Implementation of Low Power Booth's Multiplier by Utilizing Ripple Carry Adder

Sneha Manohar Ramteke,Yogeshwar Khandagre, Alok Dubey

Abstract—The multiplication operation is performed in many fragments of a digital system or digital computer. Radix 4 modified Booth algorithm can be utilized for reduction of the partial products. The parallel multiplier like radix 4 modified booth multiplier accomplishes the computations utilizing fewer adders and less iterative steps. Based on the simplification of addition operation and power reduction property in ripple carry adder(RCA),a low power radix 4 modified booth multiplier is proposed, compared with the radix 4 modified booth multiplier using carry look ahead adder(CLA),the experimental result shows that our propose design has reduce the power dissipation to 25.27 % using RCA, power has estimated as 8.22mW which was 11mW when designed with CLA Adder.

Index Terms— Booth multiplier, Low power, Modified Booth Multiplier, Multiplication, Partial Product Generation (PPG), RCA., VHDL.

1 Introduction

Multiplication is generally utilized in applications for signal processing, graphics and scientific computation. Various advanced designs of multipliers have been proposed for higher speed, lower power consumption at lesser chip area. Thus high speeds, low power compact VLSI implementations can be accomplished. These three parameters i.e. power, area and speed are always main traded off to accomplish. The two basic operations involved in multiplication process are the generation of partial products and their accumulation.

Entire advancement of microelectronic is to accomplish the energy efficient methodologies. To be energy effective these technologies must possess low-power consumption to fulfill the requirements of many portable applications. The utilization of energy efficient digital signal processing (DSP) modules is very extensive in wireless sensor networks. In the wireless sensor networks like Vehicular Ad-hoc Network (VANET), Mobile Ad-hoc Network (MANET) tens to thousands of battery-operated micro sensor nodes are deployed remotely and used to route the data to the destination node. In such systems, multiplier is essential computational unit which is applied extensively. Basically implementation of multiplier requires large chip area, long latency and consumes considerable power. Therefore design of low-power multiplier is an important concept in low-power VLSI system [6]. The fast multipliers are essential in digital signal processing systems. The speed of multiplication operation finds great significance in digital signal processing and other processors.

The design of a low power high speed Booth multiplier and its implementation on reconfigurable hardware is being proposed. For arithmetic multiplication, various multiplication architectures like array multiplier, Booth multiplier, Wallace tree multiplier and Booth Wallace multiplier have been analyzed. Then it has been found that Booth Wallace multiplier is most efficient among all, giving optimum delay, power and area for multiplication. Low power modified Booth decoder and pipelining techniques have been used to reduce power and delay.

Comparatively, the delay of Wallace tree and Booth Wallace multiplier is almost same and least but power consumption of Wallace tree multiplier is high compared to Booth Wallace multiplier. Power consumption of Booth Wallace is more than array and Booth Multiplier. Hence, Booth Wallace multiplier is used in high-speed applications.
In booth multiplier the number of summands is reduced by recording the multiplier bit into groups that select multiplies of multiplicand. From the basics of Booth Multiplication it can be proved that the addition/subtraction operation can be skipped if the successive bits in the multiplicand are same. To achieve high performance, the modified Booth encoding which reduces the number of partial products by a factor of two through performing the multiplier recoding has been widely adopted in parallel multipliers.

The hardware 3 complexity reduction and power saving can be achieved by directly removing the adder cells of standard multiplier. Due to this a huge truncation error will be introduced. To effectively reduce the truncation error, various error compensation methods, which add estimated compensation value to the carry inputs of the reserved adder cells. The error compensation value can be produced by the constant scheme or the adaptive scheme. The adaptive error compensation approaches are developed only for fixed-width array multipliers and cannot be applied to significantly

reduce the truncation error of fixed-width modified Booth multipliers directly. To overcome this problem, several

- *Sneha Manohar Ramteke is currently pursuing masters degree program inVLSI in RGPV University, India, E-mail: sneha07pro@gmail.com*

error compensation approaches have been proposed to effectively reduce the truncation error of fixed-width modified Booth multipliers. To obtain better error performance with a simple error compensation circuit, Booth encoded outputs are utilized to generate the error compensation value.

