PROFICIENT REALIZATION OF ENHANCED BOOTH MULTIPLIER FOR SIGNED AND UNSIGNED BITS

Jani Basha Shaik*, Susrutha Babu Sukhavasi[#], Suparshya Babu Sukhavasi[#],
Sai Nikhil.M.V.S[*], M.Aravind[*]
[#]Faculty, Department of ECE, K L University, Guntur, AP, India.
*B.Tech Students, Department of ECE, K L University, Guntur, AP, India.

ABSTRACT

Multipliers play vital role in most of the high performance systems. Performance of a system depends mostly on the performance of multiplier thus multipliers should be fast and consume less area and hardware. This paper introduces the configuration and execution of Enhanced Modified Booth multiplier for both signed and unsigned numbers augmentation. Generally the booth encoding method is used to generate the partial products for implementation of large parallel multiplier for all unsigned and some signed bits only, on by adopting the parallel encoding scheme. The necessity of the current circuit framework is a devoted and high speed exceptional multiplier unit for signed and unsigned numbers. The proposed efficiency enhanced booth multiplier can perform parallel encoding scheme for both signed and unsigned bits completely. The proposed one was simulated using Xilinx ISE design suite 14.2 tool and implemented on degilent nexus 2 kit, FPGA.

KEYWORDS: *Multiplier, Booth multiplier, parallel encoding, signed bits, unsigned bits.*

I. INTRODUCTION

1.1 Multiplication

Multiplication is the basic process involved in many of the electronic circuits[1]. It is mostly used in microprocessors, graphics drivers, and digital signal processors. The multiplication process consists of formation of the product of the two unsigned or positive binary numbers. This is done through a very normal procedure of simplified to base 2. As an example, the multiplication of two positive 5 bit binary numbers 21 and 25 (in decimal) proceeds as shown below.

Example

21:- 1 0 1 0 1 - multiplicand

25:- 1 1 0 0 1 - multiplier

X

---

1 0 1 0 1

0 0 0 0 0 partial

0 0 0 0 0 products

1 0 1 0 1

1 0 1 0 1

---

1 0 0 0 0 0 1 1 0 1:- 525 - product

M x N- bit multiplication P=(Y) x(X)is obtained by forming the N partial products and then adding the appropriately shifted partial products to generate an M+N bit result (P). The Binary multiplication is equivalent to the logical AND operation. Therefore, producing partial products consists of the logical AND of the approximate bits of the multiplier and multiplicand. Each column of the partial products should then be added and, if necessary, any carry values passed to the next column. We name the multiplicand as Y= $\{y_{m-1}, y_{m-2},\ldots,y_1,y_0\}$ and the multiplier as X=$\{x_{n-1},x_{n-2},\ldots,x_1, x0\}$. The product equation for the unsigned multiplication is shown below.

$$P = \left(\sum_{j=0}^{M-1} y_j 2^j\right)\left(\sum_{i=0}^{N-1} x_i 2^i\right) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} x_i y_j 2^{i+j}$$

Partial products

						y_5	y_4	y_3	y_2	y_1	y_0
						x_5	x_4	x_3	x_2	x_1	x_0
						x_0y_5	x_0y_4	x_0y_3	x_0y_2	x_0y_1	x_0y_0
					x_1y_5	x_1y_4	x_1y_3	x_1y_2	x_1y_1	x_1y_0	
				x_2y_5	x_2y_4	x_2y_3	x_2y_2	x_2y_1	x_2y_0		
			x_3y_5	x_3y_4	x_3y_3	x_3y_2	x_3y_1	x_3y_0			
		x_4y_5	x_4y_4	x_4y_3	x_4y_2	x_4y_1	x_4y_0				
	x_5y_5	x_5y_4	x_5y_3	x_5y_2	x_5y_1	x_5y_0					
p_{11}	p_{10}	p_9	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

