Area Optimized and Low Power using Modified Booth Multiplier for Unsigned Numbers

Rubi Choubey, Md. Arif

Abstract-Power consumption and small area is very important for fabricating DSP system and high performance system, requirement of present scenario computer system is dedicated for very high speed and low power unique multiplier unit for signed and unsigned number therefore in this paper focus on unsigned number by using modified booth multiplier. The unsigned 4 bit and 8 bit implementation done by some modification in booth multiplier modified booth multiplier come out to make efficient multiplier reduce N/2 partial product.

The parallel multiplier 4 bit and 8 bit modified booth multiplier does the computation using lesser adder and lesser iterative step. The implementation of unsigned 4 bit and 8 bit done in Xilinx ISE Design suite 12.1 tool by using VHDL, model Sim.

Index Terms- Array Multiplier Booth multiplier, Modified Booth Multiplier, Model Sim, Partial Product, Unsigned, VHDL, Xilinx.

I. INTRODUCTION

In present scenario digital computing system need fast multiplication, in digital computing systems multiplication is an arithmetic operation. The multiplication operation consists of producing partial products and then adding these partial products than final product is obtained. Multiplier is key component of many high performance system like DSP(Digital Signal Processing), FIR filter, FFT(Fast Fourier Transform), Microprocessor, etc. Earlier ALU's adders were used to perform the multiplication originally. There are some fast multiplier Array multiplier ,booth multiplier and modified booth multiplier. As the applications of Array multipliers were introduced the clock rates increased as well as timing constrains became austere [1]. Binary multiplication can done same as decimal number multiplication but it using long multiplication method, so it has more computational cost, and computation time is also increases. Therefore an efficient method of multiplier come out which where recursive adding of multiplicand and multiplier, there are many algorithm are used for multiplication, Booth Algorithm is more efficient than other multiplier. Booth multiplier reduces number of iterative steps compare to conventional steps[2], this algorithm can reduce the number of additions require to produce result. Advances in technology have allowed many researchers to implement multipliers which present both high-speed and regularity, so making them appropriate for VLSI implementation. In Modified Booth algorithm reduce number of partial product by half (N/2) so the less number of transistors which leads to reduced power consumption and less area, compared to conventional Booth structure. The booth's algorithm for multiplication can be modified to perform unsigned multiplication along with signed multiplication.Proposed modified booth multiplier is more efficient as compare to previous multiplier result of different parameter for unsigned 4 bits and 8 bits like look up table, input output block, number of slices, power and current because modular structure are used and coding done in VHDL in Xilinx 12.1 ISE design suite using VHDL(VHSIC Hardware Description Language) and result of both base and proposed comparison result show in table.

II. ARRAY MUTIPLIER

Array multiplier is similar as carry save array it is efficient layout of combinational multiplier with are number of full adder and half adder are used of two binary number multiplication and result of array multiplier it is long multiplication method and regular routing pattern, parallel partial generator[7].

Array of 4 bit multiplier unsigned number

			+S0[3]	s0[2]	s0[1]	s0[0]
		+S1[3]	s1[2]	s1[1]	s1[0]	0
	+S2[3]	s2[2]	s2[1]	s2[0]	0	0
+s3[3]	s3[2]	s3[1]	s3[0]	0	0	0
P[7]	p[6]	p[5]	p[4]	p[3]	p[2]	p[0]

Figure 1: Partial Product array for 4 bit unsigned number

III. BOOTH MULTIPLIER

In 1951, the Booth's Algorithm is a multiplication algorithm which was devised by Andrew Donald Booth this algorithm multiplies two signed binary numbers in two's complement notation[1]. It makes repeated addition of one two predetermined values A and S to a product P after which it performs a rightward arithmetic shift on P. Let m and e be the multiplicand and multiplier, respectively and let x and y represent.

- First the values of two predetermined A and S to obtain the product P. length of all these numbers should be equal(x+y+1)
  1. A : substitute m(binary) in MSB and append remaining bit (y+1)zeros.
  2. S : substitute -m(binary) in MSB and append remaining bit (y+1)zeros.
  3. P : Substitute x bit of zeros in MSB. Then right insert value of e and append value LSB bit for zeros.
- Now consider last two significant bit of P.
- Determine the two least significant (rightmost) bits of *P*.
  1. If they are 00, do nothing. Use *P*

Manuscript received on April 16, 2014.

Rubi Choubey, Completed B.E (Electronics and Communication). HCET, Jabalpur (M.P), India.

