A New Architecture for Signed and Unsigned Multiplier by using Radix-4 Process

Supriya Sarkar, Sanghita Deb, G. Dilip

Abstract--This paper presents the design and implementation of signed-unsigned Modified Booth Encoding (SUMBE) multiplier. The present Modified Booth Encoding (MBE) multiplier and the Baugh-Wooley multiplier perform multiplication operation on signed numbers only. 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 SUMBE multiplier. The modified Booth Encoder circuit generates half the partial products in parallel. By extending sign bit of the operands and generating an additional partial product the SUMBE multiplier is obtained. The Carry Save Adderr (CSA) tree and the final Carry Lookahead (CLA) adder used 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.

IndexTerms-SUMBE, MBE, CSA, CLA. Baugh-wooley multiplier.

I. INTRODUCTION

In digital computing systems multiplication is an arithmetic operation. The multiplication operation consists of producing partial products and then adding these partial products the final product is obtained. Thus the speed of the multiplier depends on the number of partial product and the speed of the adder. Since the multipliers have a significant impact on the performance of the entire system, many high-performance algorithms and architectures have been proposed [1-23]. 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 papers [1, 4] presents a design methodology for high-speed Booth encoded parallel multiplier. For partial product generation, a new Modified Booth encoding (MBE) scheme is used to improve the performance of traditional MBE schemes. But this multiplier is only for signed number multiplication operation.

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 [2, 3] presents a simple approach to generate a regular partial product array with fewer partial product rows and negligible overhead, there by 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 function 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 [5, 8, 11] 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 [12] and Braun array multipliers [13] 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 SUMBE multiplier.

II. CONVENTIONAL MBE MULTIPLIERS

The new MBE recoder [1] 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. Fig. 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 yLSB 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, yMSB is combined with x- signals to determine the sign extension signals. Fig. 2 shows an overview of the partial product array for an 8 × 8 multiplier. The sign extension circuitry developed in [22] and [23]. The conventional MBE partial product array has two drawbacks: 1) an additional partial product term at the (n-2)th bit position; 2) poor performance at the LSB-part. To remedy the two drawbacks, the LSB part of the partial product array is modified. Referring to Fig. 2a, the Row_LSB (gray circle) and the Neg_ cin terms are combined and further simplified using Boolean minimization. The new equations for the Row_LSB and Neg_cin can be written as (1) and (2), respectively.

Manuscript received November 15, 2013.

Supriya Sarkar , M. Tech Student, Dept. of ECE, RKDF Institute of science and technology, Bhopal , India.

Sanghita Deb, B. E Student , Dept of CSE, N.H College of engineering, BAMU , Aurangabad ,M.S ,India.

G .Dilip, M. Tech, Dept. of ECE, Jayamukhi institute of technology and science, Warangal , India.

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

$$Row\_LSB_i = y_{LSB}(x_{2i-1} \oplus x_{2i})$$

$$Neg\_cin_i = x_{2i+1}\ \overline{(x_{2i-1} + x_{2i})}\ \overline{(x_{2i-1} + y_{LSB})}\ \overline{(x_{2i} + y_{LSB})}$$

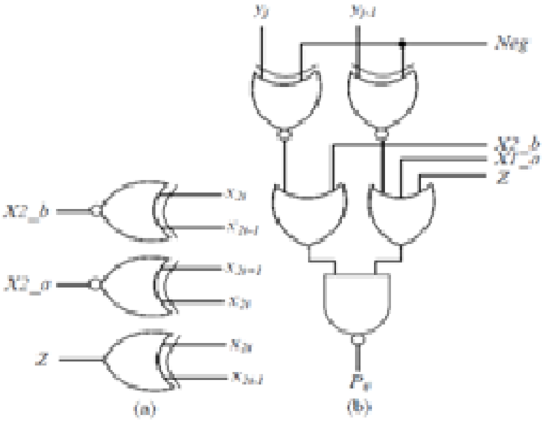
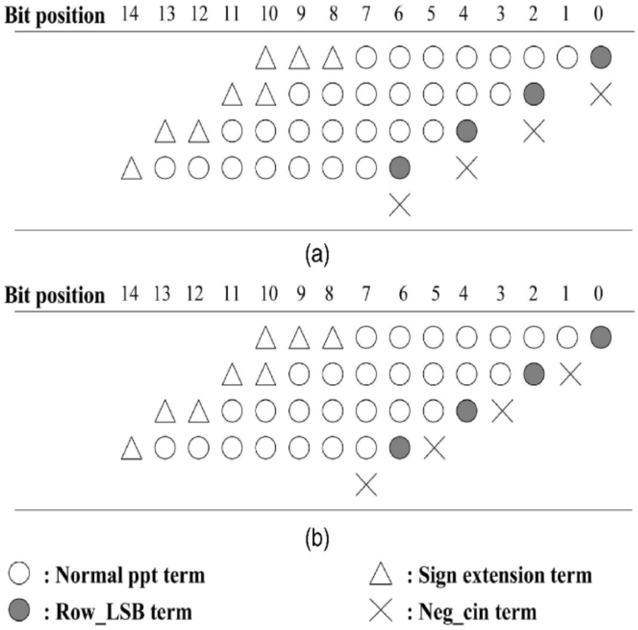


