ISSN (Online): 2230-7893
www.IJCEM.org

High Speed and Reduced Power – Radix-2 Booth Multiplier

Sakshi Rajput[1], Priya Sharma[2], Gitanjali[3] and Garima[4]

[1,2,3,4]Asst. Professor, Deptt. of Electronics and Communication,
Maharaja Surajmal Institute of Technology, New Delhi, India.

Abstract

A multiplier is one of the key hardware blocks in most digital and high performance systems such as FIR filters, digital signal processors and microprocessors etc. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the whole system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. So here idea is to find out the best trade off solution among the both of them. The result of this work helps to make a proper choice among different adders in booth multiplier that is used in different digital applications according to requirements.

Keywords: *radix-2 booth multiplier, ripple carry adder, modified carry select adder, binary to excess-1 converter*

1. Introduction

As the slope for integration keeps increasing, signal processing systems are being implemented on a VLSI chip widely. These signal processing applications not only demand for great computing ability but also utilize significant amount of energy. While performance and Area remain to be the two critical design goals, power consumed by the VLSI system design has become a critical concern. [1]. Multiplication is the most fundamental arithmetic operation used in most of the signal processing algorithms. But multipliers usually have large area, larger delay and consume more power. Therefore low-power multiplier design holds a significant part in low- power VLSI system design. There has been a lot of work being done in the field of designing low-power multipliers at technology, physical, circuit and logic levels [1]. Power refers to number of Joules dissipated over a certain amount of time whereas energy is the measure of the total number of

Joules dissipated by a circuit. In digital CMOS design, power-delay product factor is important it is generally used to calculate the merits of designs. Multiplication consists of three steps: partial products generation (PPG), partial products reduction (PPR), and last is carry-propagate addition (CPA). In general there are sequential and combinational multiplier implementations but combinational case will be considered here because the scale of integration is large enough to consider parallel multiplier implementations in digital VLSI systems. The area occupied and the time delay consumed by different adders and to found out a proper relation between time and area complexity of all the adders under consideration are to be studied. On the basis of area and delay, hence to choose the best adder for appropriate situation. In Multipliers the approach used to study different multipliers by writing verilog codes, verifying waveforms and then finally calculating number of CLBs, LUTs required along with Power consumed in the circuit. After knowing all this, calculated the delay for different multipliers which helped to determine the best multiplier.

2. Normal Booth Multiplier

In many real-time DSP applications, high performance is a critical concern. Multiplication is the most fundamental arithmetic operation used in most of the signal processing algorithms. But multipliers usually have large area, larger delay and consume more power. However, achieving this may be done at the cost of area, on chip power consumed and delays. In the binary number system the digits, called bits, are limited to the set {0, 1}. The result of multiplying any binary number by a binary bit is either 0, or the original number. This makes formation of the intermediate partial-products simple and efficient. Adding all these partial-products is time consuming task for any binary multipliers. The entire process consists of three steps partial product generation, partial product reduction and addition of partial products as shown in Fig 1. But in booth multiplication, partial product generation is done based on recoding scheme e.g. radix 2 encoding. Bits of multiplicand (Y) are grouped from left to right and corresponding operation on multiplier (X) is done in order to generate the partial product. In radix-2 booth multiplication partial product generation is done based on encoding which is as given by Table1. Parallel