Mohammed Arif, has completed M.Tech in Embedded and VLSI Design from G.G.I.T.S, Jabalpur (M.P), India.

directly in the next step.

2. If they are 01, find the value of P + A. Ignore any over-flow.
3. If they are 10, find the value of P + S. Ignore any over flow.
4. If they are 11, do nothing. Use *P* directly in the next step.

- After than only arithmetic shift right and new result of P.
- Y times repeat the process.
- At last the final product of m and r we get.

Example:- If m and e are multiplicand and multiplier where m=2,e=3,than in binary m=0010,-m=1110,e=0011,x=4,y=4

A : 0010 0000 0

S : 1110 0000 0

P : 0000 0011 0

Apply 2nd and 3rd step at y times

P : 0000 0011 0 the LSB(List Significant Bit) is 10 then P= P+S = 1110 0011 0 after shift rightmost

P= 1111 0001 1 the LSB is11 then no change shifting operating is done

P= 1111 1000 1the LSB is 01 then P=P+A

P= 0001 1000 1 after shifting right most P=0000 0110 0 the LSB is 00 then on change shifting rightmost then the final output after four loop occur that is P= 0000 0110 0

P=2*3=6

The two draw backs come out in original version of booth algorithm.

1. So there are many variable inconvenient in design parallel multiplier because of number of add subtract and shift operation.
2. When there are isolated 1's the algorithm become inefficient.

These drawback overcome modified booth algorithm.

IV. EXISTING MODIFIED BOOTH MULTIPLIER

The modified booth multiplier was proposed by D. L. Macsorley in 1961. MBA (Modified Booth Algorithm) is one of the popular multiplication algorithm. It is high speed multiplier is to enhance parallelism which help is to reduce number of partial product(pp) row, by using modified booth algorithm overall the number of partial product are decrease from N to N/2 where N is multiplicand[6]. The digital bit recoding in two step encoding and selection shown in figure 2. The recoding method is widely used to generate the partial product for implementation of parallel multiplier[2-3]. For implementation of booth algorithm booth recoding [3]is important and it is widely use for generating partial product for implementing parallel multiplier. The figure II is a architecture of modified booth multiplier are given below there are to operand A and B which are multiplicand and multiplier, the main works of decoder is to convert the given input to equivalent booth value therefore it contains more number of 0'sand the output of decoder, suppose A multiplicand and B is multiplier to form the product z the partial product generating is use AND operation and after adding we get the final product of A*B

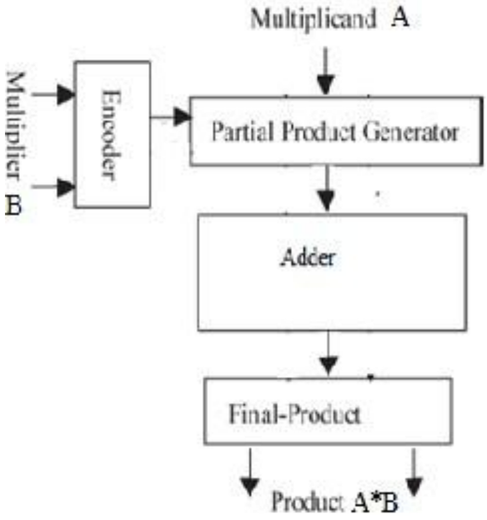


Figure 2: The basic Architecture of Modified Booth Multiplier

Booth recoding unit use to convert multiplier into equivalent booth value here the Table II of booth recoding given below

Block (multiplier bits) B	Re-coded Digit	Operation on A(multiplicand)
00	0	0*A
01	+1	+1*A
10	-1	-1*A
11	0	0*A

Table 1:- Booth Recoding algorithm Table

In Modified Booth Multiplier algorithm Radix-2 append bit 0 in LSB in rightmost to fulfill two bits overlap one bit of previous adjacent bit. According to table I

- If the multiplier bit is '00'and '11' no change put 0000.
- If the multiplier bit is '01'put the value of multiplicand.
- If the multiplier bit is '10' put the value of 2's complement of multiplicand.

Example :- Suppose B(multiplier) and A(multiplicand)

B= -4 (1100)and A= +2 (0010)

Append zeros in LSB of multiplier then by help of booth recording table find partial product and final result

If Multiplier B= -4=1100

1 1 0 0 0

2 bit pairing

a) For LSB is '00' so 0*A, put 0000.

b) Again it is'00' so 0*A, put 0000.

c) Then for '10' so -1*A, put 1110.

d) Then for '11' so 0*A, put 0000.