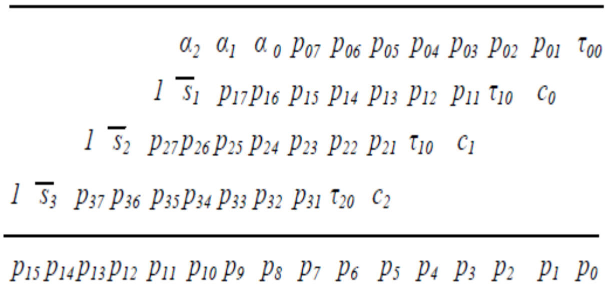
Fig. 1. The Encoder and Decoder for the new MBE scheme. (a) Simple encoder (b) Decoder.



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
△ : Sign extension term
● : Row_LSB term
✕ : Neg_cin term

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

The Fig. 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 Fig. 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 [2] produce a very regular partial product array, as shown in Fig. 3. Not only each negi 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.



α_2 α_1 α_0 p_{07} p_{06} p_{05} p_{04} p_{03} p_{02} p_{01} τ_{00}
1 $\overline{s_1}$ p_{17} p_{16} p_{15} p_{14} p_{13} p_{12} p_{11} τ_{10} c_0
1 $\overline{s_2}$ p_{27} p_{26} p_{25} p_{24} p_{23} p_{22} p_{21} τ_{10} c_1
1 $\overline{s_3}$ p_{37} p_{36} p_{35} p_{34} p_{33} p_{32} p_{31} τ_{20} c_2
p_{15} p_{14} p_{13} p_{12} p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0

Fig.3.The partial product array for 8×8 multiplier.

III. PROPOSED SUMBE MULTIPLIER

The main goal of this paper is to design and implement 8×8 multiplier for signed and unsigned numbers using MBE technique. Table 2 shows the truth table of MBE scheme. From table 2 the MBE logic diagram is implemented as shown in Fig. 4. Using the MBE logic and considering other conditions the Boolean expression for one bit partial product

TABLE 2: 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	0

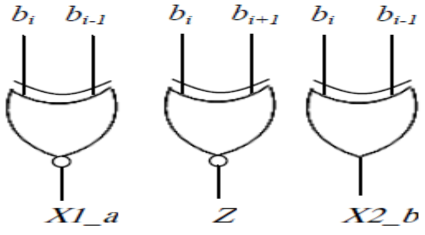


Fig. 4. Logic diagram of MBE.

Equation 3 is implemented as shown in Fig. 5. The SUMBE multiplier does not separately consider the encoder and the decoder logic, but instead implemented as a single unit called partial product generator as shown in Fig. 5. The negative partial products are converted into 2's complement by adding a negate (Ni) bit. An expression for negate bit is given by the Boolean equation 4. This equation is implemented as shown in Fig. 6. The required signed extension to convert 2's complement signed multiplier into both signed-unsigned multiplier is given by the equations 5and 6. For Boolean equations 5 and 6 the corresponding logic diagram is shown in Fig.7

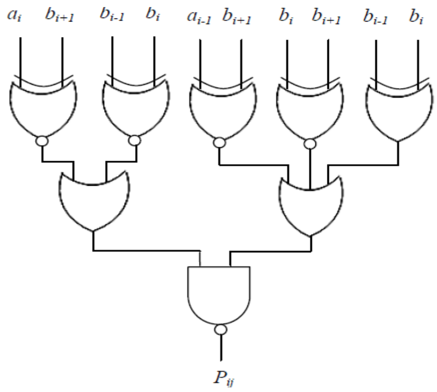


Fig. 5. Logic diagram of 1-bit partial product generator

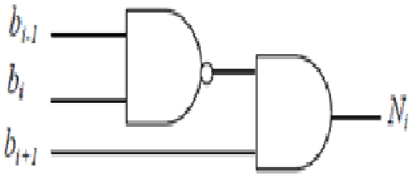


Fig. 6. Logic diagram of negate bit generater.