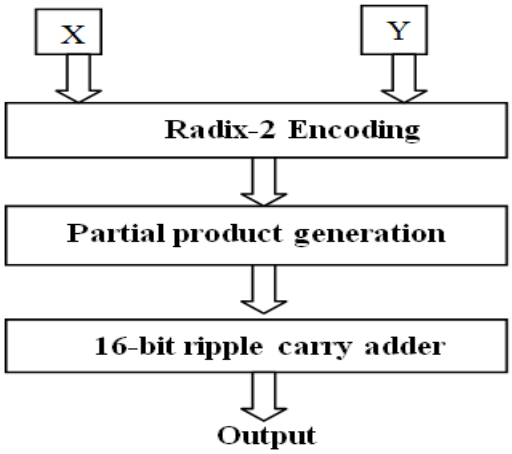


Fig. 1 Flow chart for normal Booth Multiplier

Multiplication using normal Booth's recoding algorithm technique based on the fact that partial product can be generated for group of consecutive 0's and 1's which is called Booth's recoding. This recoding algorithm is used to generate efficient partial product. These partial products always have large number of bits than the input number of bits. This increase in the width of partial product usually depends upon the radix scheme used for recoding. So, these scheme uses less partial product generation which in turn provides low power and area but in the Normal Booth multiplier Ripple Carry Adder is used shown in Fig 2.

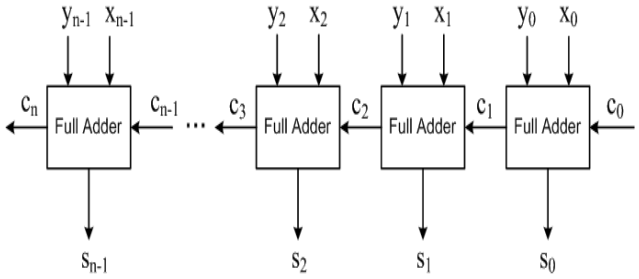


Fig. 2 Block Diagram of RCA

Drawbacks of using Ripple Carry Adder:

1 It is not efficient when large numbers of bits are used.

2 Carry propagation delay increases linearly with bit length as next stage output is dependent on previous stage output.

Logic equations:-

$C_i = x_i \& y_i$ (1)
$P_i = x_i \wedge y_i$ (2)
$S_i = P_i \wedge C_i$ (3)
P_i=*Partial product*
C_i=*Carry bit*

Recoding scheme used in radix-2 booth multiplier is shown in the Table 1.

Table 1 Recoding Table for Booth Multiplier

Q_n	Q_{n+1}	Recoded Bits	Operation
0	0	0	Shift
0	1	+1	Add X
1	0	-1	Subtract X
1	1	0	Shift

Hardware implementation of Booth Algorithm requires the register configuration. So named the Multiplier and Multiplicand as registers 'A','B and 'Q' as AC,BR as shown in Complete flow chart 2.An extra flip flop is Q_{n+1} is added to provide a double bit inspection of the multiplier. The complete fig 3 has shown condition basis implementation of the Radix-2 Booth Algorithm.

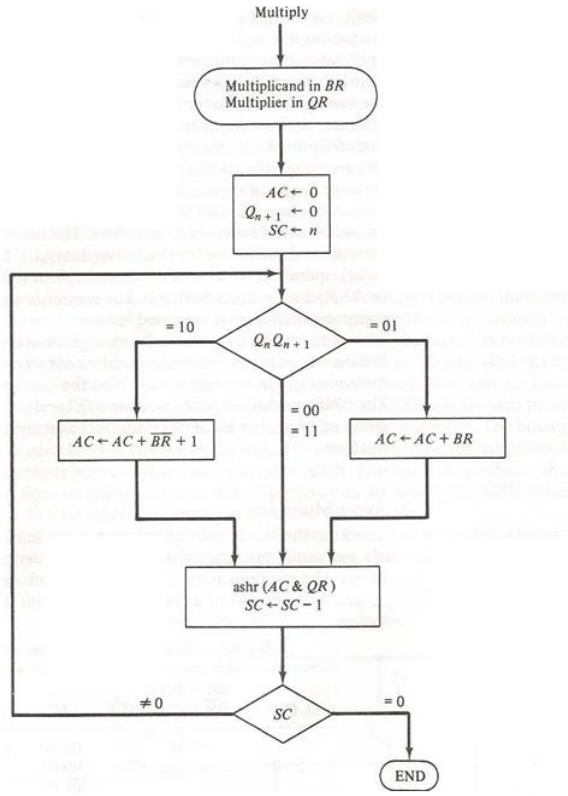


Fig. 3 Complete flow chart of normal booth multiplier

Before writing the code for the Analysis first of all, State chart is to be defined. On the basis this chart, the coding part has been done in verilog. This State chart helps to understand the basic steps involved in booth multiplier algorithm. Hence it helps to understand the coding part easily an also reduces the complexity of understanding the code as shown in Fig 4.

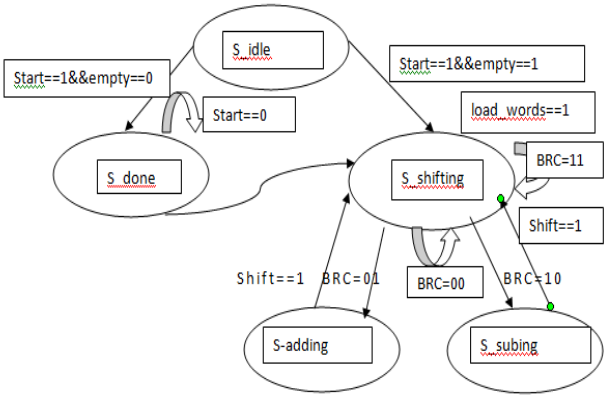


Fig. 4 State chart of the Normal Booth Multiplier

3. Booth Multiplier Using Bitwise CSA

In this multiplier instead of using ripple carry adder, Bitwise Carry Select adder is used in order to achieve speed maximization. The addition of partial products is carried out by using CSA (Carry Select Adder). Carry Select adder is used shown in Fig 5 in order to reduce the carry propagation delay that is very high in case of ripple carry adder conventionally used by multipliers. Selecting the sum and carry, bit by bit here. Hence it requires more circuitry. The logic equations for Bitwise carry Select Adder are:-

Logic Equations

$Sa_i = a_i \wedge b_i$ (4)

$Sb_i = \sim(a_i \wedge b_i)$ (5)

$S_i = c_i ? Sb_i : Sa_i$ (6)

$C_i = (a_i \& c_i) | (a_i \& b_i) | (b_i \& c_i)$ (7)

Where Sa_i shows the partial sum with no carry and Sb_i shows partial sum with carry as '1'. S_i is final sum bit and C_i as final carry bit.

Drawbacks:

1 Because of multiplexers larger area is required.

2 Have a lesser delay than Ripple Carry Adders.

3 Hence Carry Select Adder is preferred while working with smaller no of bits.

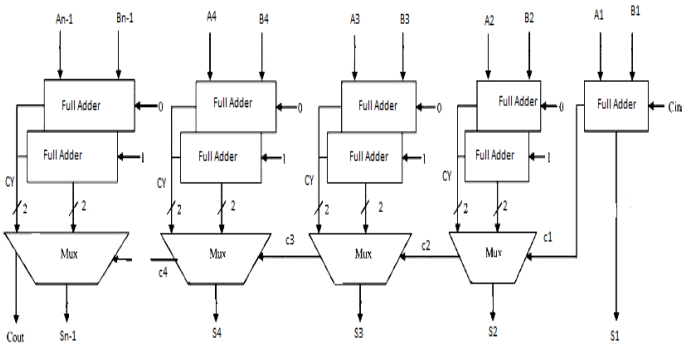


Fig. 5 Block Diagram of Bitwise CSA

4. Booth multiplier Using Square root CSA

This multiplier is same as that using Bitwise CSA instead we are using square root CSA. It uses same logic as that used in booth multiplier using bitwise CSA, instead of using bitwise CSA, Square root CSA is used in this scheme. In square root CSA is shown in Fig 6 has blocks of ripple carry adder with input more than one bits instead of single bit as in case of bitwise CSA.