2 LITERATURE SURVEY

Simran Kaur & Manu Bansal has designed FPGA implementation of Modified Booth Wallace Multiplier to make the multiplier faster & reduce the power consumption [1]. In "Parallel MAC Based on Radix-4 & Radix-8 Booth Encodings" authors has enhanced the speed of parallel MAC (Multiplier & accumulator) by using a new Radix-5 Kogge stone adder [2]. The proposed MBM utilizes the carry select Adder (CSA) and 3-stage pipelining technique to improve the performance by reducing delay time [3].

In the "VLSI Architecture of Parallel Multiplier–Accumulator Based on Radix-2 Modified Booth Algorithm" authors proposed CSA tree uses 1's-complement-based radix-2 modified Booth's algorithm (MBA) and has the modified array for the sign extension in order to increase the bit density of the operands [4]. In "Fixed Width Modified Booth Multiplier for High Accuracy", the fixed-width modified Booth multipliers are proposed for high-accuracy [5]. In "Efficient Implementation Of 16-Bit Multiplier-Accumulator Using Radix-2 Modified Booth Algorithm and Spst Adder Using Verilog" The proposed radix-2 modified Booth algorithm MAC with SPST gives a factor of 5 less delay and 7% less power consumption as compared to array MAC [6]. In "Architecture of a Floating Point Register for an Experimental RISC CPU" an 8-bit RISCCPU is designed at gate level using completely custom chip approach. CPU has an 8-bit integer unit and 16bit floating point unit [7]. Nishant Bano has proposed a design of modified Booth Multiplier for low power consumption [9].

The authors Soojin Kim and Kyeongsoon Cho in "Design of high-speed modified Booth multipliers operating at GHz ranges" has proposed the design for multiplication operation at very high speed (i.e very high frequency or in GHz range) [10]. The authors of "Low voltage Low-power VLSI Subsystems" have discussed the various low power VLSI system design issues to reduce the system power consumption [11]. Spartan-3E FPGA Starter Kit Board User Guide released in June 20, 2008 gives guidelines to use it& its versatility for the applications [14].

3 PROPOSED METHODOLOGY

To multiply, multiplicand 'X' by multiplier 'Y' using the modified Booth algorithm. First group the multiplier bits 'Y' by three bits and encoding into one of {-2, -1, 0, 1, 2} as shown in Table I. Prior to convert the multiplier, a zero is appended into the Least Significant Bit (LSB) of the multiplier. Table I shows the rules to generate the encoded signals by MBE scheme and Fig. 3(a) shows the corresponding logic diagram. The Booth decoder generates the partial products using the encoded signals as shown in Fig. 3(b).

3.1 Algorithm OF MULTIPLICATION:

To multiply A& B (8 bit), the radix 4modified booth multiplier uses n/2 cycles where each cycle examines 3 adjacent bits of A, add or subtract 0 ,B OR 2B to the partial product and shifts partial product to the right. Fig 1 shows the flowchart of the radix 4 modified booth algorithm. Following steps are followed for obtaining (A*B)

Step 1: Initialize the values of p =j=a-1= 0 and value of n =8.

Step 2: Examine 3 adjacent bits of the 8 bit number A, after appending 'zero' to the LSB.

Step 3: According to the value of the 3 bits of the multiplicand A ,operation of 0 ,+2B ,-2B ,+B,-B are perform on the partial product P.

Step 4: Check the value of j with value of n

if j<n :make P= 4P and i=j+2 and jump back to step 2 for loop 2

Step 5: If j≥n : final product of multiplication A*B =P,16 bit result is obtained in P

where A=multiplicand, B=multiplier, P=partial product

j=position of multiplier bit A

n=Length of the number(A,B)