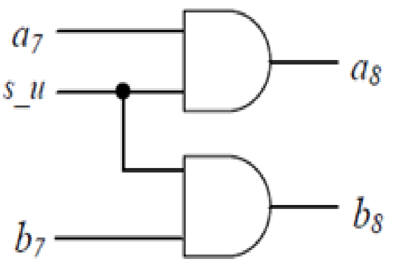
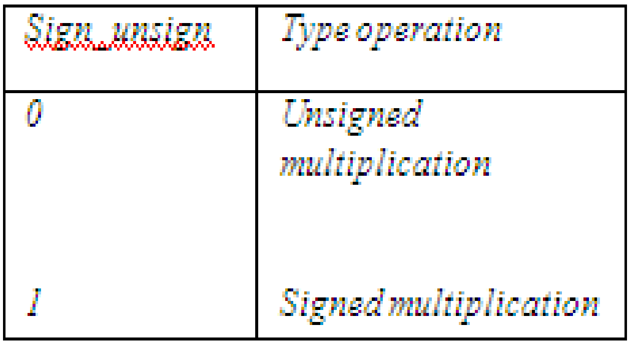


Fig. 7. Logic diagram of sign converter

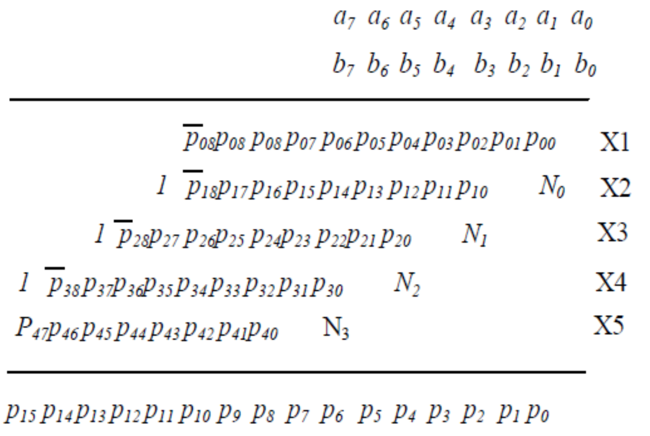
The working principle of sign extension that converts signed multiplier signed -unsigned multiplier as follows. One bit control signal called signed-unsigned (s_u) bit is used to indicate whether the multiplication operation is signednumber or unsigned number. When Sign-unsign s_u) = 0, itindicates unsigned number multiplication, and when s_u = 1,it indicates signed number ultiplication. It is required thatwhen the operation is unsigned ultiplication the signextended bit of both ultiplicand and multiplier should be extended with 0, that is a8 = a9 = b8 = b9 = 0. It is required that when the operation is signed multiplication the sign extended bit depends on whether the multiplicand is negative or the multiplier is negative or both the operands are negative. For this when the multiplicand operand is negative and muitplier operand is positive the sign extended bits should be generated are s_u = 1, a7 =1, b7 = 0, a8 = a9 =1, and b8 = b9 =0. And when the multiplicand operand is positive and muitplier operand is negative the sign extended bits should be generated are s_u = 1, a7 =0, , b7 = 1, a8 = a9=0, and b8 = b9 =1. Table 3 shows the SUMBE multiplier operation.



Sign_unsign	Type operation
0	Unsigned multiplication
1	Signed multiplication

Table 3: SUMBE operation

Fig. 8 shows the partial products generated by partaial product generater circuit which is shown in Fig. 5. There are 5-partial products with sign extension and negate bit Ni. All the 5-partial products are generated in parallel.



