Low Power High Speed Two's Complement Multiplier

P.Arulbalaji[#1], Mrs. K.Vanitha[*2]
[#1]PG Student, Department of ECE, K.S.Rangasamy College of Technology, T.N, INDIA.
[*2]Assistant Professor, Department of ECE, K.S.Rangasamy College of Technology, T.N, INDIA.

Abstract - To reduce the area of partial product array size and improve the speed which is generated by a radix-4 Modified Booth Encoded Multiplier is used. This reduction is possible without any increase in the delay of the partial product generation stage. This reduction provides faster compression of the partial product array and regular layouts in two's complement multiplier. The proposed method is that the Radix-4 (Fixed-Width) Modified Booth Multipliers are used to achieve the low power and increase the speed by modifying the partial product matrix size. The Multiplier design implemented using Xilinx. The results based on a rough theoretical analysis and on logic synthesis showed its efficiency in terms of both area and delay. It is compared with Radix-4 (short bit-width) Modified booth encoded Multiplier.

Index terms - Multiplication, Modified Booth Encoding, partial product array, Fixed-width Modified Booth multiplier.

I. INTRODUCTION

The speed of multiplication operation is of great importance in digital signal processing as well as in the general purpose processors today. In the past multiplication was generally implemented via a sequence of addition, subtraction, and shift operations. Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of addition generates a partial product. In most computers, the operand usually contains the same number of bits.

For example, if we consider the multiplication X×Y with both X and Y on n bits and of the form $x_{n-1}...x_0$ and $y_{n-1}...y_0$, then the *i*th row is, in general, a proper left shifting of $y_i \times X$, i.e., either a string of all zeros when $y_i = 0$, or the multiplicand X itself when $y_i = 1$. In this case, the number of PP rows generated during the first phase is clearly *n*.

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 [n/2] + 1 rows, each row before the last one being one of the following possible values: all zeros, ±X, ±2X. The last row, which is due to the negative encoding, can be kept very simple by using specific techniques integrating two's complement and sign extension prevention.

II. SYSTEM DESIGN

Block Diagram

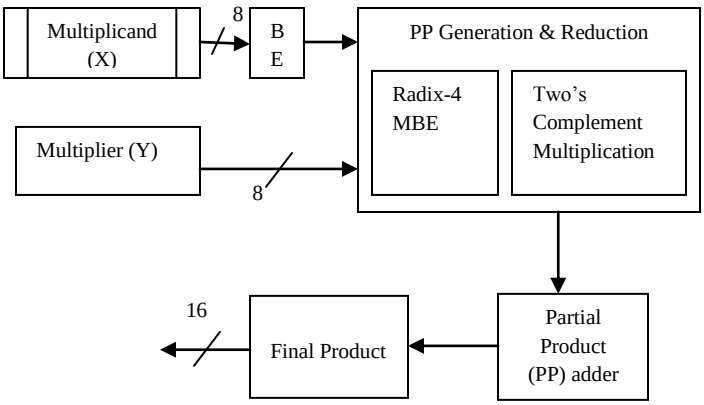


Fig.1. Reducing the computation time in two's complement multiplier.

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.

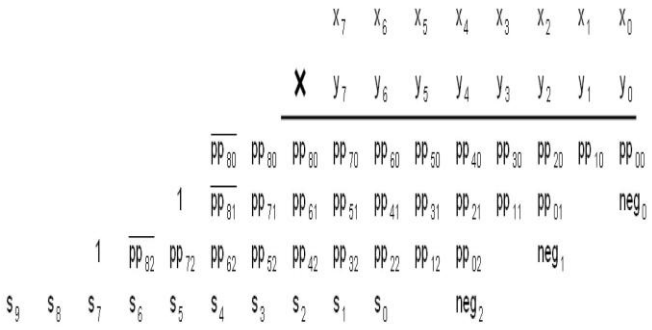


Fig.2. Partial product array by applying the two's complement computation method in the last row.

The approach is general and, for the sake of clarity, will be explained through the practical case of 8 × 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 $[n/2]$ 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 subrows, the first one containing the partial product bits (from right to left) from pp00 to pp80 and the second one with two bits set at "one" in positions 9 and 8. Then, the bit neg3 related to the fourth partial product row, is moved to become a part of the second subrow. The key point of this "graphical" transformation is that the second subrow containing also the bit neg3, can now be easily added to the first subrow, with a constant short carry propagation of three positions (further denoted as "3-bits addition"), a value which is easily shown to be general, i.e., independent of the length of the operands, for square multipliers. In fact, with reference to the notation of Fig. 5, we have that qq_{90} qq_{90} qq_{80} qq_{70} qq_{60} = 0 0 pp_{80} pp_{70} pp_{60} + 0 1 1 0 *neg*3.