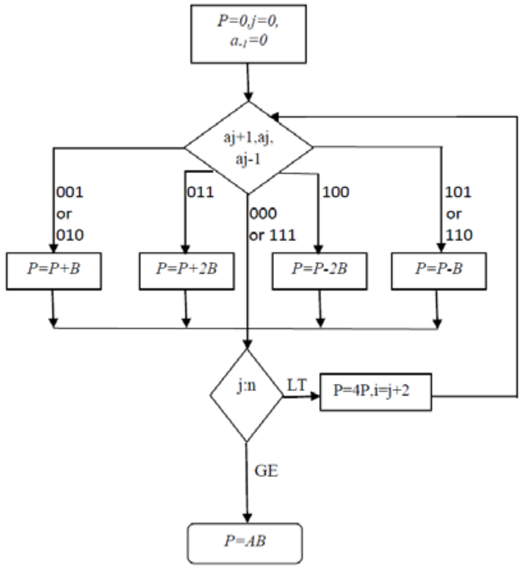


Fig. 1. Flowchart of Radix 4 Modified Booth Algorithm

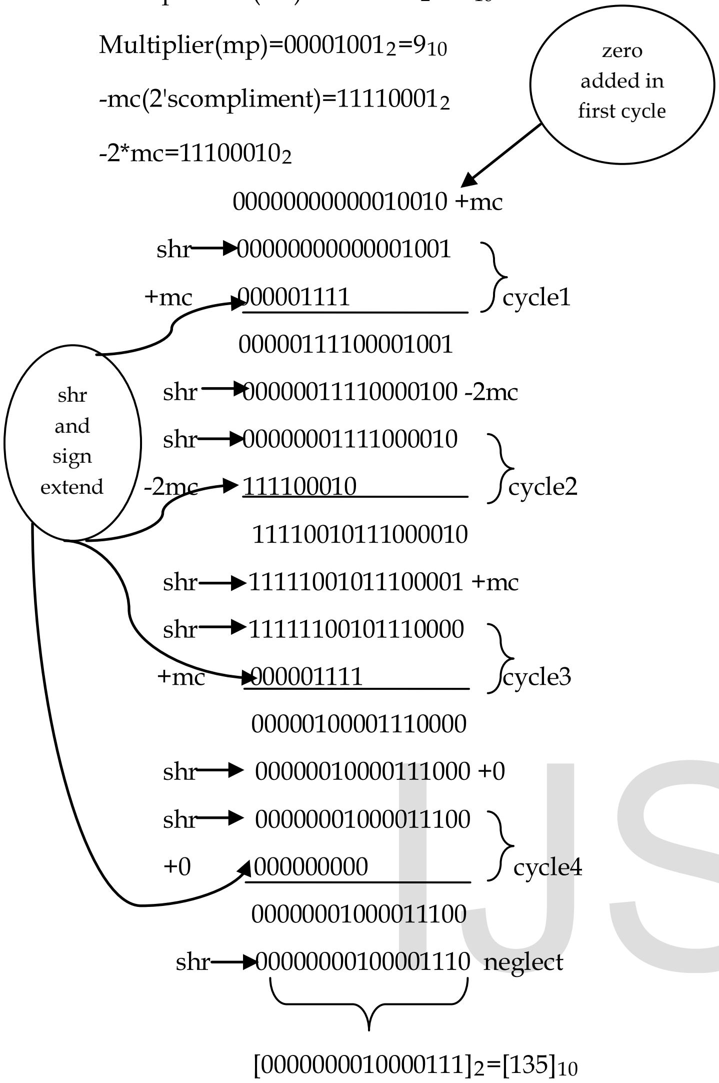
Example : multiplication of 15_{10}*9_{10}, using radix 4 modified Booth algorithm.

Multiplicand (mc)=00001111$_2$=15$_{10}$

Multiplier(mp)=00001001$_2$=9$_{10}$

-mc(2'scompliment)=11110001$_2$

-2*mc=11100010$_2$



```
                 000000000000010010 +mc
        shr→000000000000001001  }
        +mc 000001111_________  }cycle1
            00000111100001001
        shr→00000011110000100 -2mc
        shr→000000001111000010  }
       -2mc→111100010_________  }cycle2
             11110010111000010
        shr→11111001011100001 +mc
        shr→11111100101110000   }
        +mc→000001111_________  }cycle3
             00000100001110000
        shr→ 00000010000111000 +0
        shr→ 00000001000011100  }
        +0   000000000_________ }cycle4
             00000001000011100
        shr→00000000100001110 neglect
```

[0000000010000111]$_2$=[135]$_{10}$

Y_{n+1}	Y_n	Y_{n-1}	Z_n	Operate	Neg	Zero	Two
0	0	0	0	0	0	0	0
0	0	1	1	1 x M	0	1	0
0	1	0	1	1 x M	0	1	0
0	1	1	2	-2 x M	0	0	1
1	0	0	-2	-2 x M	1	0	1
1	0	1	-1	-1 x M	1	1	0
1	1	0	-1	-1 x M	1	1	0
1	1	1	0	0	0	0	0

Table 1. Codes For Movement Direction

The block diagram below shows the modified Booth Multiplier. The building blocks are Modified Booth Encoder, Partial Product Generator (PPG), Sign Extension Corrector and a Ripple Carry Adder.

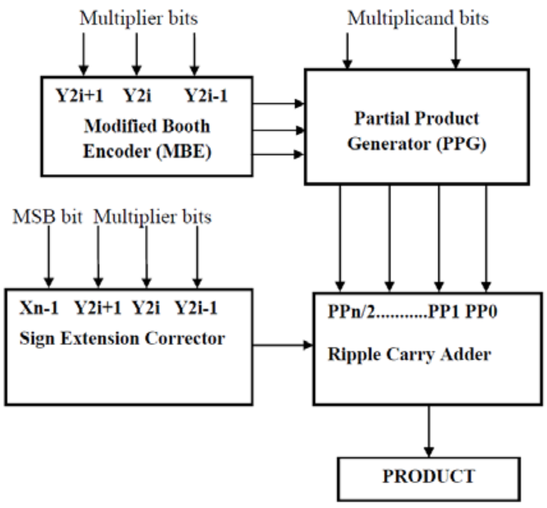
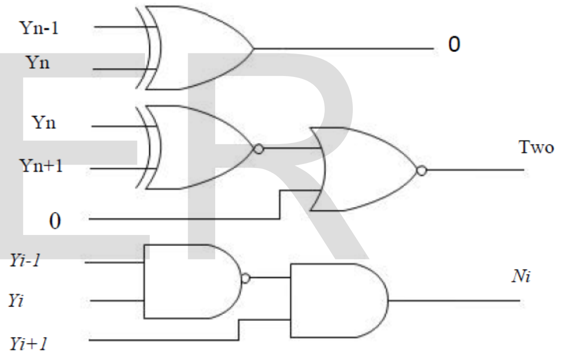
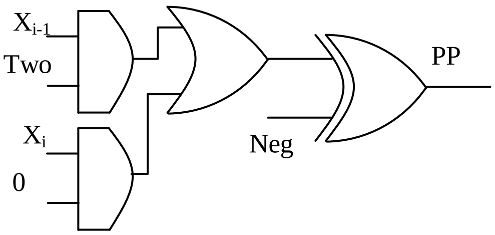


Fig .2. Block Diagram of Modified Booth Multiplier



(a)

Fig .3 (a).Modified Booth Algorithm Encoder

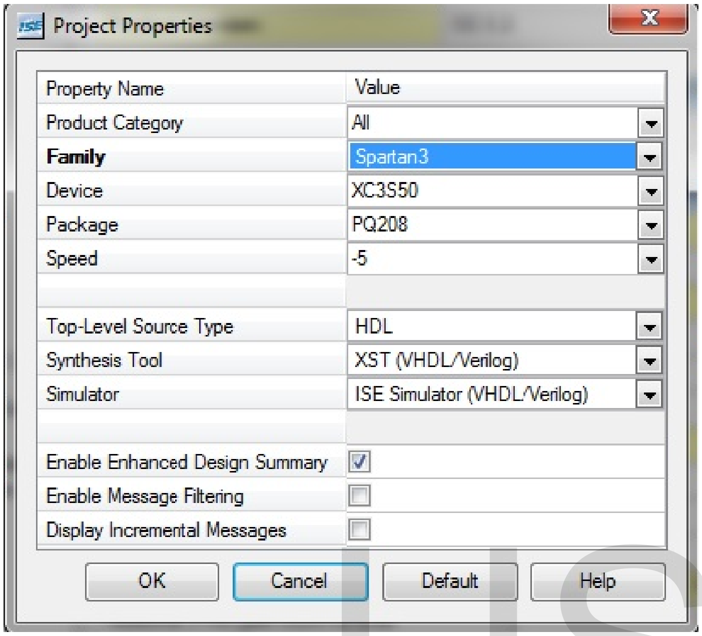


(b)

Fig .3 (b). Modified Booth Decoder

4 RESULT

Xilinx 8.2i ISE Simulator has been used to simulate the proposed methodology of multiplication of two 8 bit numbers using Radix-4 modified Booths algorithm. The device used by the simulator is XC3S50 of Spartan3 family with the speed of -5 which is shown in Fig. 4.