$a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$

$b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$

$\overline{p_{08}}p_{08}\,p_{08}\,p_{07}\,p_{06}p_{05}p_{04}p_{03}p_{02}p_{01}p_{00}$ X1

$1\ \overline{p_{18}}p_{17}p_{16}p_{15}p_{14}p_{13}\,p_{12}p_{11}p_{10}$ N_0 X2

$1\ \overline{p_{28}}p_{27}\,p_{26}p_{25}\,p_{24}p_{23}\,p_{22}p_{21}p_{20}$ N_1 X3

$1\ \overline{p_{38}}p_{37}p_{36}p_{35}p_{34}p_{33}p_{32}p_{31}p_{30}$ N_2 X4

$P_{47}p_{46}p_{45}\,p_{44}p_{43}p_{42}p_{41}p_{40}$ N_3 X5

$p_{15}\,p_{14}p_{13}p_{12}p_{11}\,p_{10}\,p_9\ p_8\ p_7\ p_6\ p_5\ p_4\ p_3\ p_2\ p_1\,p_0$

Fig. 8. 8×8 multiplier for signed-unsigned number.

In Fig. 8 there are 5-partial products namely X1, X2, X3, X4 and X5. These partial products are added by the Carry Save Adders (CSA) and the final stage is Carry Lookahead (CLA) adder as shown in Fig. 9. Each CSA adder takes three inputs and produse sum and carry in parallel. There are three CSAs, five partial products are added by the CSA tree and finally when there are only two outputs left out then finally CLA adder is used to produce the final result. Assuming each gate delay an unit delay, including partial product generator circuit delay, then the total through the CSA and CLA is 3+4 = 7 Unit delay. Thus with present Very Large Scale Integration (VLSI) the total delay is estimated around 0.7

nanosecond and the multiplier operates at giga hertz frequency.

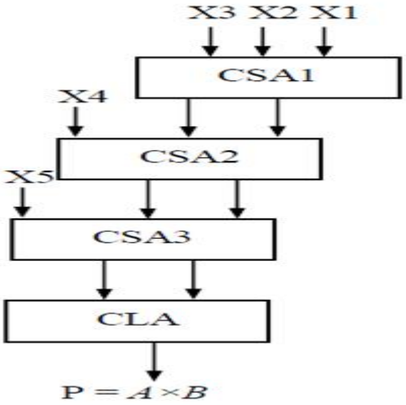


Fig. 9. Partial product adder logic.

IV. RESULTS

Verilog code is written to generate the required hardware and to produce the partial product, for CSA adder, and CLA adder. After the successful compilation the RTL view generated is shown in Fig.10.

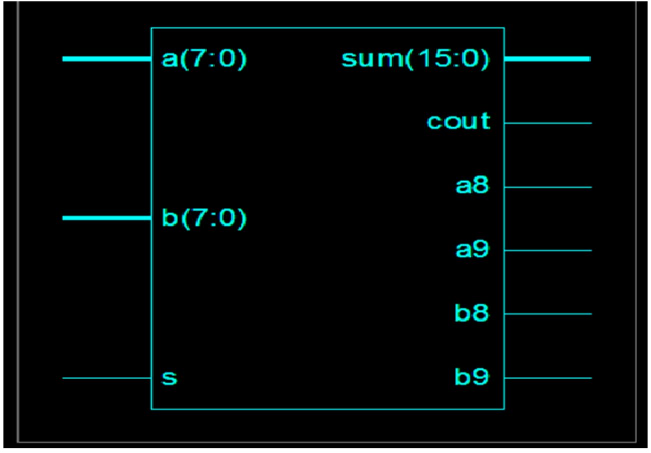
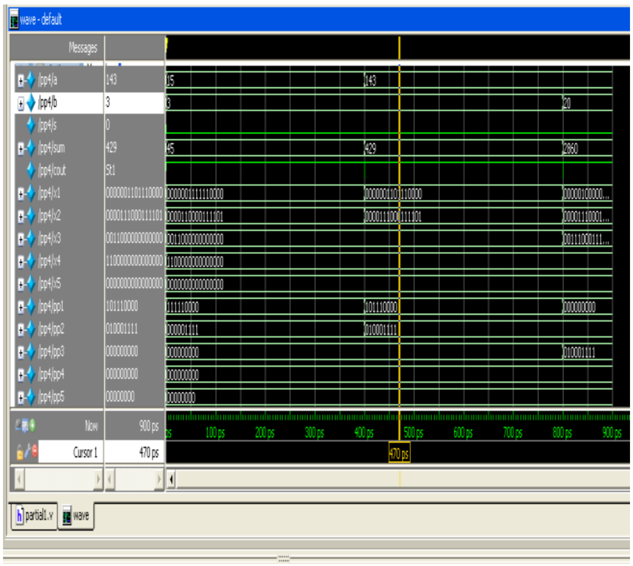


Fig. 10: RTL view of 8×8 signed-unsigned multiplier.


wave - default

Messages	
/pp4/a	143
/pp4/b	3
/pp4/s	0
/pp4/sum	429
/pp4/cout	St1
/pp4/x1	0000001101110000
/pp4/x2	0000111000111101
/pp4/x3	0011000000000000
/pp4/x4	1100000000000000
/pp4/x5	0000000000000000
/pp4/pp1	101110000
/pp4/pp2	010001111
/pp4/pp3	000000000
/pp4/pp4	000000000
/pp4/pp5	00000000
Now	900 ps
Cursor 1	470 ps

Fig11:simulation for 8x8 multiplier.