As introduced above, due to the particular value of the second operand, i.e., 0 1 1 0 neg3, we have observed that it requires a carry propagation only across the least-significant three positions, a fact that can also be seen by the implementation shown in Fig. 4. It is worth observing that, in order not to have delay penalizations, it is necessary that the generation of the other rows is done in parallel with the generation of the first row cascaded by the computation of the bits qq_{90} qq_{90} qq_{80} qq_{70} qq_{60} in Fig. 3b.

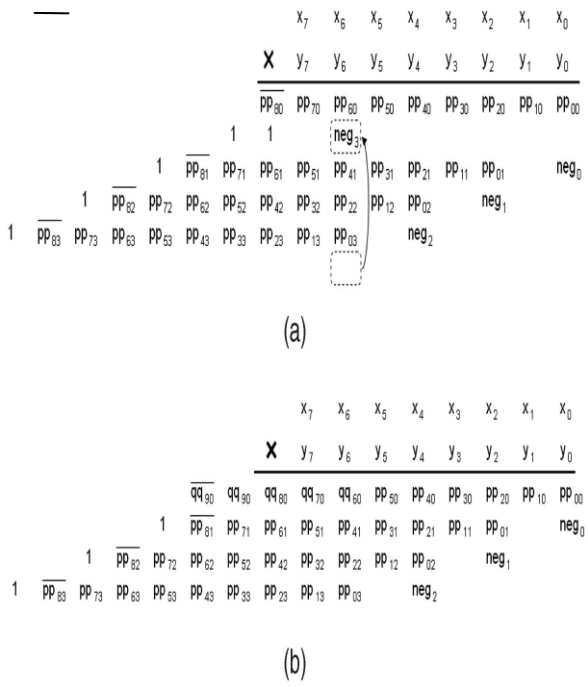


Fig.3. Partial product array after adding the last neg bit to the first row. (a)Basic idea (b) Resulting array

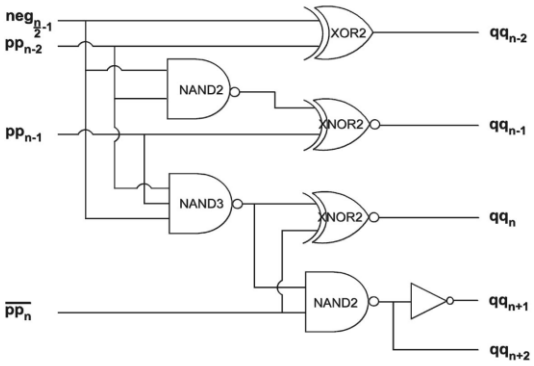


Fig.4. Gate-level diagram of the proposed method for adding the last *neg* bit in the first row

The generation of the MBE signals for the first row is simpler, and theoretically allows for the saving of the delay of one NAND3 gate. In addition, the implementation in Fig.6 has a delay that is smaller than the two parts of Fig. 5, although it could require a small amount of additional area. As we see in the

following, this issue hardly has any significant impact on the overall design, since this extra hardware is used only for the three most significant bits of the first row, and not for all the other bits of the array.

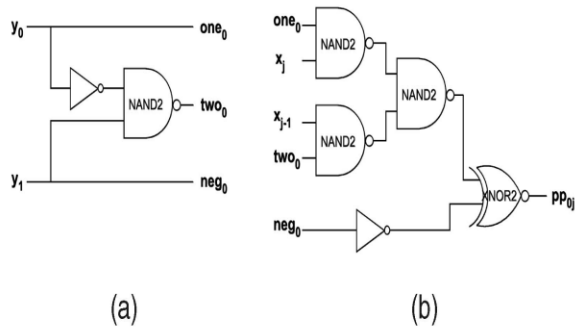


Fig.5. Gate-level diagram for first row partial product generation. (a) MBE signals generation. (b) Partial product generation.

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.