Property Name	Value
Product Category	All
Family	Spartan3
Device	XC3S50
Package	PQ208
Speed	-5
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISE Simulator (VHDL/Verilog)
Enable Enhanced Design Summary	☑
Enable Message Filtering	□
Display Incremental Messages	□

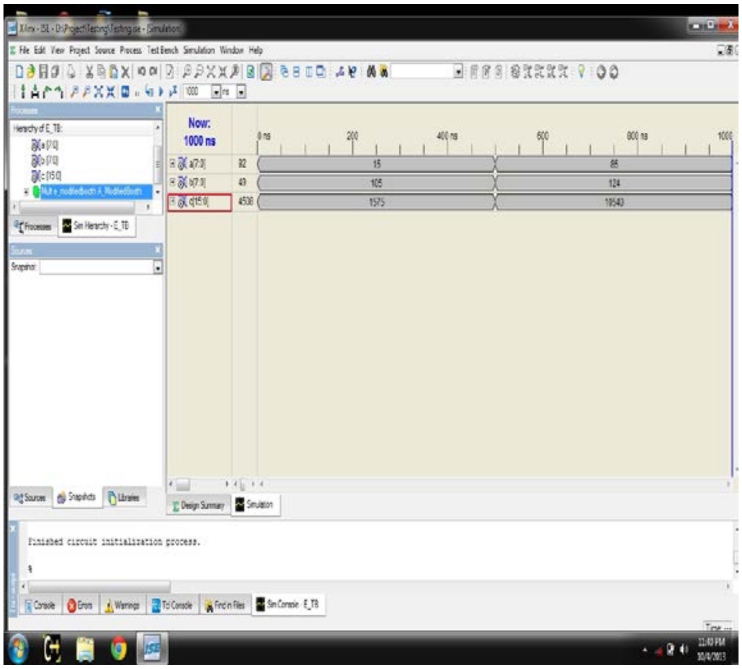
Fig. 4. Properties of Xilinx 8.2i module

The Result Can be analyze by following table,

Input a[7:0]	Input b[7:0]	Output c [16:0]	Delay (ns)
92	49	4508	0
15	105	1575	500
85	124	10540	1000

Table 2. Multiplication of a and b

The simulation result for multiplication of two 8-bit numbers is shown in the Fig. 5. Initially a=92 and b=49 with delay of 500 ns. Therefore, after 500ns the multiplication result is available in c = 4508. The power analysis of the Modified Booth's Multiplier using ripple carry adder is shown in Fig. 7. Reduction in dynamic power from 11 mW to 8.22 mW has been accomplished.



```
Finished circuit initialization process.
```

Fig. 5. Simulation results of multiplication of two 8-bit numbers.

Figure below shows the RTL schematic of the multiplier using the ripple carry adder.