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
1 1 1 1 1 0
0 0 0 0 0
1 1 1 1 1 0 0 0 = P

The disadvantages number of add/subtract operation inconvenient to design parallel multiplier to overcome from this problem modified booth multiplier Radix-4 or higher Radix come out.

It reduce the number of partial product by N/2 or half instead of adding and shifting for every column and multiply by 1 or 0 we only use second column, and multiply by ±1, ±2 or 0 to get same result and it is high speed multiplier which are used in present scenario for high speed processor because in fact 8.72% of all the instruction in typical processing is multiplier by using this modified booth multiplier because of decrease partial product by half of multiplier. This is important for VLSI circuit design it relate to propagation delay while running the circuit [3] it use less hardware rather than long multiplication method, it use three bit pairing or triplet bit for booth recoding

Table 2 :- Booth Recoding Table Radix - 4

Block(Multiplier bits)B	Re-coded Digit	Operation on A(Multiplicand)
000	0	0*A
001	+1	+1*A
010	+1	+1*A
011	+2	+2*A
100	-2	-2*A
101	-1	-1*A
110	-1	-1A
111	0	0A

Example:- Multiplier B=010011 and multiplicand A=01011 for using recoding technique first three bit pairing done for multiplier we get

+1
0 1 0 0 1 1 0
+1 -1

Three bits pairing by extend LSB by ‘0’ to make a pair of triplet bit then calculate partial product

0 0 1 0 1 1 multiplicand
0 1 0 0 1 1 multiplier
1 1 -1 booth encode multiplier

1 1 1 1 1 1 0 1 0 0
0 0 0 0 1 0 1 1
0 0 1 0 1 1
0 0 0 0 1 error correct negation
0 0 1 1 0 1 0 0 0 1 discard the carried bit = P

V. RESULT

In this section we have written simulation result of unsigned 4 - bit and 8- bit numbers by using Booth Algorithm in Xilinx 12.1design suite in hierarchical VHDL language code for multiplier, and synthesize the multiplier using the Xilinx ISE tools and simulated using Model Sim for comparison different parameters like Power Consumption, Look Up Table, Input Output Block, Current, Number of Slices.

