

Implementation of Double Precision Floating Point Multiplier Using Wallace Tree Multiplier

Y. Ramya sri ¹, V B K L Aruna ²

P.G. Student, Department of Electronics Engineering, V.R Siddhartha Engineering College, kanuru, Vijayawada, India¹
Assistant Professor, Department of Electronics Engineering, V.R Siddhartha Engineering College, kanuru, Vijayawada, India²

ABSTRACT: Multipliers are frequently used in DSP, Image processing and microprocessor applications. This paper describes an implementation of double precision floating point multiplier using Wallace tree Multiplier which is targeted on Virtex-6 using Verilog HDL. The floating-point arithmetic unit is used in most General Purpose Processors (GPP) and Application Specific Processors (ASP) due to its wide range and precise number system. As the time for addition of generated partial products is reduced, the multiplier speed can be increased. To multiply 52-bit mantissa in Double precision Floating point and to reduce the count of partial products obtained in a multiplication process Wallace tree multiplier is used. In this paper Wallace tree is constructed with the help of compressor techniques such as 4:2 compressor, 5:2 compressor, half adders and full adders for reducing the delay of multiplier.

KEYWORDS: Double Precision, Floating Point Multiplier, FPGA, Wallace Tree Multiplier, Compressor techniques.

I. INTRODUCTION

The computer arithmetic units in modern microprocessors execute advanced applications such as 3D graphics, multimedia, signal processing and various scientific computations which requires complex mathematics. The binary fixed-point number system is not sufficient to handle such complex computations. In IEEE-754 Standard, a notation for binary floating point is defined. It represents a wide range of numbers from tiny fractional numbers to extremely huge numbers. The floating-point numbers consists of three parts (sign-bit, exponent and Mantissa) so that the operations which may require complex procedures. For example, the operations frequently require the normalization, which causes an increased logic delay. Therefore, improving the performance of floating-point operations has been a research topic in the computer arithmetic field.

Sign 1-bit	Exponent 11-bits	Mantissa 52-bits
---------------	---------------------	---------------------

Fig. 1. IEEE-754 Double Precision Floating Point Format

Fig. 1 shows the IEEE-754 double precision binary Floating point format representation. Sign (S) is represented with one bit, exponent (E) is represented with 11-bits and fraction (M or Mantissa) is represented with fifty two bits respectively.

$$Z = (-1)^s * 2^{(E-Bias)} * (1.M) \quad (1)$$

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

Equation (1) represents scientific notation of Floating point numbers. ‘S’ represents Sign bit which indicates sign of the number. Mantissa represents magnitude of the number. Exponent represents number of places the fixed point is to be shifted. Bias value is 1023 in double precision which is based on $2^{(N-1)}$ -1. Here ‘N’ represents number of exponent bits. Multipliers are complex units which play an important role in area, delay, speed and power consumption of digital system design. Designing multipliers which are having high-speed, low power, and regular in layout are of substantial research interest. Multipliers play an important role in today’s digital signal processing and various other applications. With advances in technology, many researchers have tried to design multipliers having high speed, low power consumption, regularity in layout.

This paper presents implementation double Precision Floating Point Multiplier using Wallace Tree Multiplier. For real-time signal processing, a high speed and low power Multipliers-Accumulator (MAC) is required. It plays a prominent role to obtain better performance in the digital signal processing systems. The main advantage of MAC is to increase speed of multiplier. High speed can be achieved through well-known Wallace tree multiplier [1]. The number of adding stages can be reduced by using this algorithm.

II. RELATED WORK

There are different ways to implement Wallace tree multiplier. One of them is to consider the first four bits in a column and generating two bits by using 4:2 compressors [2]. Another way is to consider the first five bits in a column by using 5:2 compressors [3]. This paper presents 52 by 52 bit Wallace Tree Multiplier. Multiplying 52-bit Mantissa in Double Precision Floating Point format by using different types of compressors like 4:2 and 5:2. Suppose two numbers are being multiplied:

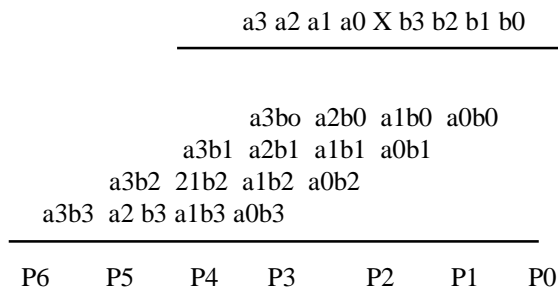


Fig. 2. Multiplication of two numbers

After multiplication process partial products will be generate as shown in above figure. Those partial products will be arranged in the form of Tree Structure according to their weights shown below. After arrangement of the weights those will be added by using compressors.

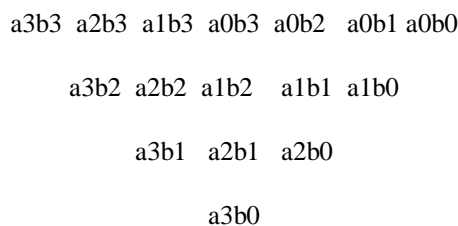


Fig. 3. Arranging Weights in Wallace Tree Structure

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

III. FLOATING POINT MULTIPLICATION ALGORITHM

Floating point numbers are one of the possible ways of representing real numbers in binary format. The IEEE 754 standard presents two floating point formats, Binary interchange format and Decimal interchange format. Multiplying floating point numbers plays an important role in DSP applications. This paper mainly focuses on double precision floating point binary interchange format [5]. It consists of a one bit sign (S), an eleven bits exponent (E), and a fifty two bits fraction (M or Mantissa).