1.2 Array Multiplication

Fast multiplication uses CSA, (Carry Save Adder), to add the output partial products[6]. A CSA specifically has a delay of about (1.5 – 2)FO4 inverters does not depend on the size of obtained partial product, whereas CPA-Carry Propagate Adder have a delay which can be given as4-15+ FO4 inverters depending on the circuit family, size and architecture. The below diagram represents 4 x 4 array multiplier with the help of an array of CSAs for unsigned numbers. Every cell consists of a AND gate with 2-inputs that produces partial product and a CSA (full adder) to add partial product into running sum. The first row transforms the 1st partial product into a carry-save unnecessary form. The Carry Save Adder (CSA) is used in each later row to add the respectively partial product to a carry-save unnecessary result of the past row and generate a carry-save unnecessary result. The most essential output bit arrive in carry save unnecessary result form and require an M-bit CPA-(carry Propagate Adder) to convert into normal form of binary system.

In the below diagram, the Carry Propagate Adder is built as a ripple-carry adder. The array is normal in construction and uses a single type of cell, so it is easy to design and make a layout. Considering the carry output as faster to the sum output in a Carry Save Adder, the dashed line in the diagram gives the critical path through the array. The adder can now easily be pipelined with the placement of registers between the rows. In practice, circuits are assigned rectangular blocks in the floor plan so the parallelogram geometry uses a lot of space.

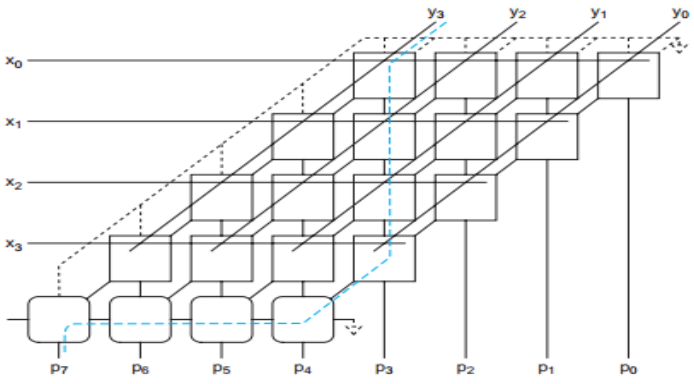


Fig 1: partial products for multiplication

II. BOOTH MULTIPLICATION

2.1 Booth encoding

The array multiplication we used is having more delay due to propagation delay in the carry save adder. The applications in the digital systems needs a very less time delay for propagation and partial product generation. Hence we need to enhance the multiplication process. So the booth multiplication was coined to fasten the serial multiplication process. Whereas the improved booth multiplier provides higher radix parallel operation without generating the difficult 3Y multiple by instead negative partial product usage. The array multiplication computes the partial products for radix 2 manner. This gives that 1 binary digit of the multiplier at the same time.The Radix 2 power r multipliers gives (N/r) partial products. These every 1 depends on 'r' bits of multiplier. Some partial products lead to a faster and smaller carry sum adder array. But an efficient booth multiplier is used in veryless time delay mode. The table that is given below shows the algorithm for the modified booth multiplier.

ENCODING OF RADIX-4 BOOTH MULTIPLIER

Events	Output products
0 0 0	0
0 0 0	1 x Multiplicand
0 0 0	1 x Multiplicand
0 0 0	2 x Multiplicand
1 0 0	-2 x Multiplicand
1 0 1	-1 x Multiplicand
1 1 0	-1 x Multiplicand
1 1 1	0

Example for booth multiplier

									0	0	1	0	0	0	1	0	Input 1 34
									1	1	0	1	0	1	1	0	Input 2 -42
						1	1	1	1	0	1	1	1	1	0	0	Obtained Product 1
				1	1	0	0	1	0	0	0	1	0	0			Obtained Product 2
		1	1	0	0	0	1	0	0	0	1	0					Obtained Product 3
1	0	1	1	1	0	1	1	1	1	0							Obtained Product 4
1	1	1	1	1	1	0	1	0	0	1	1	0	1	1	0	0	Final product - 14 28