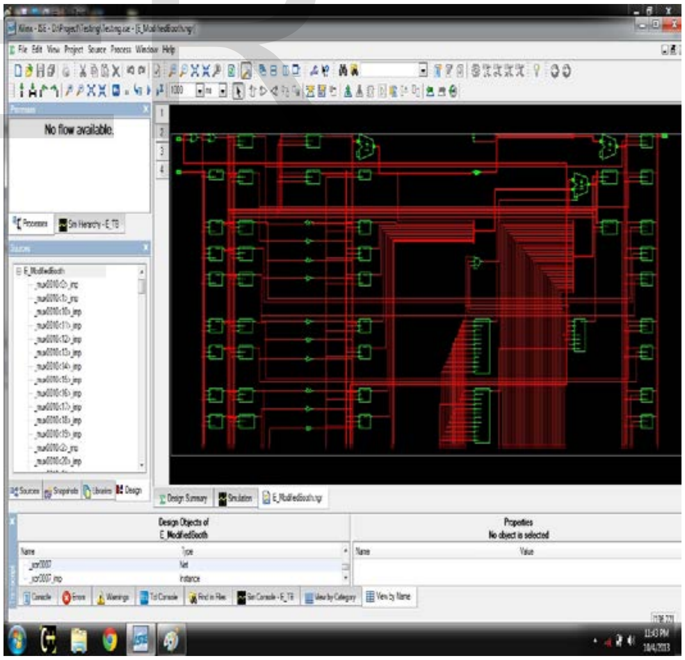
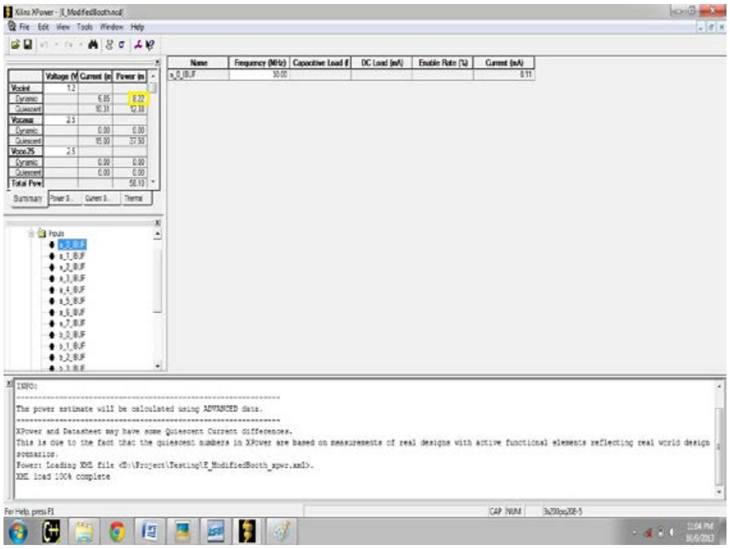


Fig. 6. RTL Schematic of multiplier



	Voltage (V	Current (m	Power (m
Vccint	1.2		
Dynamic		6.85	8.22
Quiescent		10.31	12.38
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		15.00	37.50
Vcco25	2.5		
Dynamic		0.00	0.00
Quiescent		0.00	0.00
Total Pow			58.10

Name	Frequency (MHz)	Capacitive Load (f	DC Load (mA)	Enable Rate (%)	Current (mA)
a_0_IBUF	10.00				0.11

```
INFO:
------------------------------------------------------------
The power estimate will be calculated using ADVANCED data.
------------------------------------------------------------
XPower and Datasheet may have some Quiescent Current differences.
This is due to the fact that the quiescent numbers in XPower are based on measurements of real designs with active functional elements reflecting real world design scenarios.
Power: Loading XML file <D:\Project\Testing\E_ModifiedBooth_xpwr.xml>.
XML load 100% complete
```

Fig. 7.Power analysis of proposed Modified Booth's Multiplier using Ripple Carry Adder.

5 Conclusion

The radix 4 modified booth multipliers using RCA is realized using VHDL. The analysis shows that power dissipation proposed by modified Booth's multipliers using RCA is 8.22 mW as compared to the radix 4 Booth's multiplier using CLA which is 11mW. In future, to improve performance of multiplier pipelining is proposed.

Properties	Radix4 Booth Multiplier Using CLA Adder	Radix4 Booth Multiplier Using RCA
Family	Spartan 2	Spartan 3
% Power Reduction	22.9%	25.27%
Power Dissipation (Dynamic) mW	11	8.22

Table 3.Comparision of CLA & RCA properties

References

[1] Simran Kaur & Manu Bansal , "FPGA Implementation of Modified Booth Wallace Multiplier",June 2011.

[2] Shankey goel & R.K. Sharma, "Parallel MAC Based On Radix-4 & Radix-8 Booth Encodings", International Journal of Engineering Science and Technology (IJEST) Vol. 3 No. 8 August 2011.

[3] Kulvir Singh & Dilip Kumar, "Modified Booth Multiplier with Carry Select Adder using 3-stage Pipelining Technique", International Journal of Computer Applications (0975 – 8887) Volume 44– No14, April 2012.

