

# DESIGN AND IMPLEMENTATION OF 2'S COMPLEMENT $N \times N$ MULTIPLIERS USING RADIX-4 MODIFIED BOOTH ENCODING IN FPGA

<sup>1</sup>Dongara Sandeepreddy, PG Scholar in VLSI,

<sup>2</sup>Burra Ayyappa Swamy, M.Tech, Asst. Professor, ECE Department,

<sup>1</sup>[sandeepreddy323@gmail.com](mailto:sandeepreddy323@gmail.com),

<sup>2</sup>[baswamy.mits@gmail.com](mailto:baswamy.mits@gmail.com),

<sup>1,2</sup>Madhira Institute of Technology & Sciences, Kodad, Nalgonda.

## ABSTRACT:

This paper presents a method to speed up Booth encoded multipliers by cing the size of Partial product array using Radix -4 Modified Booth encoded multiplier. This method is used for higher radices encoding for any size of mxn multiplications This reduction may allow for a faster compression of the partial product array and regular layouts. This technique is of particular interest in all multiplier designs, but especially in short bit-width two's complement multipliers for high-performance embedded cores. With the extra hardware of a (short) 3-bit addition, and the simpler generation of the first partial product row, we have been able to achieve a delay for the proposed scheme within the bound of the delay of a standard partial product row generation. We evaluated the proposed approach by comparison with some other possible solutions; the results based on a rough theoretical analysis and on logic synthesis showed its efficiency in terms of both area and delay.

*Keywords*— Multiplication, Radix-4, Modified Booth Encoding, partial product array.

## I. INTRODUCTION

In signal processing applications performance strongly depends on the effectiveness of the hardware used for computing multiplications. The high interest in this field is witnessed by the large amount of algorithm and implementations of the multiplication operations. In this short bit width (8-16 bits) two's complement multipliers with single-cycle throughput and latency have emerged to be important building blocks for high performance embedded processors and DSP execution cores. Applications for short bit-width multipliers are the design of SIMD units supporting different data formats. The basic algorithm for multiplication is based three main phases: 1) partial product (PP) generation, 2) PP reduction, and 3) final (carry propagated) addition. During PP generation, a set of rows is generated where each one is the result of the product of one bit of the multiplier by the multiplicand.

Modified Booth Encoding (MBE) is a technique that has been introduced to reduce the number of PP rows, still keeping the generation process of each row both simple and fast enough. One of the most commonly used schemes is radix-4 MBE, for a number of reasons, the most important being that it allows for the reduction of the size of the partial product array by almost half, and it is very simple to generate the multiples of

the multiplicand. More specifically, the classic two's complement  $n \times n$  bit multiplier using the radix-4 MBE scheme, generates a PP array with a maximum height of  $\lfloor n/2 \rfloor + 1$  rows, each row before the last one being one of the following possible values: all zeros,  $\pm X$ ;  $\pm 2X$ .

The PP reduction is the process of adding all PP rows by using a compression tree. Since the knowledge of intermediate addition values is not important, the outcome of this phase is a result represented in redundant carry save form i.e., as two rows, which allows for much faster implementations. The final addition has the task to sum these two rows and to present the final result in non redundant form i.e., as a single row.

Our aim is to produce a PP array of maximum height of  $\lfloor n/2 \rfloor$  to be then reduced by the compressor tree stage. This is the common case of values  $n$  which are power of 2, can lead to  $n$  implementation where the delay of the compressor tree is reduced by one XOR2 gate.

## II. SYSTEM DESIGN

### Block Diagram

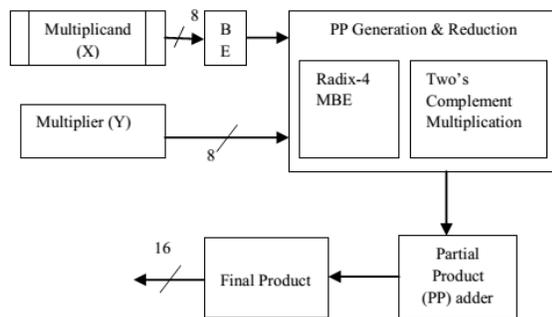


Fig.1. Reducing the computation time in

Two's complement multiplier. The multiplication can be realized by the shift-add algorithm by generating partial products. Thus, multiplication is proportional to the number of partial products to be added. High-radix

multiplication algorithms can reduce the number of partial products.

There are three major steps to any multiplication.

- The partial products (PP) are generated.
- The partial products are reduced to one row of final sums and one row of carriers.
- The final sums and carriers are added to generate the result.

One of the most commonly used schemes is radix-4 MBE, which it allows for the reduction of the size of the partial product array by almost half, and it is very simple to generate the multiples of the multiplicand.

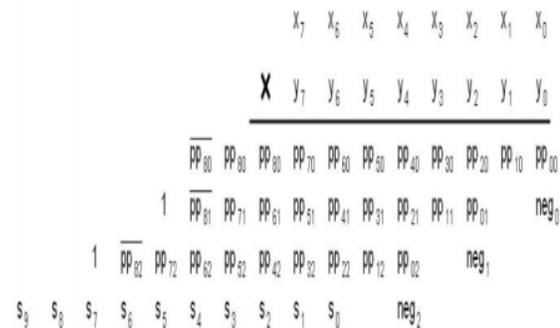


Fig.2. Partial product array by applying the two's complement computation method in the last row.

The approach is general and, for the sake of clarity, will be explained through the practical case of  $8 \times 8$  multiplication (as in the previous figures). As briefly outlined in the previous sections, the main goal of our approach is to produce a partial product array with a maximum height of  $\lfloor n/2 \rfloor$  rows, without introducing any additional delay. Let us consider, as the starting point, the form of the simplified array as reported in Fig. 2, for all the partial product rows except the first one. As depicted in Fig. 3a, the first row is temporarily considered as being split into two sub rows, the first one containing the partial



product generation.

The high-level description of our idea is as follows:

- Generation of the three most significant bit weights of the first row, plus addition of the last neg bit, possible implementations can use a replication of three times cascaded by the circuit of Fig. 5 to add the neg signal.
- Parallel generation of the other bits of the first row: possible implementations can use instances of the circuitry depicted in Fig. 6, for each bit of the first row, except for the three most significant;
- Parallel generation of the bits of the other rows: possible implementations can use the circuitry replicated for each bit of the other rows.

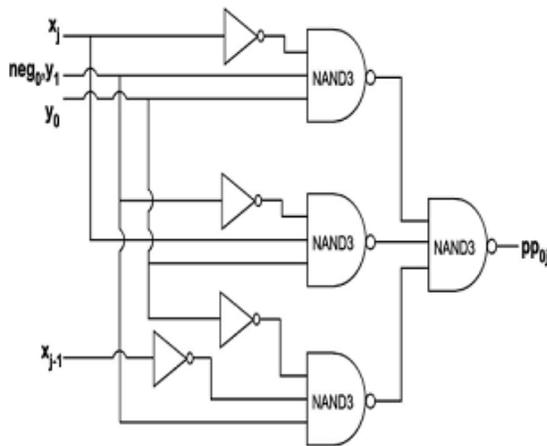


Fig. 6. Combined MBE signals and partial product generation for the first row (improved for speed)

Fig.6. Combined MBE signals and partial product generation for the first row (improved for speed) All items 1 to 3 are independent, and therefore can be executed in parallel. Clearly if, as assumed and expected, item 1 is not the bottleneck (i.e., the critical path), then the implementation of the proposed idea has reached the goal of not introducing time penalties.

### III. MODIFIED BOOTH ENCODING (RADIX-4)

Table 1: Modified Booth Encoding (Radix-4)

| $y_{2i+1}$ | $y_{2i}$ | $y_{2i-1}$ | Generated partial products |
|------------|----------|------------|----------------------------|
| 0          | 0        | 0          | $0 \times X$               |
| 0          | 0        | 1          | $1 \times X$               |
| 0          | 1        | 0          | $1 \times X$               |
| 0          | 1        | 1          | $2 \times X$               |
| 1          | 0        | 0          | $(-2) \times X$            |
| 1          | 0        | 1          | $(-1) \times X$            |
| 1          | 1        | 0          | $(-1) \times X$            |
| 1          | 1        | 1          | $0 \times X$               |

The method is to compute the product of a multiplicand  $X$  and a multiplier  $Y$ , is to produce the partial product array by generating one row for each bit of the multiplier  $Y$ . This methodology produces  $n$  rows, where  $n$  is the size of the multiplier. In general, a radix- $B = 2^b$  MBE leads to a reduction of the number of rows to about  $\lceil n/b \rceil$  while, on the other hand, it introduces the need to generate all the multiples of the multiplicand  $X$ , at least from  $-B/2 \times X$  to  $B/2 \times X$ . Radix-4 is easy to create the multiples of the multiplicand  $0$ ;  $\pm X$ ;  $\pm 2X$ .  $\pm 2X$  can be simply obtained by single left shifting of the corresponding terms  $\pm X$ .

Radix-4 MBE scheme consists of scanning the multiplier operand with a three-bit window and a stride of two bits. For each group of three bits ( $y_{2i+1}, y_{2i}, y_{2i-1}$ ), only one partial product row is generated according to the encoding in Table 1. For each partial product row, Fig. 1a produces the one, two, and neg signals. These signals are then exploited by the logic in Fig. 1b, along with the appropriate bits of the multiplicand, in order to generate the whole partial product array. The use of radix-4 MBE allows for the (theoretical) reduction of the PP rows to  $\lceil n/2 \rceil$ , with the possibility for each row to host a multiple of  $y_i \times X$ , with  $y_i \in \{0; \pm 1; \pm 2\}$

To generate the positive terms  $0, X$ , and  $2X$  at least through a left shift of  $X$ , some attention is required to generate the terms  $-X$  and  $-2X$  which, as observed in Table 1, can arise from three configurations of the  $y_{2i+1}, y_{2i}$ , and  $y_{2i-1}$  bits. To avoid computing negative encodings, i.e.,  $-X$  and  $-2X$ , the two's complement of the multiplicand is

generally used. The use of two's complement requires extension of the sign to the leftmost part of each partial product row, with the consequence of an extra area overhead. Thus, a number of strategies for preventing sign extension have been developed. For 2's complement it requires a neg signal to be added in the LSB position of each partial product row. For  $n \times n$  multiplier, only  $\lfloor n/2 \rfloor$  partial products are generated, the maximum height of the partial product array is  $\lfloor n/2 \rfloor + 1$ .

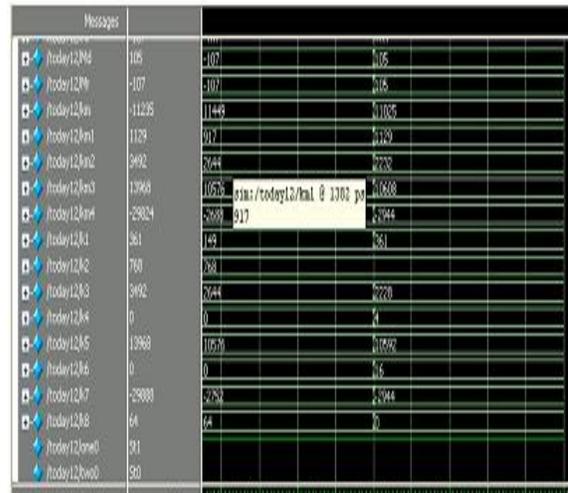
When 4-to-2 compressors are used the reduction of the extra row may require an additional delay of two XOR2 gates. By properly connecting partial product rows and using a Wallace reduction tree, the extra delay can be further reduced to one XOR2. However, the reduction still requires additional hardware, roughly a row of  $n$  half adders. This issue is of special interest when  $n$  is a power of 2, which is by far a very common case, and the multiplier's critical path has to fit within the clock period of a high performance processor. For instance, in the design presented in, for  $n = 16$  the maximum column height of the partial product array is 9, with an equivalent delay for the reduction of six XOR2 gates. For a maximum height of the partial product array of 8, the delay of the reduction tree would be reduced by one XOR2 gate. Alternatively, with a maximum height of 8, it would be possible to use 4 to 2 adders, with a delay of the reduction tree of six XOR2 gates, but with a very regular layout.

**IV. EXPERIMENTAL RESULTS**

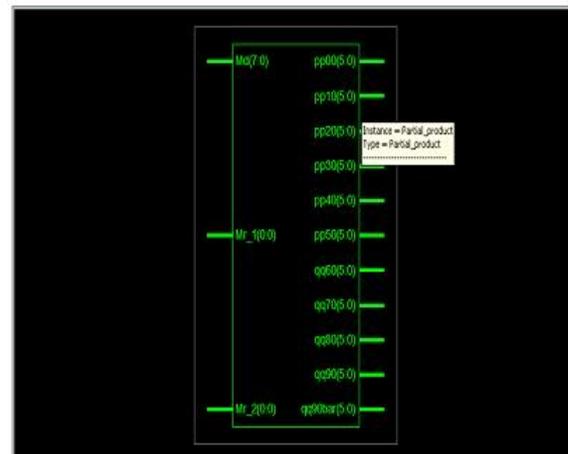
The proposed booth multiplier to perform multioperand multiplication has been simulated and the synthesis report can be obtained by using Xilinx ISE 12.1i. The various parameters which have been noted are shown in the table.

| Power Summary                      | Before reduction of PP row |       | After reduction of PP row |       |
|------------------------------------|----------------------------|-------|---------------------------|-------|
|                                    | I(mA)                      | P(mW) | I(mA)                     | P(mW) |
| Total estimated power consumptions | -                          | 201   | -                         | 133   |
| Vccint 1.80V                       | 108                        | 195   | 70                        | 126   |
| Vcco33 3.30V                       | 2                          | 7     | 2                         | 7     |

Simulation waveform:



Synathesis:



RTL Schematic

#### IV. CONCLUSION

Two's complement  $n \times n$  multipliers using radix-4 Modified Booth Encoding produce  $\lceil n/2 \rceil$  partial products but due to the sign handling, the partial product array has a maximum height of  $\lceil n/2 \rceil + 1$ . We presented a scheme that produces a partial product array with a maximum height of  $\lceil n/2 \rceil$ , without introducing any extra delay in the partial product generation stage. With the extra hardware of a (short) 3-bit addition, and the simpler generation of the first partial product row, we have been able to achieve a delay for the proposed scheme within the bound of the delay of a standard partial product row generation.

#### 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] O. 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  $54 \times 54$ -bit 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.