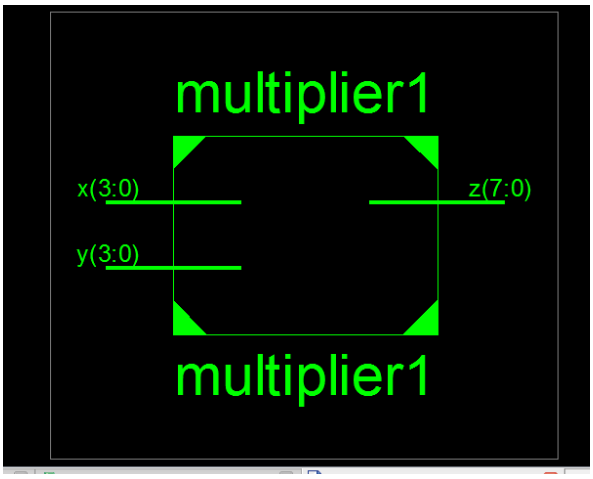


Figure 3: RTL view of 4- bit unsigned number

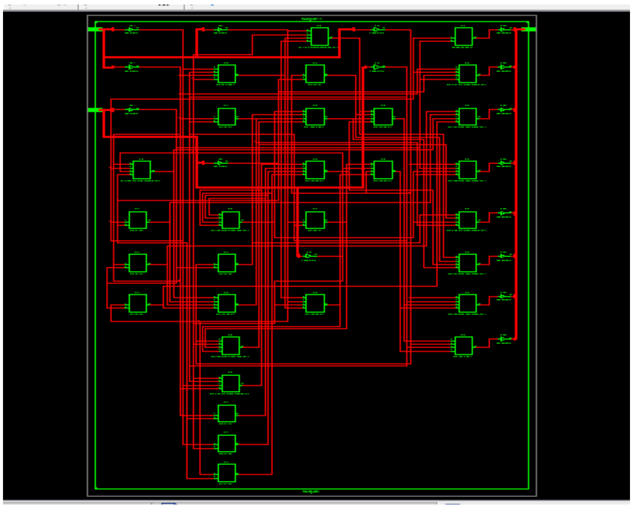
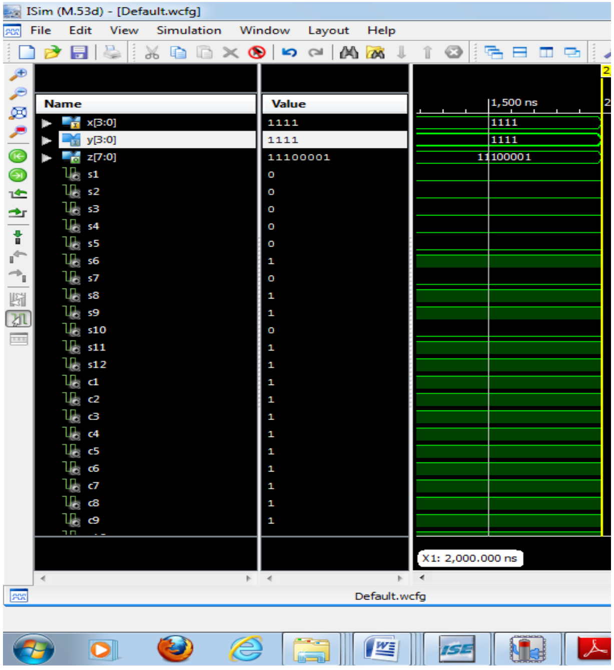


Figure 4: RTL view Technology Schematic for 4- bit Unsigned number



Name	Value
x[3:0]	1111
y[3:0]	1111
z[7:0]	11100001
s1	0
s2	0
s3	0
s4	0
s5	0
s6	1
s7	0
s8	1
s9	1
s10	0
s11	1
s12	1
c1	1
c2	1
c3	1
c4	1
c5	1
c6	1
c7	1
c8	1
c9	1

Figure 5: Product of 4- bit unsigned booth multiplier

Above figure show the waveform of two 4 binary bits 1111 * 1111 and the product is 11100001. The simulator is Isim

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

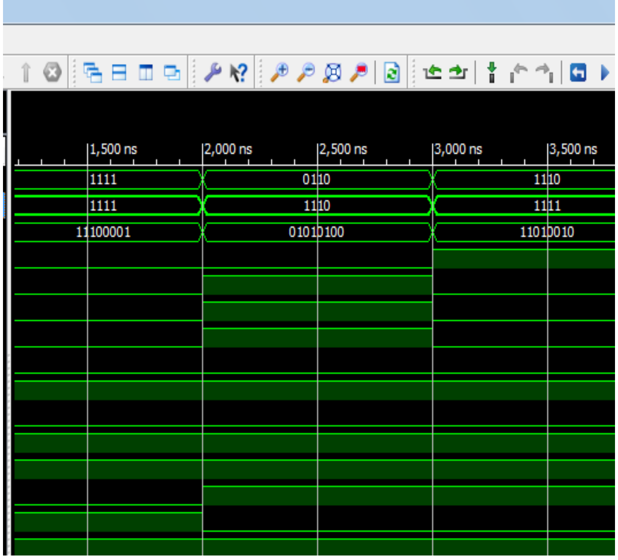


Figure 6: Product of three unsigned 4- bit booth multiplier

Above waveform show the three 4- bit binary number1111*1111=11100001, 0110*1110=01010100 and 1110*1111=11010010 get this product by VHDL coding