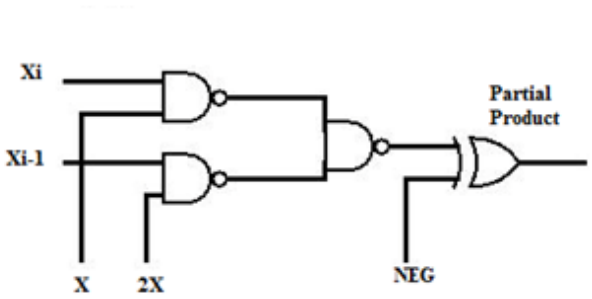


Fig 2: Partial Product Generator

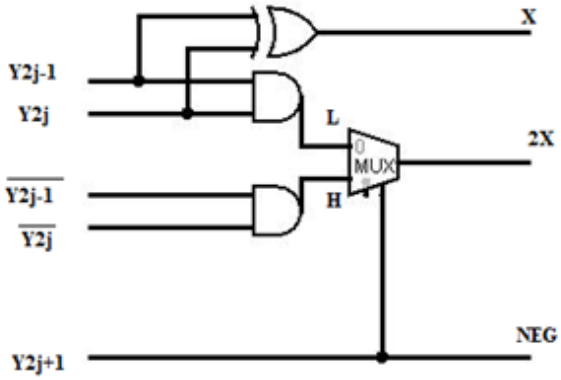


Fig 3: Booth's Multiplier

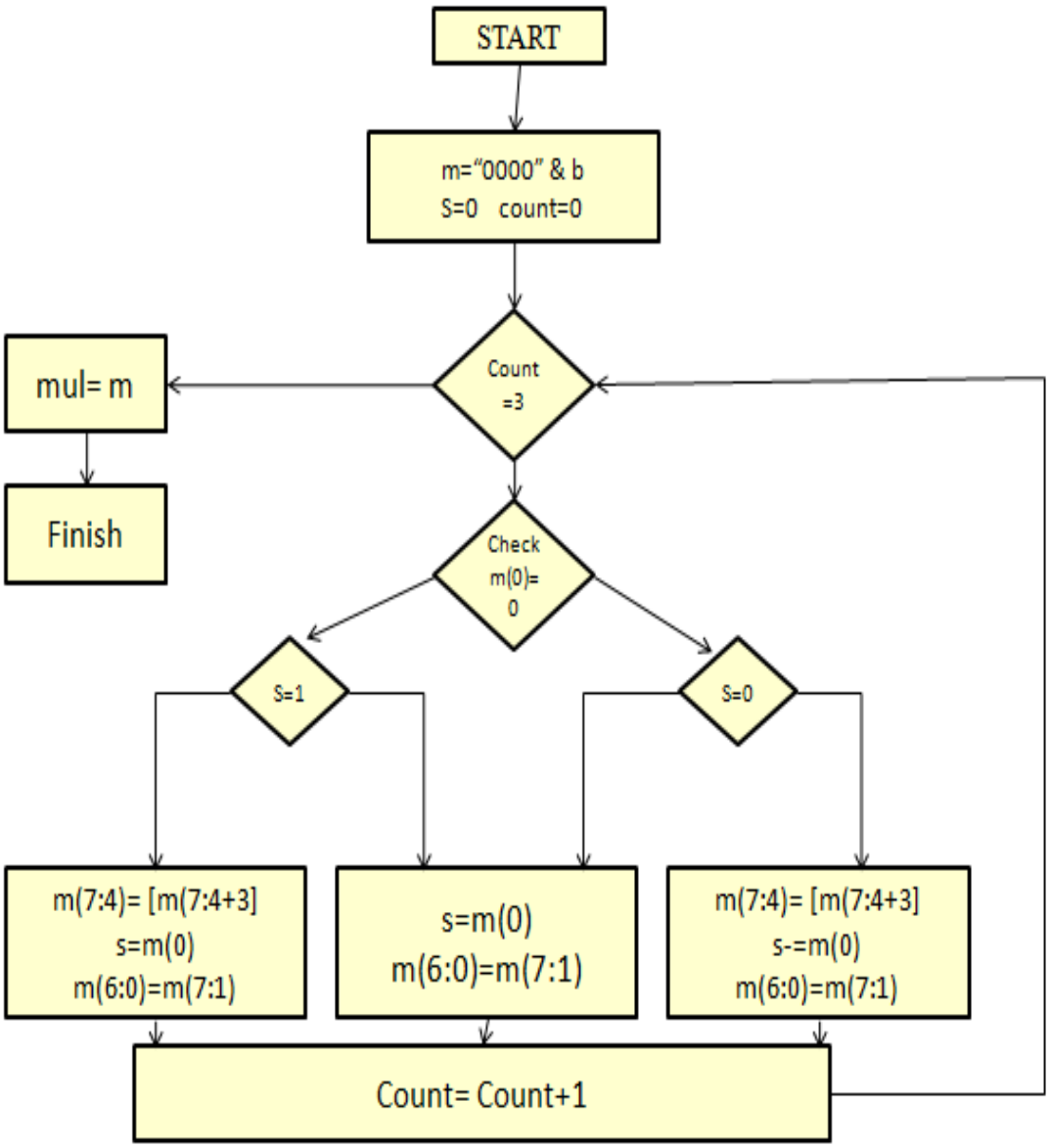


Fig4: Flow chart for Booth's Algorithm

III. HARD WARE SET UP

Fig: 5 FPGA SET UP

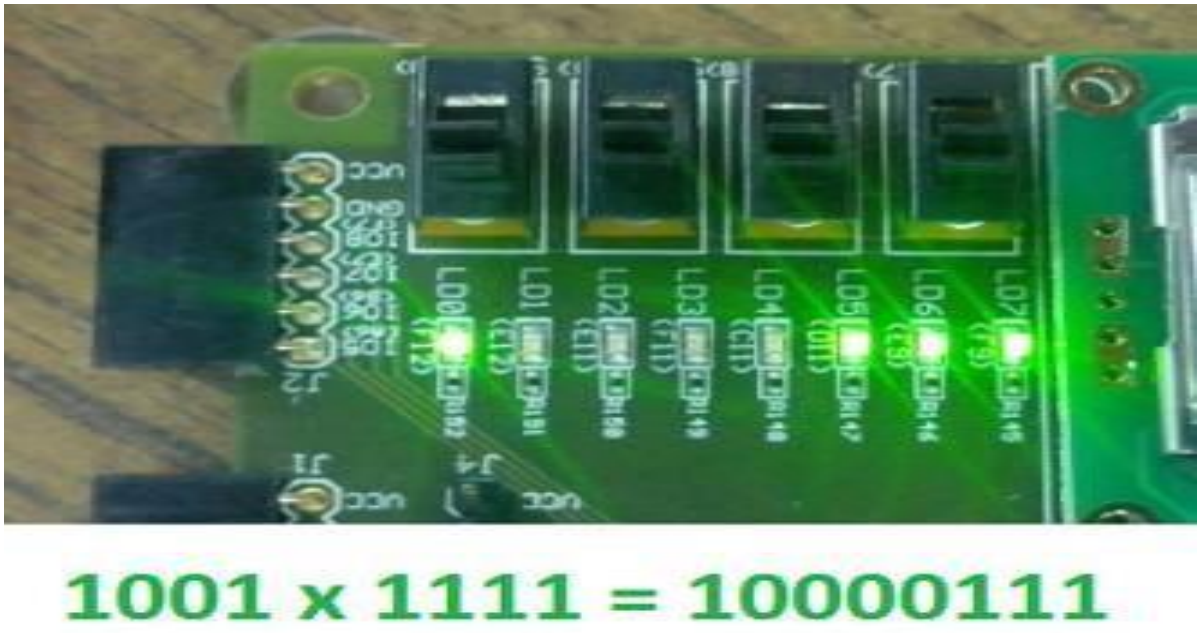


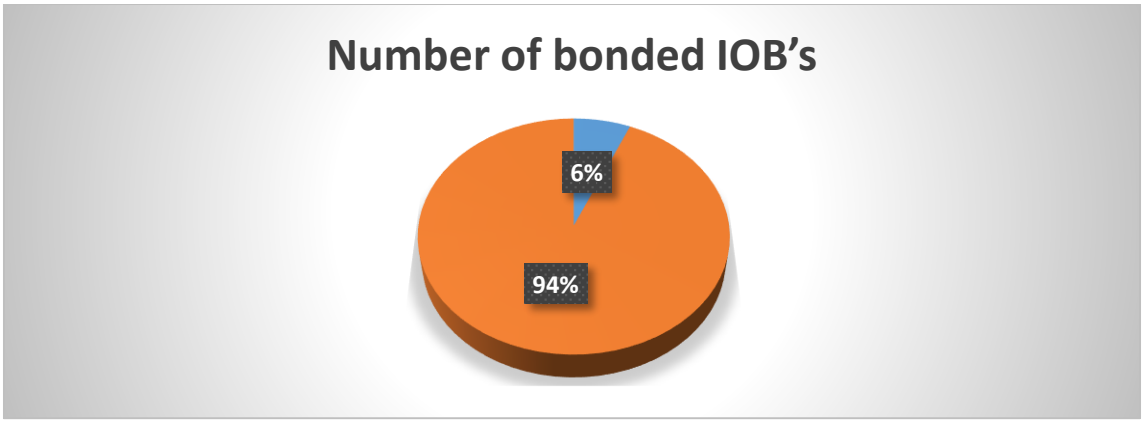
Fig: 6 Hard Ware Implementation Of Modified Booth Multiplier

IV. ANALYSIS

Device Utilization Summary

LOGIC UTILIZATION	USED	AVAILABLE	USED
Number of slices	21	4656	0%
Number of 4input LUT'S	37	9312	0%
Number of bonded IOB's	16	232	6%

Percentage of Bonded Iobs



V. OUTPUT WAVEFORMS

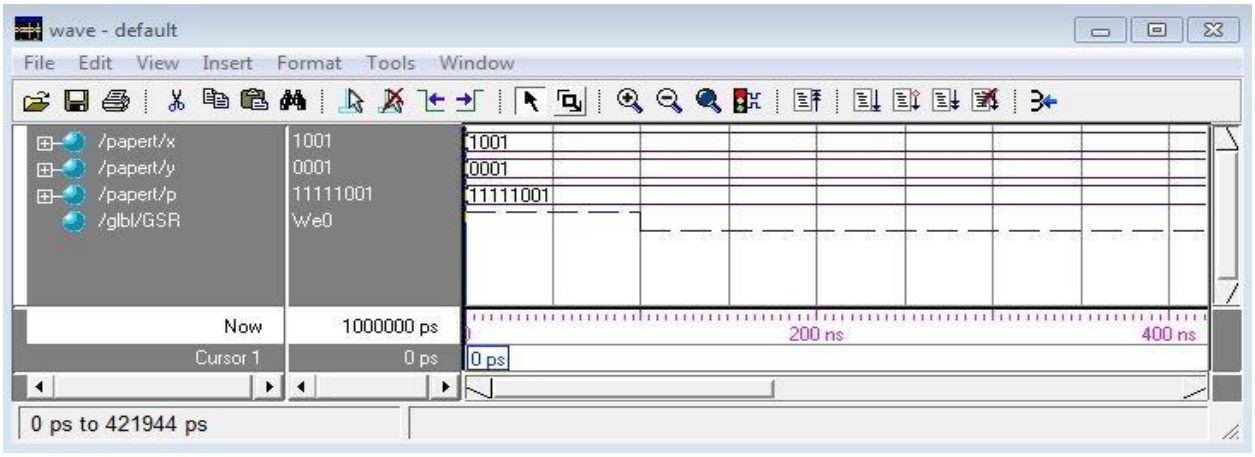


Fig:-7 improved booth multiplication of 1001 x 0001

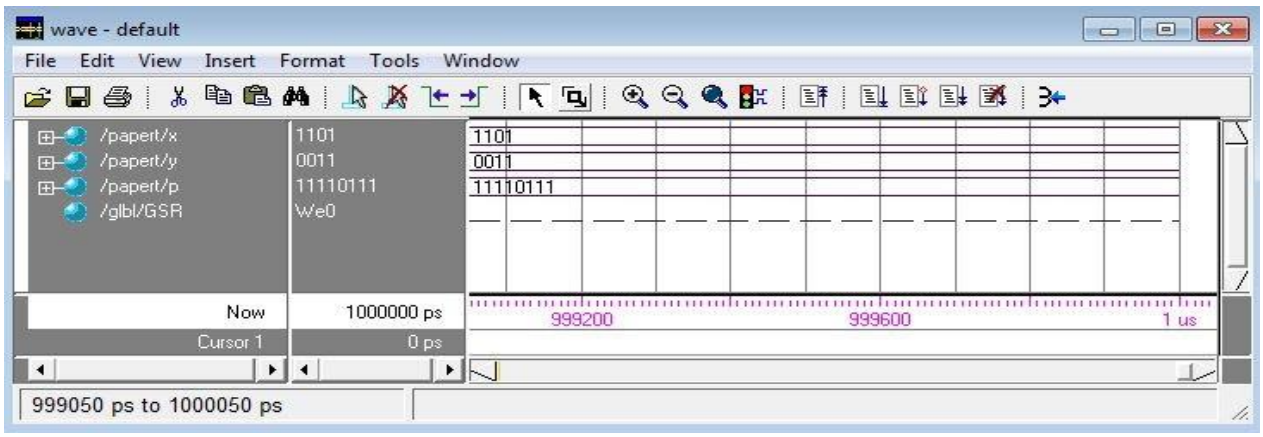


Fig:- 8 improved booth multiplication of 1101 x 0011

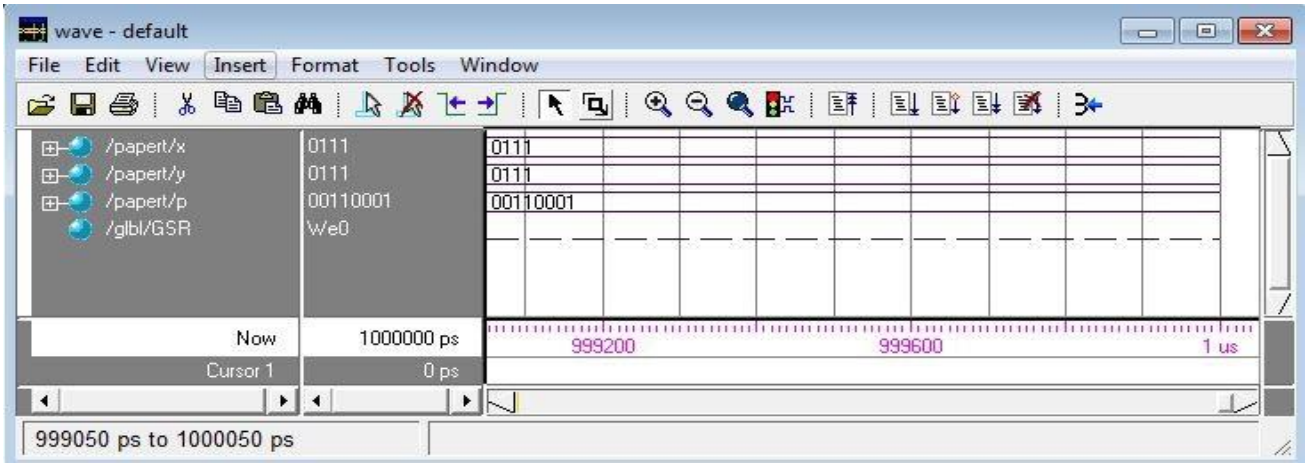
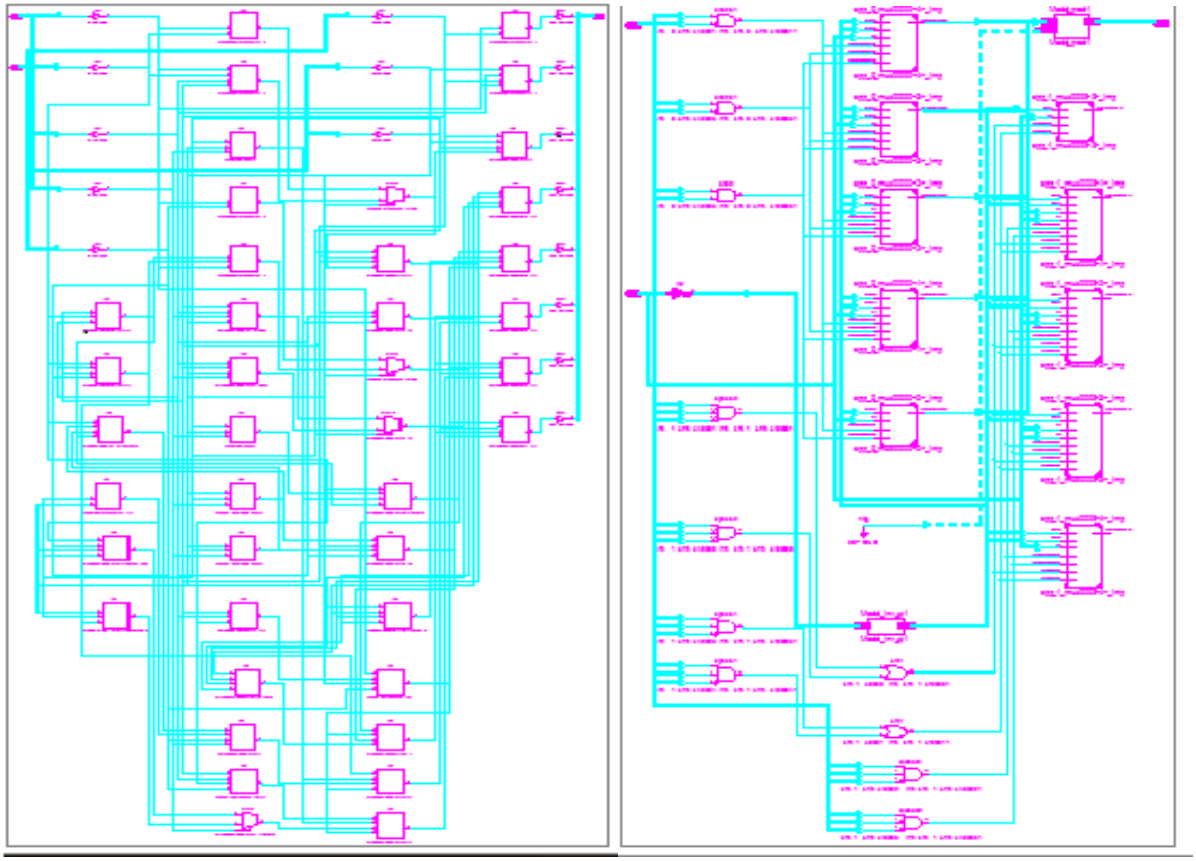


Fig:-9 improved booth multiplication of 0111 x 0111

5.1 Rtl Schematic



VI. CONCLUSION

We proposed a newly efficient booth multiplier with improved capability to do its multiplication not only to unsigned bits but also to signed bits .Because of having unique identity of multiplying both signed and unsigned bits ,the outcome of the multiplication techniques will cause decay to the maximum extent of the partial product array.