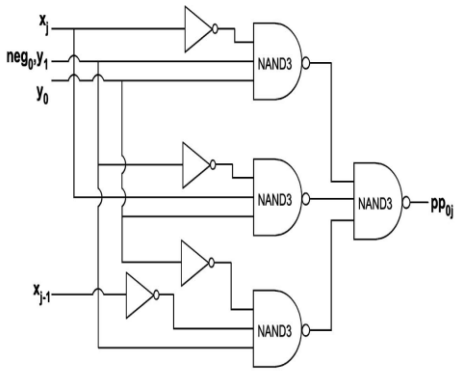


Fig.6. Combined MBE signals and partial product generation for the first row (improved for speed)

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. FIXED-WIDTH MODIFIED BOOTH MULTIPLIER

Modified Booth encoding is popular for reducing the number of partial products. The 2L -bit product P can be expressed in two's complement representation as follows:

$$P = X \times Y$$

Booth encoding, where L is an even number [4]. Taking 8 × 8 Booth multiplier as an example, due to the two's complement computation, n_i is equal to 1 when is y'_i negative; otherwise, n_i is equal to 0. The fixed-width Booth multiplier design [2], some of products are truncated by using a rounding operator to hold the data length fixed in L-bit. Therefore, an extra one binary bit 1 added into the most significant column of truncation part (TP) in Fig. 2, which indicates the rounding off operation of the P-T Booth multipliers.

IV. PROPOSED METHOD

Block Diagram

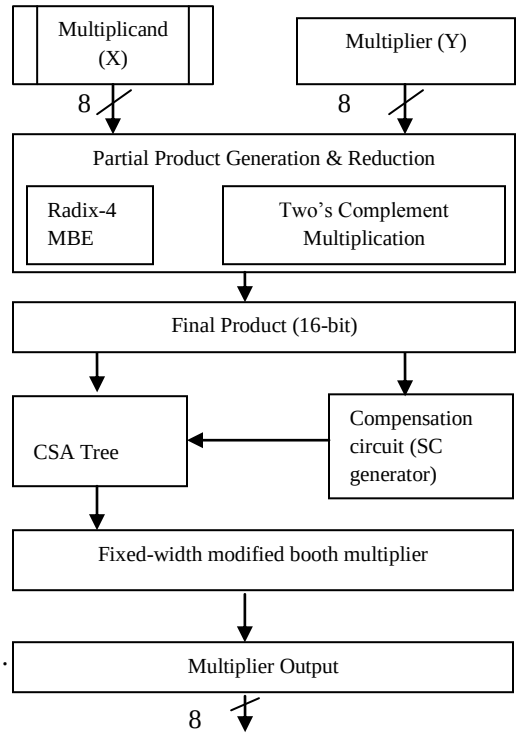


Fig.7. Low Power High Speed Two's Complement Multiplier.

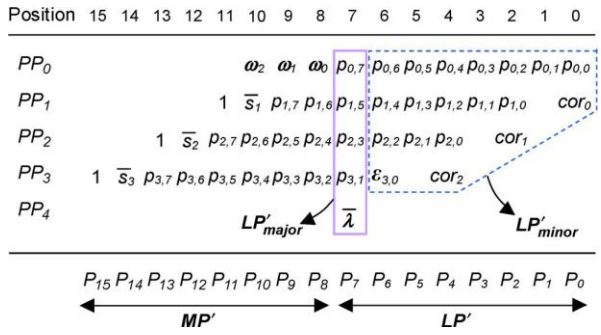


Fig. 8. The proposed 8 × 8 modified Booth partial product matrix

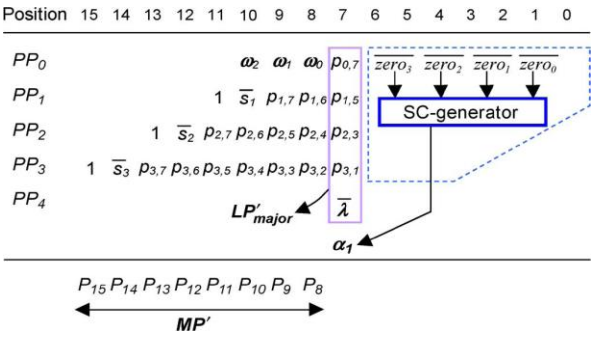


Fig. 9. Final partial product matrix of proposed fixed-width modified Booth multiplier for n=8.

The values of partial product bits are heavily dependent on the outputs of Booth encoders; we first explore the relation between the outputs of Booth encoders and the carry value propagated from LP_{minor} to LP_{major}. Next, an effective and simple error compensation function, which takes the outputs of Booth encoders as inputs and then generates the approximate carry value, is derived to reduce the truncation error and make the error distribution as symmetric and centralized as possible [7]. Finally, an uncomplicated and fast compensation circuit is constructed to form a high-accuracy fixed-width multiplier.

The Proposed 8×8 Modified booth partial product matrix is shown in fig.10 and it's used to decrease the partial product bits in so that the carry value generated by can be estimated more accurately and easily.

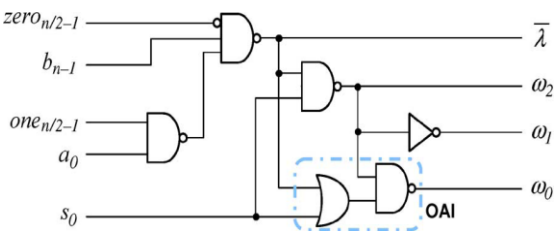


Fig. 10. The circuit to produce λ and ω_2 ω_1 ω_0.

V. PROPOSED LOW ERROR COMPENSATION CIRCUIT

SC Generator (Signal Conditioning Generator)

The signal is generated as a result of truncation process in partial product array. This output signal will be added with MSB bits of the partial product array [6].

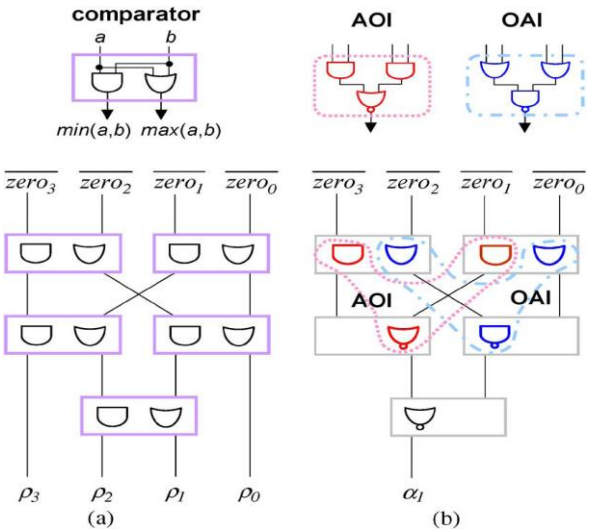


Fig. 11. (a) Odd-even merge sorting network for n = 8. (b) Proposed SC-generator for n = 8.

VI. EXPRIMENTAL 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.

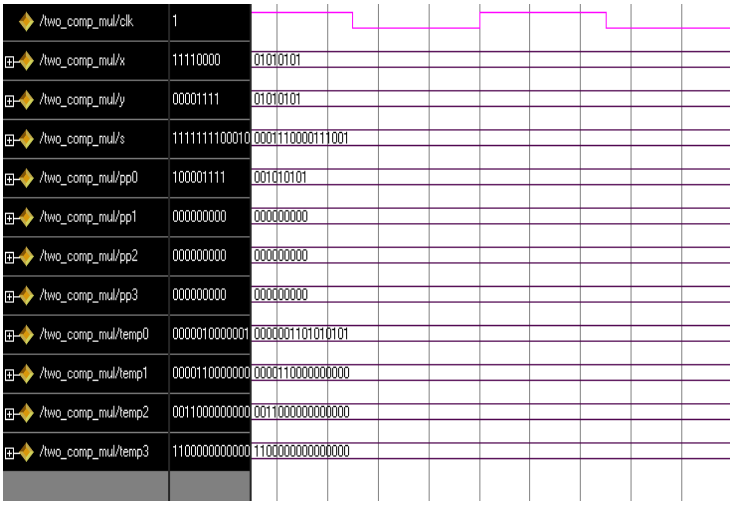


Fig.12 Simulation output for after reduction of pp row.

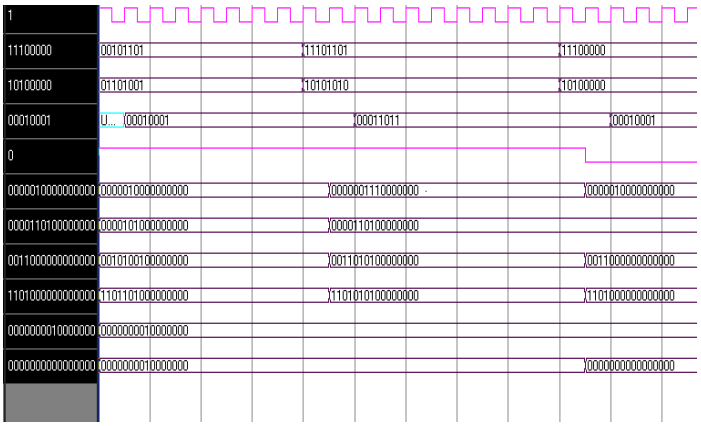


Fig.13. Simulation Output for SC Generator

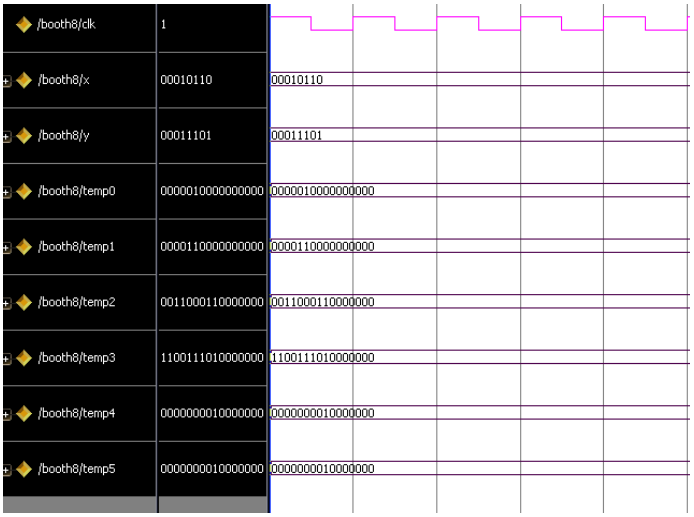


Fig.14. Simulation Output for Truncation Process

TABLE 1
OUTPUT FOR POWER CONSUMPTION

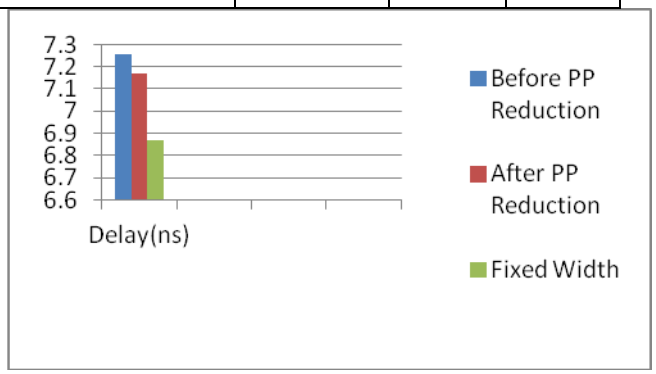
Power Summary	Before reduction of PP row		After reduction of PP row	
	I(mA)	P(mW)	I(mA)	P(mW)
Total estimated power consumptions	-	201	-	133
Vccint 1.80V	108	195	70	126
Vcco33 3.30V	2	7	2	7

TABLE 2
OUTPUT FOR SPEED

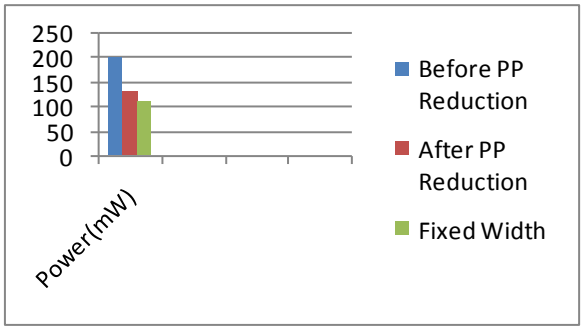
Timing Summary Speed grade:-7	Before reduction of PP row		After reduction of PP row	
	ns	*MHz*	*ns*	*MHz*
Min. Period	7.257	-	7.167	-
Max. Frequency	-	137.7	-	139.5
Min. input arrival time before clock	9.534	-	5.802	-
Min. output required time after clock	7.247	-	6.140	-
Max. combinational path delay	No	-	No	-