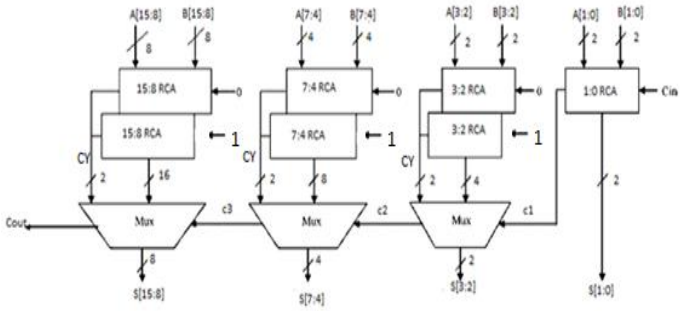


Fig. 6 Block Diagram of Square root CSA

Drawbacks:

Multiplexer requires more area. Hence this circuit uses more area but less than that of Bitwise Carry Select Adder.

5. Multiplier using Proposed CSA

In the proposed CSA instead of using ripple carry adder with input carry as 1, Binary to Excess-1 Converter (BEC) shown in Fig 7 is used in the proposed CSA in order to reduce the area occupied by the square root CSA. Here, for the addition of partial products proposed CSA is used. For n-bit ripple carry adder we will use n+1 bit BEC. Fig 7 shows the Block diagram for BEC converter.

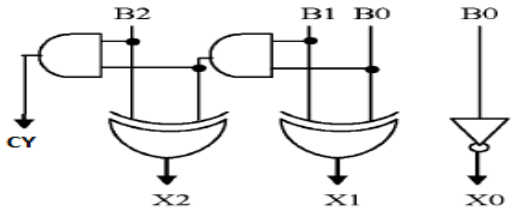


Fig. 7 Block Diagram of BEC Converter

Logic equation for Binary to Excess-1 Converter:

$X0 = \sim B0$ (8)

$X1 = B1 \wedge B0$ (9)

$X2 = B2 \wedge (B1 * B0)$ (10)

$CY = B0 \& B1 \& B2$ (11)

Where CY is the carry bit and the whole system work as the adder with carry bit as '1'.

In the binary to excess-1 converter carry select adder the least significant bits are added using conventional RCA, while other blocks are added in parallel along with the given BEC. Once all the interim sums and carries of blocks are calculated, the final sums are computed using multiplexers having minimal delay. The multiplexer block as shown in Fig 8, receives the two sets of input and selects the final sum based on the selected input from the previous stage.

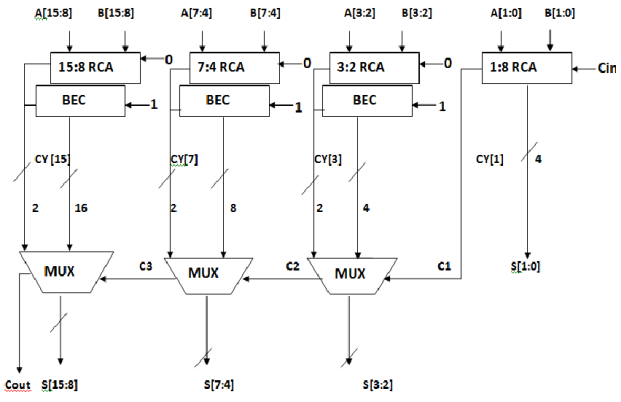


Fig. 8 Block Diagram of Proposed CSA

The importance of the BEC logic emerges from the large silicon area reduction when the CSLA with large number of bits are designed. Use of BEC with multiplexer thus achieves fast incrementing action with reduced device count. Thus, the proposed CSA excels the conventional CSA circuit in terms of area and power.

In this type of multiplier, proposed CSA is used for the reduction of the carry propagation delay of Normal booth multiplier. Basic concept used is same that of multiplier using square root CSA the only difference is that instead of using ripple carry adder block with the input carry as 1, BEC is designed as separate unit to implement in this. Partial product so generated is added using proposed root CSA.

6. Simulation and Implementation Results

6.1 Simulation results

Simulation is the imitation of the operation of a real-world process or system over time. The act of simulating something first requires that a model be developed; this model represents the key characteristics of the selected physical or abstract system or process.

The model represents the system itself, whereas the simulation represents the operation of the system over time. In this waveform, the input variables word1 and word2 of 8 bit with certain key conditions of setting 'start' and 'clk' and correspondingly their output which is stored in register of 16 bit named product.

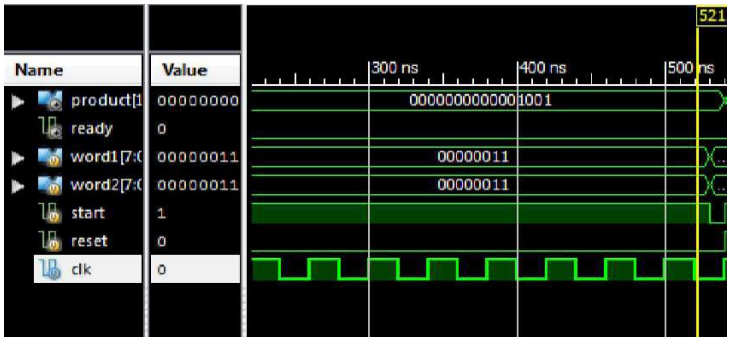


Fig. 9 Simulation Result of Proposed Booth Multiplier

6.2 Comparison of Multipliers

In this research, comparison of different multiplier on the basis of their speed and power parameters has been done. Xilinx ISE version 12.1 is used for simulation of different multipliers and knowing their delays. Table 2 to Table 6

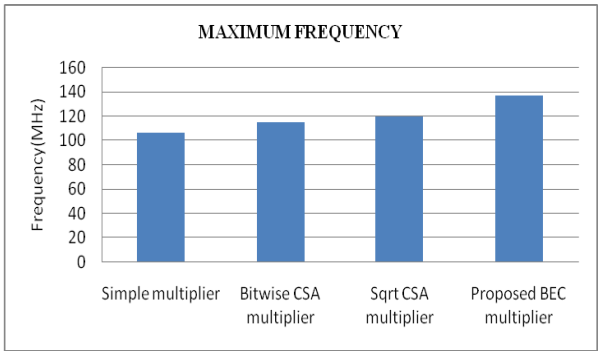
shows the comparison results for various multipliers with their respective Graphs from no. 1 to 5.

Maximum frequency comparison Table 2 depicts the maximum frequency of each multiplier used in this analysis. More is the frequency lesser will be the delay as shown. Frequency is observed in Mega Hertz. From this table it has been noticed that maximum frequency is more for the Proposed Binary to Excess-1 converter Multiplier having 136.874 MHz while other multipliers have lesser maximum frequency.

Table 2: Comparison of multipliers on the basis of Maximum Frequency

Maximum frequency (MHz)			
Simple multiplier	**Bitwise CSA multiplier**	**Sqrt CSA multiplier**	**Proposed BEC multiplier**
106.530	115.367	119.933	136.874

Graph 1 show the Comparison chart of multipliers for maximum frequency as frequency is calculated from the minimum time period of the clk covered for critical path in the design system. Here this Graph 1 concludes that maximum frequency value is for proposed BEC multiplier that is 136.874 MHz. This graph represents the Y- axis reads frequency in MHz and X- axis tells about the type of Multiplier considered by blue columns. Simple multiplier has 106.530 MHz of Maximum frequency, Bitwise CSA multiplier has 115.367 MHz and Sqrt CSA multiplier has 119.933 MHz of Maximum frequency.



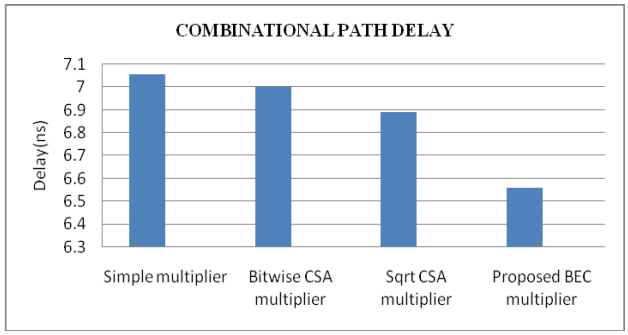
Graph 1 Comparison Chart on the basis of maximum frequency

Comparison of Multipliers on the basis of combinational path delay shows in Table 3 shows that combinational circuitry logic path covered for specific time period combinational path Delay is calculated in nano second or 10^{-9} sec. From this table it has been noticed that Combinational path Delay is minimum for the proposed Binary to Excess-1 converter Multiplier having 6.56 ns while other multipliers have more delay.

Table 3: Comparison table on the basis of Combinational path delay

Combinational path Delay(ns)			
Simple multiplier	**Bitwise CSA multiplier**	**Sqrt CSA multiplier**	**Proposed BEC multiplier**
7.055	7.0	6.89	6.56

Graph 2 shows the comparison between four designed multipliers using different adders on the basis of combinational path delay, which shows the overall delay in the whole logic. Combinational path delay is the way to know the speed maximization achieved as desired. It depicts the delay for the combinational logic circuitry in design. Here this Graph 2 concludes that Combinational path Delay value is for Proposed BEC multiplier that is 6.56 ns. This graph represents the Y- axis reads Delay in 10^{-9} sec and X- axis tells about the type of Multiplier considered by blue columns. Simple multiplier has 7.055 ns of Combinational path Delay, Bitwise CSA multiplier has 7.0 ns and Sqrt CSA multiplier has 6.89 ns of Combinational path Delay.



Graph 2 Comparison Chart on the basis of Combinational path Delay

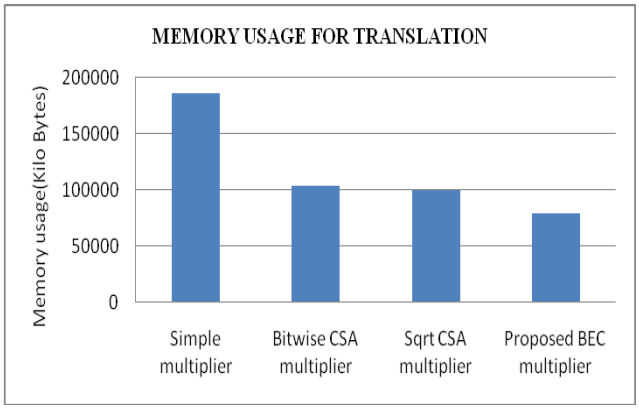
Memory usage depicts the amount of memory occupied by the whole logic as shown in Table 4. Memory usage for translation is calculated in Kilo Bytes. From this table it has been noticed that Memory usage for translation is minimum for the proposed Binary to Excess-1 converter Multiplier having 79772 kb while other multipliers have more Memory usage.

Graph 3 shows the Comparison Graph of multipliers for memory usage for translating the design into the hardware RTL and technological logic. Here this Graph 3 concludes that Memory usage for translation value is for Proposed BEC multiplier that is 79772 kb. This graph represents the

Y- axis reads Memory usage in Kilo Bytes and X- axis tells about the type of Multiplier considered by blue Columns. Simple multiplier has 186360 kb of Memory usage for translation, Bitwise CSA multiplier has 103584 kb and Sqrt CSA multiplier has 100032 kb of Memory usage for translation.

Table 4: Comparison on the basis of Memory Usage

MEMORY USAGE FOR TRANSLATION(Kilo Bytes)			
Simple multiplier	**Bitwise CSA multiplier**	**Sqrt CSA multiplier**	**Proposed BEC multiplier**
186360	103584	100032	79772



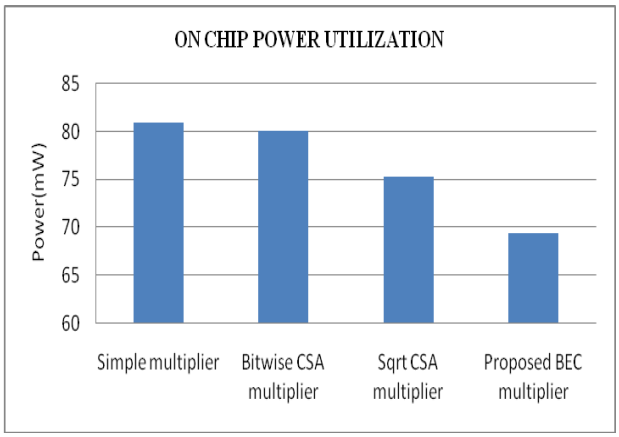
Graph 3 Comparison Chart on the basis of memory usage

On chip power utilization gives the total information about the net power consumed in the whole logic as shown in Table 5 and Graph 4 shows the comparison on the basis of on chip power utilization. On chip power utilization is calculated in milliwatts. From this table it has been noticed that power utilization is minimum for the Proposed Binary to Excess-1 converter Multiplier having 69.45 mW.

Table 8: Comparison on the basis of On Chip Power Utilization

ON CHIP POWER UTILIZATION(mW)			
Simple multiplier	**Bitwise CSA multiplier**	**Sqrt CSA multiplier**	**Proposed BEC multiplier**
80.98	80.1	75.37	69.45

Graph 4 shows that on chip power utilization value for Proposed BEC multiplier that is 69.45 mW. This graph represents the Y- axis reads Power in milliwatts and X- axis tells about the type of Multiplier considered by blue Columns. Simple multiplier has 80.98 mW of On chip power utilization, Bitwise CSA multiplier has 80.1 mW and Sqrt CSA multiplier has 75.37 mW of on chip power utilization.



Graph 4 Comparison chart on the basis of on chip power utilization

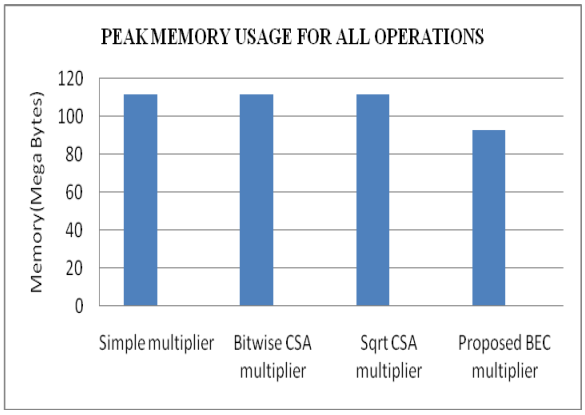
Peak memory usage is the maximum amount of memory logic can occupy in Megabytes as shown in Table 6. From this table it has been noticed that Peak memory usage for all operations is minimum for the proposed multiplier having 93 Mb while other multipliers have more Memory usage.

Table 6: Comparison on the basis of Peak Memory Usage

PEAK MEMORY USAGE FOR ALL OPERATIONS(Mega Bytes)			
Simple multiplier	**Bitwise CSA multiplier**	**Sqrt CSA multiplier**	**Proposed BEC multiplier**
112	112	112	93

It shows the Comparison chart of multipliers for Peak memory usage for all operations to implement the logic into hardware design, mapping, routing i/o definition etc. Here this Graph 5 shows that Peak memory usage for all operations value is for Proposed BEC multiplier that is 93Mb. This graph represents the Y- axis reads Memory in Mega Bytes and X- axis tells about the type of Multiplier considered by blue Columns. Simple multiplier, Bitwise CSA multiplier and Sqrt CSA multiplier has 112 Mb of Peak memory usage for all operations respectively.
From the above comparison it is clear that Proposed BEC is suitable for low power high speed optimization. It

requires less memory usage among all and also less power. Hence here is the desired goal of this research work.