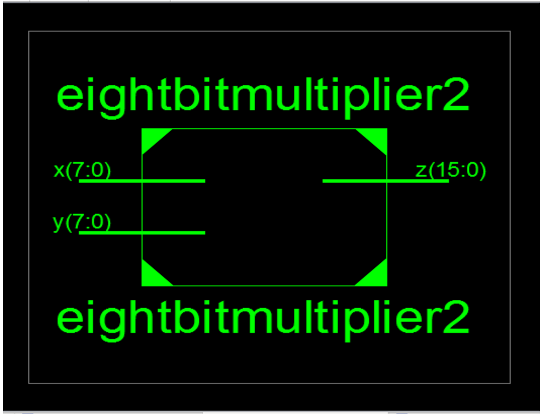


Figure 7: RTL view of 8- bit unsigned number

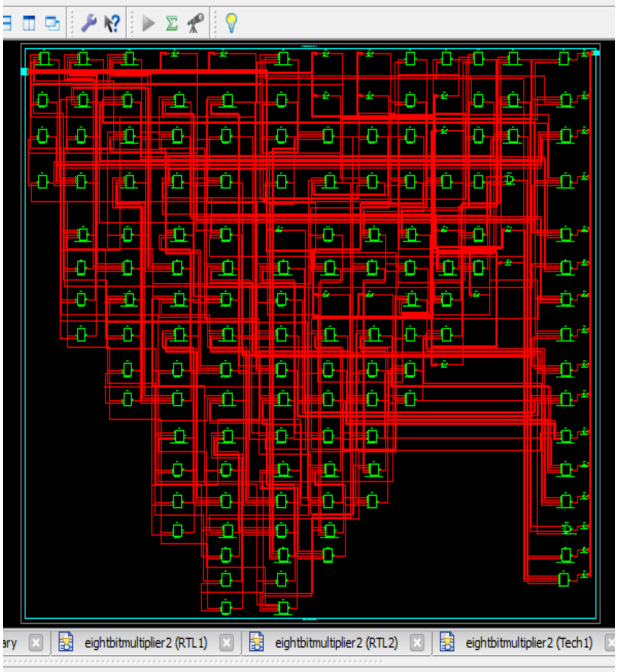
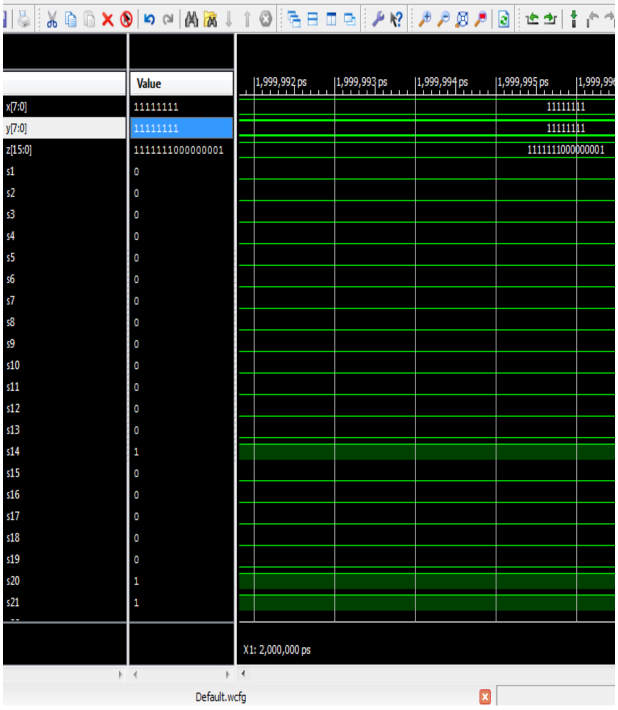

eightbitmultiplier2 (RTL1)
eightbitmultiplier2 (RTL2)
eightbitmultiplier2 (Tech1)

Figure 8: RTL view Technology schematic for 8- bit Unsigned booth multiplier

The above figure show the RTL view of eight bit booth multiplier Technology schematic diagram for unsigned number and simulation result in waveform are below the figure.



	Value
x[7:0]	11111111
y[7:0]	11111111
z[15:0]	1111111000000001
s1	0
s2	0
s3	0
s4	0
s5	0
s6	0
s7	0
s8	0
s9	0
s10	0
s11	0
s12	0
s13	0
s14	1
s15	0
s16	0
s17	0
s18	0
s19	0
s20	1
s21	1

Figure 9: The waveform for product of 8- bit unsigned booth multiplier.

Above waveform show the multiplication of unsigned binary8-bitnumber 11111111*11111111=1111111000000001