TABLE 3
COMPARISON FOR EXISTING AND PROPOSED METHOD RESULTS FOR DELAY

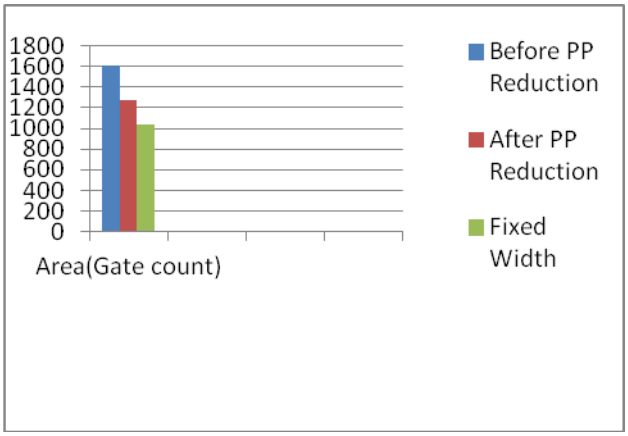
Multiplier Types	Power(mW)	Delay (ns)	Area (Gate count)
Before PP Reduction (Conventional Multiplier)	201	7.257	1613
After PP Reduction (Short bit-width)	133	7.167	1277
Fixed-width MBE	110	6.870	1033



RESULTS FOR POWER



RESULTS FOR AREA



VII. CONCLUSION

Two's complement $n \times n$ multipliers using radix-4 Modified Booth Encoding produce $[n/2]$ partial products but due to the sign handling, the partial product array has a maximum height of $[n/2] + 1$. A scheme that produces a partial product array with a maximum height of $[n/2]$ is presented, without introducing any extra delay in the partial product generation stage. With the extra hardware of a (short bit-width) 3-bit addition, and the simpler generation of the first partial product row, to achieve a delay for the proposed scheme within the bound of the delay of a standard partial product row generation. The outcome of the above is that the reduction of the maximum height of the partial product array by one unit may simplify the partial product reduction tree, both in terms of delay and regularity of the layout. Radix-4 (Fixed-Width) Modified Booth Multipliers height and area of the partial product is reduced. Hence the power consumption is also less. This is clearly depicted in our results. This speeds up the calculation and makes the system faster.

REFERENCES

[1] Fabrizio Lambert and Nikos Andrikos, "Reducing the Computation Time in (Short Bit – Width) Two's Complement Multipliers", *IEEE Trans. on Computers*, vol. 60, no. 2, Feb.2011.

[2] Yuan-Ho Chen, Student Member, IEEE, and Tsin-Yuan Chang, Member, IEEE, "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, march 2012.

[3] Tso-Bing Juang, Student Member, IEEE, and Shen-Fu Hsiao, Member, IEEE, "Low-Error Carry-Free Fixed-Width Multipliers With Low-Cost Compensation Circuits," *IEEE Transactions on circuits and systems—II: Express Briefs*, Vol. 52, No. 6, June 2005.

[4] J.Y. Kang and J.L. Gaudiot, "A Simple High-Speed Multiplier Design", *IEEE Trans. on Computers*, vol. 55, no. 10, pp. 1253-1258, Oct.2006.

[5] 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.

[6] F. Lamberti, N. Andrics, E.Antelo, and P.Mountuchi, "Speeding – Up Booth Encoded Multipliers by Reducing the size of Partial Product Array", internal report, http://arith.polito.it/it_mbe.pdf pp. 1-14, 2009.

[7] K. J. Cho, K. C. Lee, J. G. Chung, and K. K. Parhi, "Design of low-error fixed-width modified Booth multiplier," *IEEE Trans. Very Large Scale Integration. (VLSI) Syst.*, vol. 12, no. 5, pp. 522–531, May 2004.

[8] XLINX Synthesis and Simulation Design Guide. http://www.xilinx.com/itp/xilinx10/books/docs/sim/sim.pdf

AUTHORS BIOGRAPHIES

P. Arulbalaji received B.E. degree in Electrical and Electronics Engineering from PGP College of Engineering and Technology at Namakkal, Affiliated to Anna University - Chennai in 2007. M.E degree in Applied Electronics from K.S.Rangasamy College of Technology at Tiruchengode, Affiliated to Anna University – Chennai in 2013. He is having 4 years of experience in the field of Operation and Maintenance in Electrical panel boards Industry, presently pursuing final year M.E degree in Applied Electronics. He is a member of the IEEE. He is Interested in area of the Low Power VLSI.

Mrs. K.Vanitha received B.E. degree in Electronics & Communication Engineering from PET Engineering College at Valliyur, Affiliated to Manonmaniyam Sundaranar University - Tirunelveli in 2003. M.E degree in VLSI Design from K.S.Rangasamy College of Technology at Tiruchengode, Affiliated to Anna University - Coimbatore in 2010. She is working as an Assistant Professor in Dept. of ECE in K.S.Rangasamy College of Technology, Tiruchengode. She is Interested in area of the Low Power VLSI. She is a member of the IEEE.