Graph 5 Comparison Chart on the basis of peak memory usage

7. Conclusion

After going through all the hard work and facing problems, this review managed to complete its objectives that are to implement Radix-2 Booth Multiplier using different adders and learn the Power and Time trade off among them to design **Efficient Faster Low Power Multiplier**.
Different adders are compared by using critical parameters like Delay, Power, and Area etc. to make clear ideas of, which adder was best suited for situation. After comparing all, it was concluded that Carry Select Adders are best suited for situations where Speed is the critical concern. Coming to Multipliers ,implementation of Radix-2 Booth Multiplier is done using different adder and came to final conclusion that parallel multipliers are much better than the serial multipliers due to less area consumption and hence the less power consumption. So after making these comparisons with the use of different types of adder through the comparison graphs it can be inferred that Radix-2 booth multiplier using proposed CSA that is using BEC is more power-efficient than the other three and has much less delay, which is the prime concern for any VLSI industry.

Acknowledgement

Team effort together with precious proper n mindful guidance makes daunting tasks achievable. It is a pleasure to acknowledge the direct and implied help we have received at various stages while developing the project. It would not have been possible to develop such a project without the assistance of numerous individuals. It takes a great privilege to thank Mr. S.P. Singh, Chairman, Maharaja Surajmal Institute of Technology(MSIT); Dr. Navin Kapur ,Director, MSIT; Dr. V. Kumar (HOD,ECE Dept. ,MSIT) for giving us a great opportunity to carry out this research and our heartfelt thanks to our mentor Ms. Sakshi Rajput (Astt. Professor, ECE Dept, MSIT) whose generous support at all stages of this research helped us achieve the desired objectives.

References

[1] Parth Sarthy Mohanty ,"Design and implementation of low power fast multipliers", thesis report NIT Rourkela, 2009.
[2] B. Ramkumar, H.M. Kittur, and P. M. Kannan, "ASIC implementation of modified faster carry save adder," *Eur. J. Sci. Res., vol. 42, no. 1, pp. 53–58, 2010.*
[3] K.S.Rao, M.Reddy, A.V.Babu and P.Srinivasulu, "Energy and area efficient carry select adder," *IJERA, vol. 34, issue 2, pp. 436-440, Mar-Apr 2012.*
[4] Y.Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," *Electron. Lett., vol. 37, no. 10, pp. 614–615, May 2001.*
[5] Z.Huang, "High-Level Optimization Techniques for Low-Power Multiplier Design," *PhD dissertation, Univ. of California, Los Angeles, June 2003*
[6] M. Morris mano: Computer system architechture Prentice Hall Date: 1992- 10-29,pp 344
[7] Uma Nirmal, Geetanjali Sharma, Yogesh Misra," A Low Power High Speed Adders using MTCMOS Technique", *International Journal of Computational Engineering and Management IJCEM, Vol. 13, July 2011*.

Sakshi Rajput supervises a successful research group consisting of graduate students conducting experiments in the field of VlSI design and implementations in at Maharaja Surajmal Institute of Technology affiliated to Guru Gobind Singh Indrapastha University. She received B.E. (Hons) degree in Electronics and Communication from University of Rajasthan,Jaipur and M.Tech(Hons) degree in VLSI Design from Guru Gobind Singh Indraprastha University, Delhi . Her research interest includes Low power VLSI design. Currently she is the Assistant Professor in Electronics and Communications Dept., Maharaja Surajmal Institute of Technology, Guru Gobind Singh Indraprastha University, New Delhi, India. She has published 4 books on VLSI in international publication.

Priya Sharma is currently pursuing B.tech (4th year) in Electronics and Communications from Maharaja Surajmal Institute of Technology affiliated to Guru Gobind Singh Indraprastha University, New Delhi, India.

Gitanjali is currently pursuing B.tech (4th year) in Electronics and Communications from Maharaja Surajmal Institute of Technology affiliated to Guru Gobind Singh Indraprastha University, New Delhi, India.

Garima is currently pursuing B.tech (4th year) in Electronics and Communications from Maharaja Surajmal Institute of Technology affiliated to Guru Gobind Singh Indraprastha University, New Delhi, India.