V. CONCLUSION

In all multiplication operation product is obtained by adding partial products. Thus the final speed of the multiplier circuit depends on the speed of the adder circuit and the number of partial products generated. If radix 8 Booth encoding technique is used then there are only 3 partial products and for that only one CSA and a CLA is required to produce the final product.

REFERENCES

1. verilog hdl by padbanabam.
2. Shiann-Rong Kuang, Jiun-Ping Wang, and Cang-Yuan Guo, “Modified Booth multipliers with a Regular Partial Product Array,” IEEE Transactions on circuits and systems-II, vol 56, No 5, May 2009.
3. Li-Rong Wang, Shyh-Jye Jou and Chung-Len Lee, “A well-tructured Modified Booth Multiplier Design” 978-1-4244-1617-2/08/$25.00 ©2008 IEEE.
4. Soojin Kim and Kyeongsoon Cho “Design of High-speed Modified Booth Multipliers Operating at GHz Ranges” World Academy of Science, Engineering and Technology 61 2010.
5. Magnus Sjalander and Per Larson-Edefors. “The Case for HPMBasedBaughWooleyMultipliers,”ChalmersUniversityofechnology, Sweden, March 2008.
6. Z Haung and M D Ercegovac, “High erformance Low Power left to .right array multiplier design” IEEE trans.Computer, vol 54 no3, page 272-283 Mar 2005.
7. Hsing-Chung Liang and Pao-Hsin Huang, “Testing Transition Delay Faults in Modified BoothMultipliers by Using C-testable and SIC Patterns”IEEE2007, 1-4244-1272-2/07.
8. Aswathy Sudhakar, and D. Gokila, “Run-Time Reconfigurable Pipelined Modified Baugh-Wooley Multipliers,” Advances in Computational Sciences and Technology ISSN 0973-6107 Volume 3 Number 2 (2010) pp. 223–235.
9. Myoung-Cheol Shin, Se-Hyeon Kang, and In-Cheol Park, “An Area-Efficient Iterative Modified-Booth Multiplier Based on Self-Timed Clocking,” Industry, and Energy through the project System IC 2010, and by IC Design Education Center (IDEC).
10. Leandro Z. Pieper, Eduardo A. C. da Costa, Sérgio J. M. de Almeida, “Efficient Dedicated Multiplication Blocks for2´s Complement Radix-2m Array Multipliers,” JOURNAL OF COMPUTERS, VOL. 5, NO. 10, OCTOBER 2010.
11. C R Baugh and B. A Wooley, “ A two’s complement parallel array multiplication algorithm,” IEEE Transaction on Computers, Vol. 22, n0.12,pp 1045-1047, Dec.1973.
12. Neil H E Weste, David Harris, Ayan Banerjee, “CMOS VLSI Design A circuits and Systems Perspective ” Third edition, Pearson Education, pp.347-349.
13. Pucknell Douglas A, Eshraghan, Kamran, “Basic VLSI Design,”Third edition 2003, PHI Publication, pp.242-243.
14. A Chandrakasan and R Brodersen, “ Low power CMOS digital design,” IEEE J solid state circ, vol 27 no. 4, April 1992, pp. 473-484.
15. C. S Wallace, “A suggestion for a fast multiplier” IEEE Transaction on Electronic Computers, pp-14-17, Feb-1974.

AUTHORS PROFILE

Supriya Sarkar, was born in India, Tripura. He received the B.E degree from SRTMU , Nanded, Maharashtra, India, in 2006, he has six year teaching experince, presently pursuing his M.Tech from RGPV, Bhopal, India. His research interests include VLSI and embedded systems .

Sanghita Deb, was born in India, Tripura. she is a B.E. student of Dr Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra.

Dilip .G, was born in A.P,India. He received his M .tech from JNTU, Hyderabad. Presently he is working as a VLSI design engineer.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication