed 2-bits to the left with respect to the previous row.

2.2 Sign extension Corrector

In Booth's multiplication algorithm, negative encoding is performed for both signed and unsigned multiplication. The reason for dealing with signed numbers incorrectly is the absence of sign bit expansion. When bit 7 of the multiplicand X (X_7) is zero (unsigned number) and Y_{2i+1} is equal to one, then sign E will have one value (become signed number of resulted partial products). It is the same when the bit 7 of the multiplicand

X (X_7) is one (signed number) and Y_{2i+1}, is equal to zero, the sign E will have a new value. However, when both the value of X_7 and Y_{2i+1} are equal either to zero or one, the sign E will have a value zero (unsigned number). In the case, when all three bits of the multiplier value Y_{2i+1}, Y_{2i} and Y_{2i-1} are equal to zero or one, the sign E will direct have a zero value independent of the X_7 value. In this brief, the focus is on the first step (i.e., Modified Booth encoder and decoder) to reduce the area and delay of multipliers, as the MBE and PPG block occupy one-third part of the complete multiplier architecture shown in Fig. 1.

III. RELATED WORK

Two different Booth encoder and decoder schemes are evaluated here for their distinct unique features. Goto introduced a sign-select Booth encoder with four output signals [4]. In the usual encoder, only *neg* is generated which represents the negative partial products. In Goto's sign-select encoder, an extra control signal PL_j is added to represent the selection of positive partial product. PL_j and M_j become active when the partial product is positive and negative, respectively. The X_j and $2X_j$ signals show whether or not the partial product is doubled.

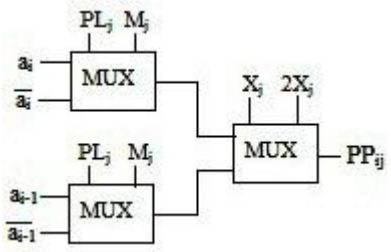

Fig. 3: Booth Decoder [15].

The sign-select booth encoder uses many inverters to reduce the delay imbalances but the area occupied is large. This scheme is represented as the *Scheme_1*. Fig. 3 illustrates the simplified PPG than the one presented in [4]. The simplicity of this circuit leads to smaller area usage. The second scheme represented as the *Scheme_2* was proposed in [9] by Yeh. In this scheme the modified Booth encoder produces four output signals namely *neg, X1_b, Z* and *X2_b.* In this scheme, the *neg* bit is as the input multiplier bit $Y_{2i+1.}$ So, the encoder generates the three signals *X1_b, X2_b* and *Z.* The *Z* signal makes the output zero to compensate the incorrect *X2_b* and *neg* signals. The PPG proposed in [9] is the improved form of the PPG proposed by Fried in [14]. The PPG in [9] has the same speed performance as of that in [14] but occupies moderate area.

IV. PROPOSED WORK

As the focus is on high performance in terms of delay and area of the modified Booth encoder and PPG block of the parallel Multiplier, the proposed Encoding and PPG scheme when put in the structure presented in Fig. 5 generates the partial products with less delay and occupies a moderate area when compared to the above discussed schemes. The proposed modified Booth encoding table is shown in Table 2. In the *Proposed scheme*, the *neg* signal is same as the input bit $Y_{2i+1.}$ The signals *one* and *two* are generated as shown in Fig. 4a. The signal *cor* in the modified Booth encoding table is the correction bit used to compensate the incorrectness of the signal *two.* The PPG is designed using two multiplexers and an XOR gate as shown in Fig. 4b.

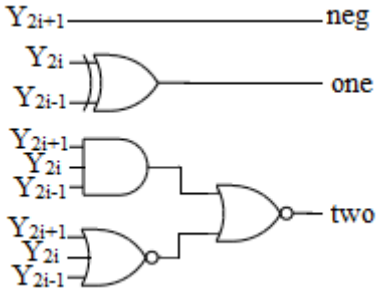

Fig. 4a: Proposed Modified Booth Encoder.

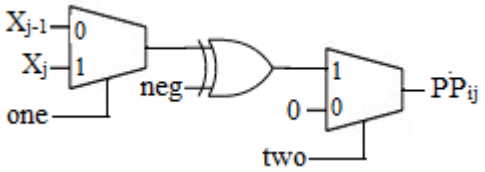

Fig. 4b: Proposed Booth Decoder.

Table 2: Proposed modified booth encoding.

Y_{2i+1}	Y_{2i}	Y_{2i-1}	neg	one	two	cor
0	0	0	0	0	0	0
0	0	1	0	1	1	0
0	1	0	0	1	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	0
1	1	1	1	0	0	0

It is to be observed that in Fig. 4b; in the second multiplexer one of the input is '0' instead of the signal *cor* to compensate the incorrectness of signal *two*. The desired functionality is obtained by using the bit '0', thereby reducing the area overhead of generating the signal *cor*. This novel Encoder and decoder perform better in terms of speed and the gate count is also reduced when compared to the schemes discussed in the previous section.

V. VLSI IMPLEMENTATION

The structure shown in Fig. 5 is designed using VHDL for the existing schemes and the proposed scheme. The implementation i.e., Synthesis and Simulation are performed on Xilinx 9.1 ISE. The device used is XC2S100-5TQ144 of family Spartan-II FPGA. If the multiplicand or the multiplier is a signed number then it is first converted to 2's complement form and then the recoding is performed by using modified Booth encoder.

Area analysis is done by analyzing the parameters like no. of 4-input LUTs (FPGA function generators), no. of occupied slices and the total equivalent gate count which are obtained in the design summary upon implementing the design, as shown in Table 3 and Table 4.

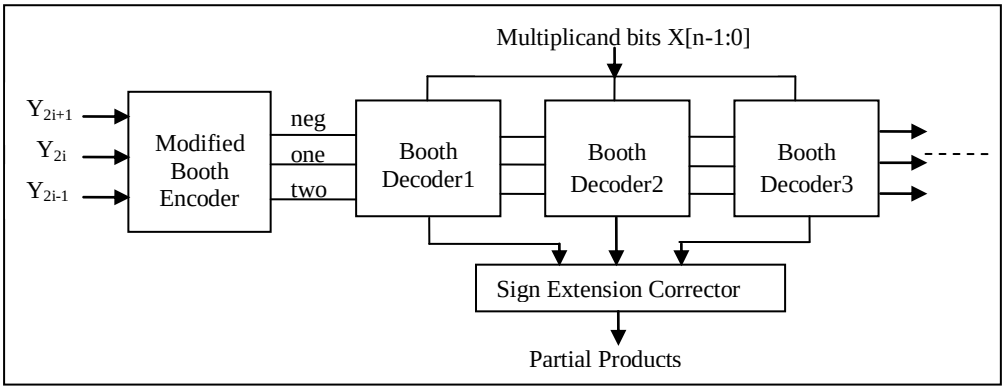

Fig. 5: Structure for generating the partial products.

Table 3: FPGA Resource Utilization For MBE Schemes For Spartan-II (Package: tq144, Speed Grade: -5)

Schemes	Delay (ns)	No. of 4-input LUT's	No. of occupied slices	Total equivalent gate count
Scheme_1	10.134	5	3	30
Scheme_2	9.884	4	2	24
Proposed scheme	9.884	3	2	18

Table 4: FPGA Resource Utilization For Generating Partial Products Of 4×4 Multiplication For Spartan-II (Package: tq144, Speed Grade: -5)

Schemes	**Delay (ns)**	**No. of 4-input LUT's**	**No. of occupied slices**	**Total equivalent gate count**
Scheme_1	13.037	15	8	90
Scheme_2	13.492	17	9	111
Proposed scheme	11.634	16	8	105

VI. RESULTS

For MBE schemes, the *Proposed scheme* gives 2.46% improvement in delay over *Scheme_1*. And gives 40% and 25% improvement in area over *Scheme_1 and Scheme_2*, respectively. Upon implementing the structure in Fig. 5, the *Proposed scheme* gives 10.76% and 13.77% improvement in delay over *Scheme_1 and Scheme_2* and 5.4% improvement in area over *Scheme_2*. Though the area occupied by *Scheme_1* is less, but the delay is more. Hence, the experimental results demonstrate that the *Proposed scheme* achieves significant improvement in area and delay than the existing ones as shown in Table 5. The proposed logic can be successfully used to build a high performance 4×4 bit multiplier. Similarly, the higher bit-width multipliers can also be built.

Table 5: Performance Analysis of Schemes

Schemes	**Delay**	**Area**
Scheme_1	Large	Less
Scheme_2	Large	More
Proposed scheme	Less	Moderate

REFERENCES

[1] F. Lamberti, N. Andrikos, E. Antelo, and P. Montuschi, Reducing the Computation Time in (Short Bit-Width) Two's Complement Multipliers, IEEE Trans. on Computers, vol. 60, no. 2, pp. 148-156, Feb. 2011.

[2] A. D. Booth, A Signed Binary Multiplication Technique, Quarterly Journal of Mechanics and Applied Mathematics, vol. 4, no. 2, pp. 236–240, June 1951.

[3] 0. L. MacSorley, High-speed arithmetic in binary computers, Proc. IRE, Jan. 1961, vol. 49, pp. 67-91.

[4] G. Goto, *et. al.*, A 4.1-ns Compact 54×54-b Multiplier Utilizing Sign-Select Booth Encoders, IEEE J. of Solid-State Circuits, vol. 32, no. 11, pp. 1676-1681, Nov. 1997.

[5] C.S.Wallace, A Suggestion for a Fast Multiplier, IEEE Trans. on Computers, vol. 13, no. 2, pp. 14-17, 1964.

[6] Ron S. Waters and Earl E. Swartzlander, A Reduced Complexity Wallace Multiplier Reduction, IEEE Trans. on Computers, vol. 59, no. 8, pp. 1134-1137, August 2010.

[7] J.-Y. Kang, W.-H. Lee, and T.-D. Han, A Design of a Multiplier Module Generator Using 4-2 Compressor, Proc. Korea Inst. Of Telematics and Electronics (KITE) Fall Conf., 1993, vol. 16, pp. 388-392.

[8] Ki-seon Cho, Jong-on Park, Jin-seok Hong, Goang-seog Choi, 54x54-bit Radix-4 Multiplier based on Modified Booth Algorithm, ACM, Proceedings of the 13th ACM Great Lakes symposium on VLSI, April 2003, pp. 233-236.

[9] W.-C. Yeh and C.-W. Jen, High-Speed Booth Encoded Parallel Multiplier Design, IEEE Trans. on Computers, vol. 49, no. 7, pp. 692-701, July 2000.

[10] Kyung-Ju Cho, *et. al.*, Design of Low-Error Fixed-Width Modified Booth Multiplier, IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 12, no. 5, pp. 522-530, May 2004.

[11] Ravindra p Rajput, M. N Shanmukha Swamy, High speed Modified Booth Encoder multiplier for signed and unsigned numbers, 14th IEEE International Conference on Modelling and Simulation, 2012, pp. 649-654.

[12] Gary W. Bewick, Fast Multiplication Algorithms and Implementation, Ph.D. dissertation, Stanford University, Feb. 1994.

[13] Shiann-Rong Kuang, Jiun-Ping Wang, Cang-Yuan Guo, Modified Booth Multipliers With a Regular Partial Product Array, IEEE Trans. On Circuits and Systems-II: Express Briefs, vol. 56, no. 5, pp. 404-408, May 2009.

[14] R. Fried, Minimizing Energy Dissipation in High-Speed Multipliers, Proc. International Symp. Low Power Electronics and Design, 1997, pp. 214-219.

[15] Kiat-Seng Yeo and Kaushik Roy, Low-Voltage, Low-Power VLSI Subsystems, Tata McGraw-Hill Edition, 2009, ISBN: 0070677506, pp. 135-139.