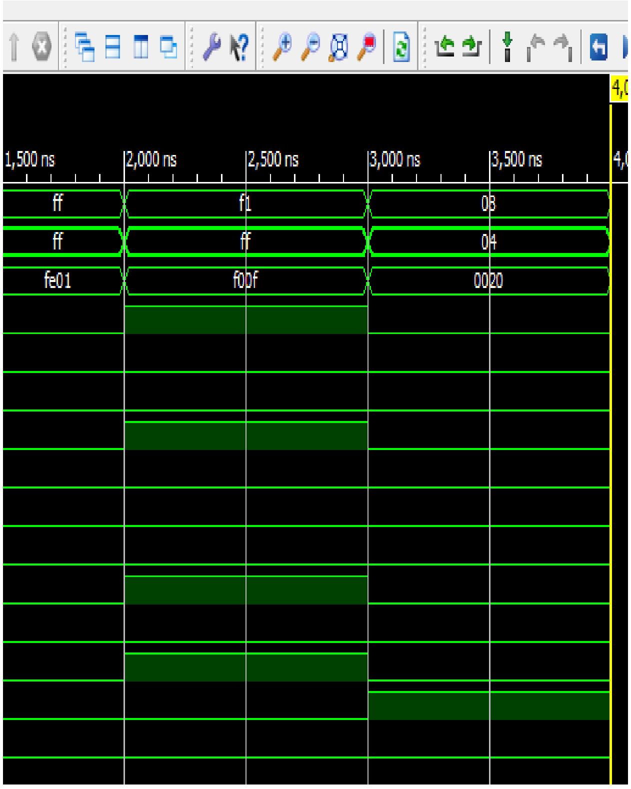


Figure 10: Product of three unsigned 8- bit booth multiplier.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The above figure three multiply 8- bit unsigned number given in hexadecimal form ff * ff = fe01, f1 * ff = f00f and 08 * 04 = 0020

VI. COMPARATIVE RESULT

The comparative result of base and proposed are given below of different parameters for 4- bit and 8- bit unsigned number and reduce parameter occur as compare to previous papers.

Parameter	Base		Proposed	
	Unsigned 4- bit	Unsigned 8- bit	Unsigned 4- bit	Unsigned 8- bit
No. of Slices	9	15	3	7
No. of 4 input LUT	25	43	6	14
No. of bond I/O Block	16	32	12	24
Total Current	14.23mA	14.53mA	0.00097 m A	0.00097 m A
Total Power	36.87mW	37.23mW	0.00097 m W	0.00097 m W

Table 3: Comparative result.

VII. CONCULSION

Multiplier take important role in high performance system, to reduce power and area optimization become an important concern in multiplier because of this Modified Booth Multiplier Algorithm is most popular algorithm used in this paper. It has been implemented in 4- bit and 8- bit unsigned number by using Xilinx 12.1platform for different parameters and reduce power, reduce current and reduce LUT come out.

ACKNOWLEDGEMENT

The study was support “Low Power and Area Optimized for Unsigned Number using Modified Booth Multiplier Algorithm” project at Gyan Ganga Institute of Technology and Science, Jabalpur under guidance of Arif Ansari.

REFERENCES

1. Ravindra P.Rajput, M.N Shanmukha Swamy “ High Speed Modified Booth Encoder Multiplier for Signed and Unsigned numbers” 14th International Conference on Modelling and Simulation 978-0-7695-4682-7/12© 2012 IEEE.
2. W. C. Yen, C. W. Jen, “High Speed Booth encoded Parallel Multiplier Design,” IEEE transactions on Computer, Vol. 49, No. 7, pp. 692-701, July 2000.
3. A.D. Booth, “A Signed Binary Multiplication Technique Quarterly Journal of Mechanics and Applied mathematics, Vol-IV, pt-2-1951.
4. Rainishmi Ranjan, Pramod Mohanty, “A New VLSI Architecture of Parallel Multiplier based on Radix- 4Modified Booth Algorithm Using VHDL”, Inter-national Journal of Computer Science & EngineeringTechnology, ISSN:2229-3345, Vol. 3 No. 4 April 2012.
5. Lenardo Lde Oliveira, Eduaro Costa, Sergio Bampi,Joao Baptista and Jose Monteiro, “array hybridMultiplier versus Modified Booth Multiplier: comparing Area and Power consumption of layout Implementations of signed Radix-4 Architecture”, IEEE, 2004.
6. A.S.Prabhu, V.Elakya, “Design of modified Low Power Booth Multiplier”,IEEE, 2012.
7. Kavita, Jasbir Kaur, “Design and Implementation of an Efficient Modified Booth Multiplier using VHDL”, International Conference on Emerging Trend in Engineering and Management, ISSN:2231-0347, Vol.3(3, July2013).
8. Sukhmeet Kaur, Suman, Manpreet Singh Manna, “Implementation of Modified Booth Algorithm (Radix-4) and its comparison with Booth Algorithm(Radix-2)”, Advance Electronic and Electric Engineering, ISSN 2231-1297, Vol.3, November 6(2013), pp. 683-690.
9. N.H.E. Weste, K.Eshraghain, “Principle Of CMOS VLSI Design, A systems Perspective”, Pearson Education, 2010.