REFERENCES

[1].AbenetGetahun, ―Booth Multiplication Algorithm,‖ Fall 2003 CSCI 401.
[2]. Jean-Luc Beuchat, Jean-Michel Muller, ―Multiplication Algorithms for Radix-2 RN-Codings and Two's Complement Numbers.
[3] Susrutha Babu Sukhavasi., Suparshya Babu Sukhavasi., "Low Power Synthesis In Digital Design By Automatic Insertion Of Clock Gating And Operand Isolation Cells**",** International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 2, pp.1086-1092., Mar-Apr 2012., ISSN: 2248-9622.
[4]. Stephen Brown|ZvonkoVranesic, ‖Fundamentals of Digital Logic Degin with VHDL‖, Tata McGraw-Hill .
[5]. Fabrizio Lamberti, Nikolaos Andrikos, Elisardo Antelo, Paolo Montuschi, ―Speeding-up Booth Encoded Multipliers by Reducing the Size of Partial Product Array**,**" Internal Report Dauin/Delen-Politecnico Di Torino And University De Santiago De Compostela, 2009 .
[6]Premananthan.G, Amudha.K, Sreenath.G, ― A Faster Carry Save Adder In Radix-8 Booth Encoded Multiplier,‖ International Journal of VLSI & Signal Processing Applications, Vol. 2,Issue2,April 2012, ISSN 2231-3133, (171-175).
[7].Susrutha Babu Sukhavasi., Suparshya Babu Sukhavasi," Implementation Of Low Power Parallel Compressor For Multiplier Using Self Resetting Logic"**,** International Journal of Computer Applications (IJCA), Vol.47 – No. 3, pp. 47(3):27-32, June 2012, ISSN: 0975 – 888.
[8]. CMOS Logic Circuit Design 2002 John P Uyemura.
[9]. Susrutha Babu Sukhavasi., Suparshya Babu Sukhavasi., " Implementation Of Synchronous Up Counter By Using Self Resetting Logic"**,** International Journal of Engineering Research and Applications (IJERA),**.**, Vol. 2, Issue 1, pp.972-981., Jan-Feb 2012., ISSN: 2248-9622.
[10]. M. Lehman and N. Burla, ―Skip techniques for highspeed carry Propagation in binary arithmetic units**,** IRE Trans. Electron.comput., vol. EC-10, pp. 691–698, Dec. 1962.
[11]. Suparshya Babu Sukhavasi, Susrutha Babu Sukhavasi, "Analysis and Comparison of combinational Circuits by using low Power Techniques"**,** International Journal of Advances in Engineering & Technology (IJAET), Vol. 3, No.1. ISSN: 2231-1963
[12]. Susrutha Babu Sukhavasi, Suparshya Babu Sukhavasi," Power Competent Cmos Comparator For Analog To Digital Converter Circuits", International Journal of Advances in Engineering & Technology (IJAET), Nov. 2013. Vol. 6, Issue 5, pp. 2196-2210. ISSN: 2231-1963

AUTHORS

S.Susrutha Babu was born in India, A.P, in 1987. He received the B.Tech degree from JNTU, A.P, and M.Tech degree from SRM University, Chennai, Tamil Nadu, India in 2008 and 2010 respectively. He worked as Assistant Professor in Electronics Engineering in Bapatla Engineering College for academic year 2010-2011 and from 2011 to till date working in K L University. His research interests include antennas, FPGA Implementation, Low Power Design and wireless communications. He had published in various international journals and 4 IEEE conferences

S.Suparshya Babu was born in India, A.P, in 1987. He received the B.Tech degree from JNTU, A.P, and M.Tech degree from SRM University, Chennai, Tamil Nadu, India in 2008 and 2010 respectively. He worked as Assistant Professor in Electronics Engineering in Bapatla Engineering College for academic year 2010-2011 and from 2011 to till date working in K University. His research interests include antennas, FPGA Implementation, Low Power Design and wireless communications. He had published in various international journals and 4 IEEE conferences

Shaik Jani Bashawas born on 20th Dec, 1994 in Sattenapalli, Guntur district, Andhra Pradesh India. He is undergraduate student from Electronics and Communication Engineering from K L University, Guntur with specialization in VLSI Design.

M.V.S Sai Nikhil was born on 17th Jul, 1995 in Hyderabad, Rangareddy district, Telangana, India. He is undergraduate student from Electronics and Communication Engineering from K L University, Guntur with specialization in Speech Processing.

M.Aravind was born on 2nd Aug, 1995 in Guntur, Guntur district, Andhra Pradesh, India. He is undergraduate student from Electronics and Communication M.V.S Sai Nikhilwas born on 17th Jul, 1995 in Hyderabad, Rangareddy district, Telangana, India. He is undergraduate student from Electronics and Communication Engineering from K L University, Guntur with specialization in Speech Processing.

.