[4] Mr.M.V.Sathish, Mrs Sailaja, "Vlsi Architecture Of Parallel Multiplier–Accumulator Based On Radix-2 Modified Booth Algorithm", International Journal of Electrical and Electronics Engineering (IJEEE), Volume-1, Issue-1, 011.

[5] S.SHABEERKHAN et. al, "Fixed Width Modified Booth Multiplier For High Accuracy", International Journal of Research in Advanced Electronics -IJRAE Vol 01, Issue 01; April 2012.

[6] Addanki Purna Ramesh, Dr.A.V. N. Tilak and Dr.A.M.Prasad, "Efficient Implementation Of 16-Bit Multiplier-Accumulator Using Radix-2 Modified Booth Algorithm And Spst Adder Using Verilog", International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.3, June 2012.

[7] Ajay A Joshi, Siew Lam, Yee Chan, "Architecture of a Floating Point Register for an Experimental RISC CPU", International Journal of Engineering and Technology Volume 2 No. 5, May, 2012.

[8] Pooya Asadi, " A New Opitimized Tree Structure In Speed Modified Booth Multiplier Architecture", American Journal of Scientific ResearchIssue 52 (2012), pp. 48-56.

[9] Nishat Bano, "VLSI Design of Low Power Booth Multiplier"International Journal of Scientific & Engineering Research, Volume 3, Issue 2, February -2012.

[10] Soojin Kim and Kyeongsoon Cho, "Design of high speed modified Booth multipliers operating at GHz ranges", World Academy of Science, Engineering and Technology, 2010.

[11] Roy, Kaushik, Yeo, and Kiat-Seng, "Low voltage Lowpower VLSI Subsystems", McGraw-Hill, pp.124-141.

[12] A. Dandapat, S. Ghosal, P.Sarkar, D.Mukhopadhyay, "A 1.2ns 16X16-bit binary multiplier using high speed compressors", International Journal of Electrical and Electronics Engineering 4:3, 2010.

[13] Gina R. Smith, "FPGAs 101: Everything you need to know to get started", Elsevier, 2010.

[14] "Spartan-3E FPGA Starter Kit Board User Guide", UG230 (v1.1) June 20, 2008.

[15] Razaidi Hussin, Ali Yeon Md. Shakaff, Norina Idris, Zaliman Sauli, Rizalafande Che Ilsmail, and Afzan Kamaraudin, "An efficient modified Booth multiplier architecture", International Conference on Electronic Design, 978-1-4244-2315-6/08,2008 IEEE.

[16] S. K. Mangal and R. M. Badghare, "FPGA Implementation of Low Power Parallel Multiplier", 20th International Conference on VLSI Design, IEEE, 2007.

[17] Deming Chen, Jason Cong, and Peichan Pan, "FPGA Design Automation: A Survey", Foundations and Trends in Electronic Design Automation, vol. 1, Issue 3, November 2006.

[18] Ken Chapman, "Initial Design for Spartan-3E Starter Kit (LCD Display Control)", Xilinx Ltd 16th February 2006.

[19] K.H. Tsoi, P.H.W. Leong, "Mullet - a parallel multiplier generator," fpl, pp.691-694, International Conference on Field Programmable Logic and Applications, 2005.

[20] Weste, Neil H.E. Eshraghian, and Kamr an, "CMOS VLSI Design: A Circuits and Systems Perspective", 3rd Edition, Pearson Education, pp. 345-356, 2005.

[21] M. C. Wen, S. J. Wang, Y.N. Lin, "Low -Power parallel multiplier with column bypassing", ELECTRONICS LETTERS, vol. 41, no. 10, 12th May 2005.

[22] Oscal T. C. Chen, et.al, "Minimization of switching activities of partial products for designing low power multipliers", IEEE Trans. VLSI systems, pp. 418-433, vol.11, no. 3, June 2003.

[23] M.O Lakshmanan, Alauddin Mohd Ali, "High Performance Parallel Multiplier Using Wallace-Booth Algorithm," IEEE International Conference on Semiconductor Electronics, pp. 433-436, 2002.

[24] A. A. Fayed and M. A. Bayoumi, "A Novel Architecture for Low-Power Design of Parallel Multipliers," Proceedings of the IEEE Computer Society Workshop on VLSI, pp.149-154, 2001.