IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 12, December 2021

DOI: 10.17148/IARJSET.2021.81241

Modified booth Wallace multiplier using MSQRT Carry Select Adder with Common Boolean Logic

Aastha Gupta[1], Dr. Ravi Sindal[2]

ME(DC), Electronics &Communication Department , IET DAVV, Indore[1]

Associate Professor, Electronics &Communication Department, IET DAVV, Indore[2]

Abstract: Multiplier is one of the key blocks in most of the high performance and digital systems such as the microprocessors, digital signal processors & FIR filters. This paper presents a model of 8-bit multiplier i.e. Radix -8 booth Wallace multiplier using common Boolean logic. Radix 8 Booth algorithm is responsible to generate the partial products. Wallace tree structure using 3:2 compressors is responsible for the reduction and accumulation of partial products to two rows. Whereas the SQRT CSLA with Common Boolean logic is responsible to add the last two rows of partial products. The use of compressors in Wallace tree structure and common Boolean logic in the adder results in the reduction of overall delay and power consumption of the multiplier. A Verilog code has been written which is synthesized in Xilinx ISE 14.7 software and simulated in ModelSim. After simulation the performance of this multiplier is compared with the modified booth wallace multiplier using carry save adder & is found that the power and delay of the proposed multiplier is reduced with the increase in the number of gate count.

Keywords: Booth Encoder, Wallace tree, Compressors, MSQRT CSLA, CBL.

1. INTRODUCTION

In digital signal processing (DSP), multiplier is one of the basic functions which is used. Multiplier is the one which requires more hardware resources and the time of processing than any other arithmetic operations such as addition and subtraction. The central processing unit in computers have to give a considerable amount of time to implement arithmetic operations, particularly multiplication. Many of the high-performance systems depend on the multiplication for achieving high data throughput. Currently many multiplication-based operations are implemented in the digital signal processing applications such as Fast Fourier Transform (FFT), convolution, filtering etc. Since

the execution time of most DSP algorithm is dominated by the multiplier, so high speed multiplier is required.
In the past, many novel ideas for multipliers have been proposed to achieve high performance. The demand for high speed processing has been increasing as a result of expanding computer and signal processing applications. Many researchers have proposed various structure of the multipliers. For many applications, reducing the time delay and power requirement are very essential requirements. It is seen that on reducing multiplier delay we have to compromise with its area and power consumption. There is a need to design a multiplier which provides better performance in terms of area, power and delay.
In the proposed 8-bit multiplier design, radix 8 booth algorithm is used for the generation of partial products. Using Radix-8 the number of partial products is reduced to n/3.Then Wallace tree structure is implemented which will reduce these partial products to two rows. Wallace tree structure is designed using 3:2 compressors which will contribute in reducing the delay of the multiplier. Then, the two rows of the multiplier is finally added by modified square root carry select adder using common Boolean logic (CBL). This adder is used in place of carry save adder which will further reduce the delay and power consumption of the multiplier.

2. LITERATURE SURVEY

The Booth multiplier uses booth algorithm. It is an algorithm for the multiplication of two signed binary numbers in 2's complement form. The algorithm is based on recoding the multiplier bits. The two main drawbacks of booth algorithm are the inefficiency of the circuit when isolated 1's are present and difficulty in designing parallel multipliers as shift and add operations may vary. Hence modified booth algorithm was developed. This algorithm reduces the number of partial product rows to (n+2)/2 where n is no. of bits. It includes radix-4 , radix-8 and radix-16 booth multipliers.

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 12, December 2021

DOI: 10.17148/IARJSET.2021.81241

The other multiplier is Wallace tree multiplier. Half adders, full adders, compressors are used in this multiplier to produce two rows of partial products which are finally added using suitable adder. But its operation is limited to signed integers alone.

Based on the parameter's performance of booth and Wallace tree multiplier, booth encoded Wallace tree multiplier is designed. It has the advantage of both booth and Wallace tree multiplier.

Any suitable adder can be used for the addition of two rows. Ripple carry adder has the least area and longest delay. Carry look ahead adder is one of the fastest adder but consume very large area. Carry select adder provides good trade-off between area and speed.

3. PROPOSED MODEL

The proposed model of 8-bit modified booth Wallace multiplier using MSQRT CSLA with CBL is given below. It consists of four major modules: Booth encoder, partial product generator, Wallace tree and MSQRT CSLA.

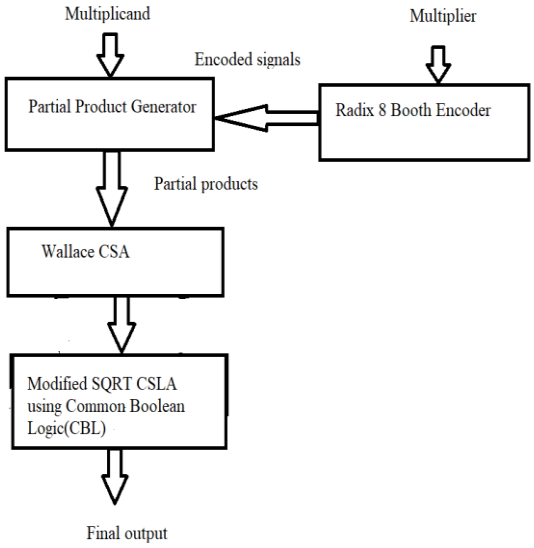


Fig 1: Block diagram of Radix 8 booth Wallace multiplier using MSQRT CSLA

3.1 BOOTH ENCODER

Booth algorithm consists of booth recoding which results in the reduction of multiplier bits. Different booth recoding algorithms have been proposed. These are Radix-2, Radix-4, Radix-8, Radix-16 and Radix-32. Radix-2 is conventional booth algorithm whereas Radix-4, Radix-8, Radix-16 and Radix-32 are modified booth recoding algorithm. In radix-4, 3 multiplier bits are paired and encoded which reduce the no. of bits to n/2. In radix-8, 4 bits are paired and encoded which reduce the no. of bits to n/3.Radix means the no. of operations that can be performed. For ex- Radix-4 can perform 4 operations: +1, -1, +2, -2. Radix-8 can perform 8 operations: +1, -1, +2, -2, +3, -3, +4, -4. Generally, radix-4 and radix-8 are mostly preferred because as we move towards higher radix the circuit complexity also increases. In radix-8 the no. of partial products are reduced to n/3. So, only n/3 partial products are generated. The reduction in partial product results in the reduction of delay of the multiplier. Radix-4 produces more no. of partial products than radix-8. So, radix-8 is used in this paper.

3.2 PARTIAL PRODUCT GENERATOR

Partial product generator generates partial products by multiplying the multiplicand with one bit of the encoded multiplier. Partial product generation is the intermediate step of finding the final product. It performs the operations like 0y,1y, -1y,2y, -2y,3y, -3y,4y, -4y where y is multiplicand. Multiply by 0 results in 0. Multiply by 1 result in same number as multiplicand. Multiply by -1 means the product is the two's complement form of the multiplicand. Multiply by 2 means shift left the multiplicand by one place. Multiply by -2 means shift left one bit the 2's complement of multiplicand value. Multiply by -4 means shift left two bit the two's complement of multiplicand. Multiply by 4 means just shift left the multiplicand by two bits. For 3y, we need to perform 2y+y i.e. to add the number with the same number shifted one position to the left.

IARJSET
ISSN (O) 2393-8021, ISSN (P) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 8, Issue 12, December 2021
DOI: 10.17148/IARJSET.2021.81241

3.3 WALLACE TREE STRUCTURE

Wallace tree architecture is used to reduce the no. of partial products and to accumulate the partial products to two rows. Wallace tree method uses 3 steps to process the multiplication operation.

i. Formation of Partial products.
ii. Reduction of partial products to two rows.
iii. Addition of two rows of partial products.

Conventional Wallace tree structure consists of Half adders and Full adders. Half adders are used where there are 2 bits and full adder is used where there are 3 bits in each column. If full adders are replaced with the compressors then the addition stages of partial products can be reduced. It provides the advantage of accumulation of partial products at an expense of least possible power dissipation. The simplest and the most widely used compressor is the 3:2 compressor. The 3:2 compressor make use of carry save adder. It has 3 inputs and provides two outputs. The 3 numbers can be reduced to 2 by doing the addition while keeping the carries and the sum separate. i.e. addition is performed in parallel without relying on the result of previous column. The sum and carry can be recombined in a normal addition to form the correct result. The compressor is governed by the basic equation-

x1+x2+x3=sum+2*carry

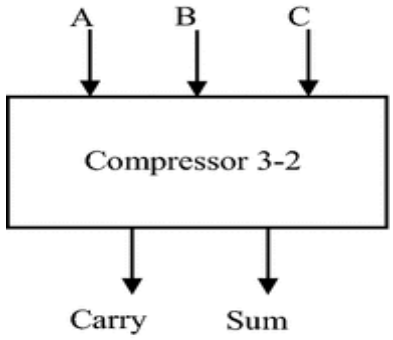


Fig 2: Block diagram of 3:2 compressor (CSA)

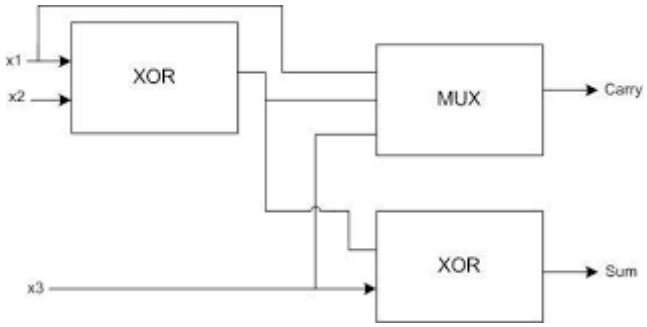


Fig 3: XOR and MUX based 3:2 compressor

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 12, December 2021

DOI: 10.17148/IARJSET.2021.81241

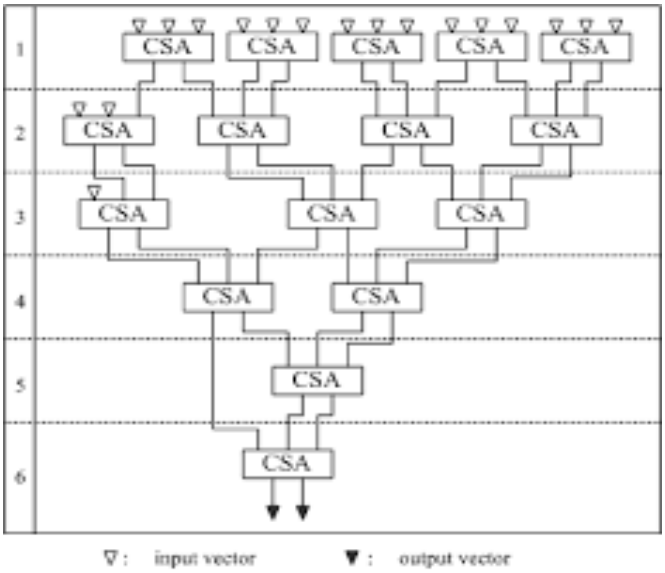


Fig 4: Wallace tree using 3-2 compressor

3.4 MSQRT CSLA with CBL

To remove the drawback of Ripple carry adder (RCA), Carry select adder (CSLA)is presented. The Carry Select Adder generally consists of Ripple Carry adder and a MUX. Two RCA are used in CSLA architecture to perform the calculation twice, first with Cin as zero and second with Cin as 1. After two results are calculated, the MUX selects correct sum and Cout. However, the CSLA is not area and speed efficient due to pair of RCAs'. So, MSQRT CSLA is proposed to overcome the drawback of CSLA. It contains different size groups of RCAs' which provides parallel path for carry propagation to reduce the overall delay of the adder.

The MSQRT CSLA design is further improved and now one of the RCA out of the two is replaced with Common Boolean Logic (CBL). The CBL circuit is given in fig 5.

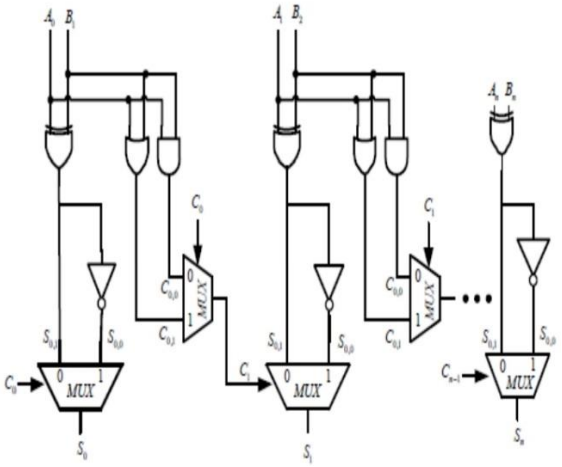


Fig 5: Common Boolean Logic circuit

To generate a summation pair, we only need to use one XOR gate and one INV gate. For the carry propagation path, need one OR gate and one AND gate to anticipate possible carry input values in advance. The use of CBL logic in MSQRT CSLA reduces the power delay product. Thus, this adder

when used as a final adder gives better performance than other types of adder in terms of power and delay but it is not area efficient.

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 12, December 2021

DOI: 10.17148/IARJSET.2021.81241

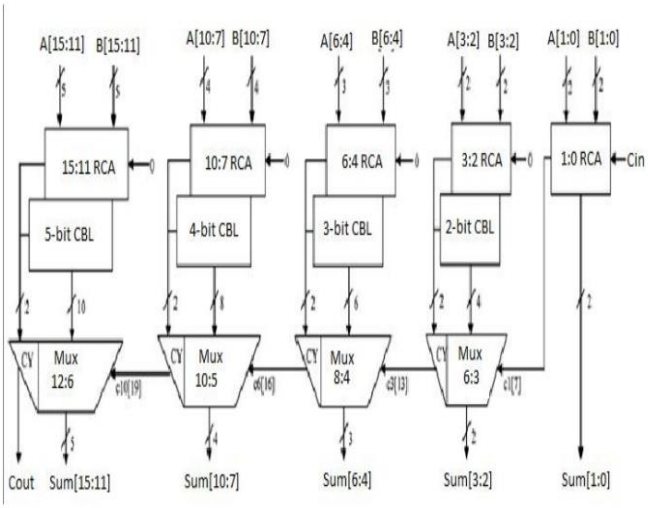


Fig 6: Block diagram of MSQRT CSLA with CBL

4. MODEL ANALYSIS

The above proposed multiplier design is synthesized using Xilinx ISE 14.7 software. The area, power and delay of this multiplier is analysed as shown in Fig. 7, 8 and 9 respectively. The simulation is performed using ModelSim. The RTL schematic of the proposed multiplier is given in Fig.10 and Fig.11 and the simulation result is shown in Fig.12.Table 1 shows the comparison results of the Modified booth Wallace multiplier using carry save adder and the Radix-8 booth Wallace multiplier using MSQRT CSLA with CBL.

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	213	3,840	5%
Logic Distribution			
Number of occupied Slices	118	1,920	6%
Number of Slices containing only related logic	118	118	100%
Number of Slices containing unrelated logic	0	118	0%
Total Number of 4 input LUTs	223	3,840	5%
Number used as logic	213		
Number used as a route-thru	10		
Number of bonded IOBs	32	97	32%
Number of MULT18X18s	2	12	16%
Total equivalent gate count for design	9,617		
Additional JTAG gate count for IOBs	1,536		

Fig 7: Area analysis

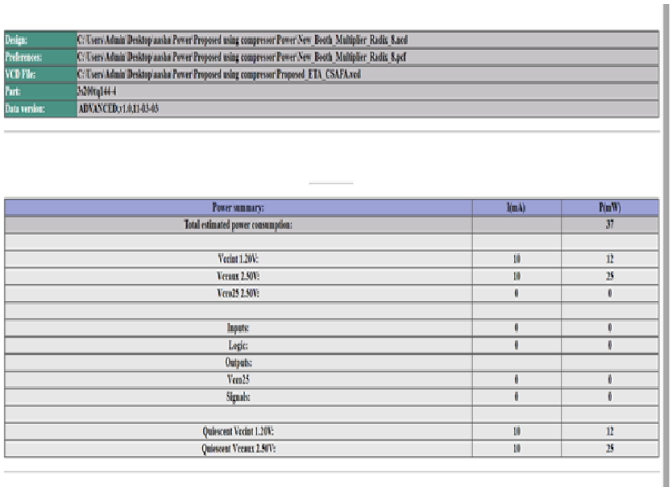
IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 12, December 2021

DOI: 10.17148/IARJSET.2021.81241



Design:	C:\Users\Admin\Desktop\aasha Power\Proposed using compressor\Power\New_Booth_Multiplier_Radix_8.ncd
Preferences:	C:\Users\Admin\Desktop\aasha Power\Proposed using compressor\Power\New_Booth_Multiplier_Radix_8.pcf
VCD File:	C:\Users\Admin\Desktop\aasha Power\Proposed using compressor\Proposed_ETA_CSAFA.vcd
Part:	3s500eq144-4
Data version:	ADVANCED,v1.0,11-03-05

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		37
Vccint 1.20V:	10	12
Vccaux 2.50V:	10	25
Vcco25 2.50V:	0	0
Inputs:	0	0
Logic:	0	0
Outputs:		
Vcco25	0	0
Signals:	0	0
Quiescent Vccint 1.20V:	10	12
Quiescent Vccaux 2.50V:	10	25

Fig 8: Power analysis

```
LUT4:I1->O        2  0.551   0.903 final adder/M4/Cout1 (final adder/Carry<4>)
LUT4:I3->O        2  0.551   1.216 final adder/M5/Cout1 (final adder/Carry<5>)
LUT4:I0->O        2  0.551   1.216 final adder/M6/Cout1 (final adder/Carry<6>)
LUT3:I0->O        2  0.551   1.216 final adder/M7/Cout1 (final adder/Carry<7>)
LUT3:I0->O        2  0.551   1.216 final adder/M8/Cout1 (final adder/Carry<8>)
LUT3:I0->O        2  0.551   1.216 final adder/M9/Cout1 (final adder/Carry<9>)
LUT3:I0->O        2  0.551   1.216 final adder/M10/Cout1 (final adder/Carry<10>)
LUT3:I0->O        2  0.551   1.216 final adder/M11/Cout1 (final adder/Carry<11>)
LUT3:I0->O        2  0.551   1.216 final adder/M12/Cout1 (final adder/Carry<12>)
LUT3:I0->O        2  0.551   1.072 final adder/M13/Cout1 (final adder/Carry<13>)
LUT3:I1->O        1  0.551   0.801 final adder/M14/S1 (P_14_OBUF)
OBUF:I->O             5.644       P_14_OBUF (P<14>)
----------------------------------------
Total                 7.165ns (6.364ns logic, 0.801ns route)
                      (88.82% logic, 11.17% route)
```

Fig 9: Delay analysis

5. SIMULATION RESULTS

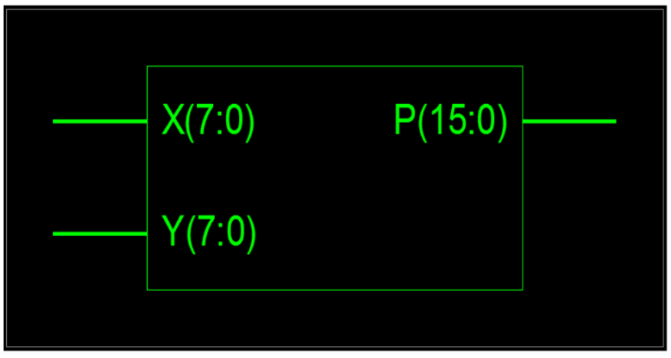


Fig 10: RTL schematic 1 of multiplier

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 12, December 2021

DOI: 10.17148/IARJSET.2021.81241

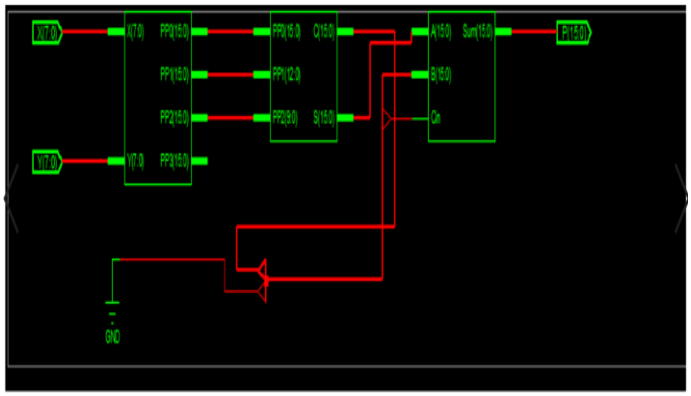


Fig 11: RTL Schematic 2 of multiplier

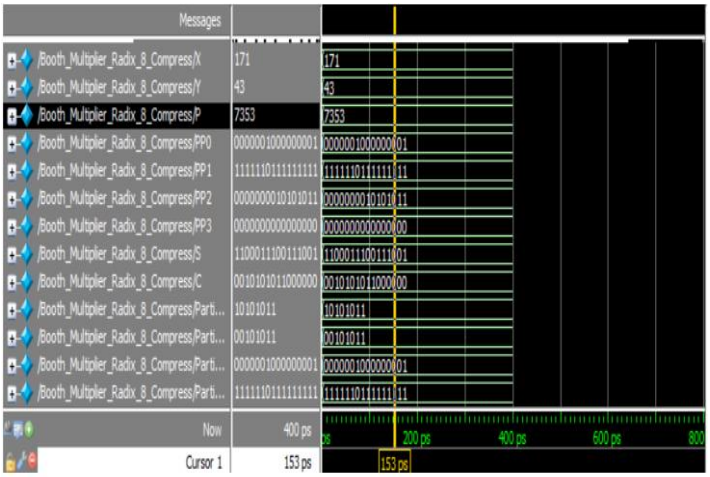


Fig 12: Simulation result of multiplier

Table 1
Comparison of multipliers

Multiplier	Area	Power	Delay
Modified booth Wallace multiplier using carry save adder	5103	288mw	8.413ns
Modified booth Wallace multiplier using MSQRT CSLA	9617	37mw	7.165ns

5. CONCLUSION

This paper presents a modified booth Wallace multiplier using MSQRT CSLA with CBL. In the existing model, carry save adder is used for final addition. Whereas in the proposed multiplier, the modified square root carry select adder using common Boolean logic has been used. Also, the Wallace tree structure is replaced by modified structure. i.e.3:2 compressors are used in Wallace tree structure. These two changes in the existing model resulted in the reduction in delay

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 12, December 2021

DOI: 10.17148/IARJSET.2021.81241

and power consumption with slight increase in the area. The delay of the proposed model is reduced to 7.16 ns and the power consumption is reduced to 37mw.

So, the proposed multiplier is giving better performance in terms of speed and power with the compromise in terms of area. Hence, the proposed 8-bit Radix 8 booth Wallace multiplier can be used for high speed and low power applications.

6. ACKNOWLEDGEMENT

I would like to express my gratitude to Dr. Sanjeev Tokekar, director IET-DAVV Indore. A special thanks to my guide, Dr. Ravi Sindal whose encouragement helped me to finish my work.

7. REFERENCES

[1] Vikas Kaushik and Himanshi Saini. 'A review on comparative performance analysis of different digital multipliers'. Advances in computational sciences and technology. ISSN 0973-6107, Volume 10, 1257-1272. 2017.

[2] K.Sindhuja, C.Thiruvenkatesan. 'Design of low power approximate radix 8 booth multiplier'.International journal of Engineering Research & Technology(IJERT).ISSN:2278-0181,Issue 2017.

[3] Jasbir kaur, Sumit Kumar. 'Performance comparison of higher Radix booth multiplier'. International journal of Innovative Research in Science, Engineering and Technology.vol.5, ISSN (0):2319-8753 Issue 1, January 2016.

[4] Priya Meshram, Prof. Mamta Sarode. 'Design of modified area efficient square root carry select adder (SQRT CSLA)'. International journal of Industrial Electronics and Electrical Engineering. ISSN:2347-6982, Issue-4 June 2015.

[5] Himanshu Bansal, K.G. Sharma, Tripti Sharma. 'Wallace Tree multiplier designs: A performance comparison Review'.Innovative Systems Design and Engineering. ISSN 2222-2871(0), vol.5, No.5,2014.

[6]K.Gopi Krishna, B.Santhosh. 'Design of Wallace tree multiplier using compressors', International Journal of Engineering Sciences& Research Technology. ISSN:2277-9655, September, 2013.

[7] Athira Koranath, Sonali Agrawal.'Comparison of different multiplier algorithms and 1D-DWT as an application'.IOSR Journal of Electronics and Communication Engineering(IOSRJECE). ISSN:2278-2834, Volume1, Issue 1(May-June 2012).