The algorithm for multiplying floating point numbers is as follows:

1. Sign is obtained by XOR operation ; i.e. $S_a \text{ xor } S_b$.
2. Adding the exponents; i.e. $(E_1 + E_2 - Bias)$.
3. Multiplying the significand i.e. $(1.M_1 * 1.M_2)$.
4. Placing the decimal point in the significand result.
5. Normalizing the result; i.e. the MSB of the results significand is 1.
6. Rounding the result to its nearest value.
7. Checking for overflow or underflow occurrence.

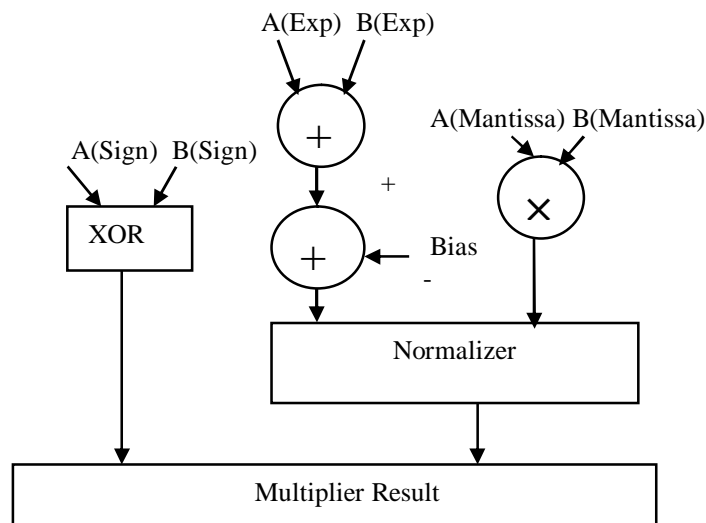


Fig. 4. Floating point Multiplier structure

IV. WALLACE TREE MULTIPLIER

Wallace tree multiplier was first invented by Chris Wallace, an Australian scientist in 1964. It is the hardware implementation of a digital circuit which multiplies two integers. The partial products can be added by using these compressors. Hence the partial product count can be reduced by using this multiplier. Hence it is efficient.

Fig.5. Shows block diagram of floating point multiplier having sign bit which can be calculated by using XOR operation. Two exponents are added by using 11-bit ripple carry adder and 52-bit Mantissa's are multiplied by using Wallace tree multiplier. The result of Wallace tree is 104-bit product which can be normalized to 52-bit by using normalization.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

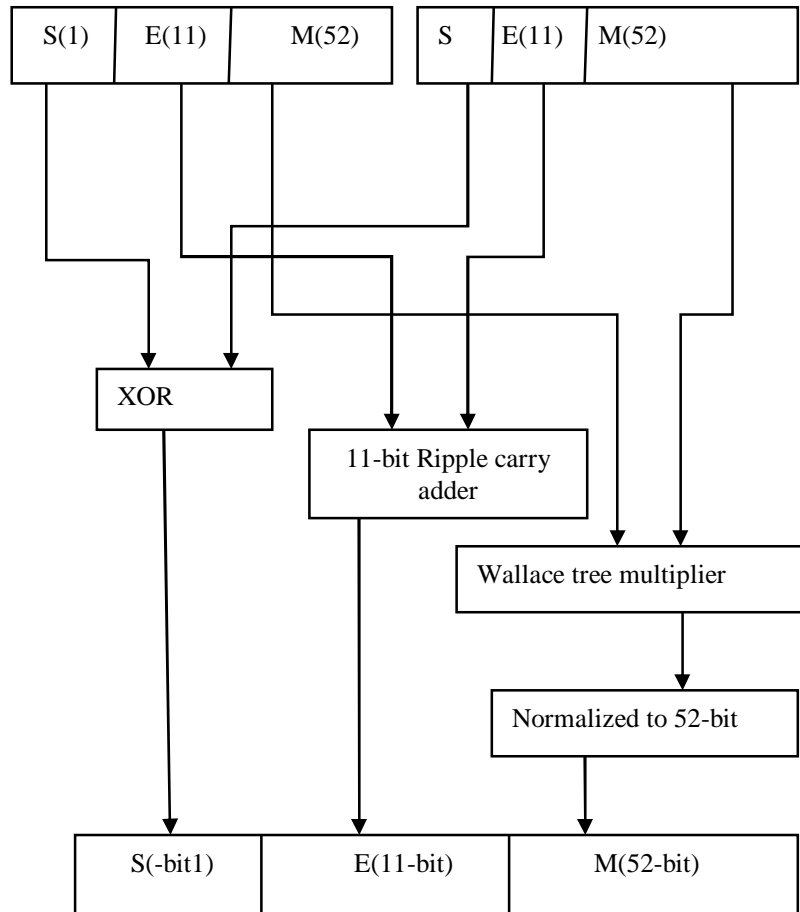


Fig. 5. Block diagram of double precision floating point multiplier using Wallace tree multiplier

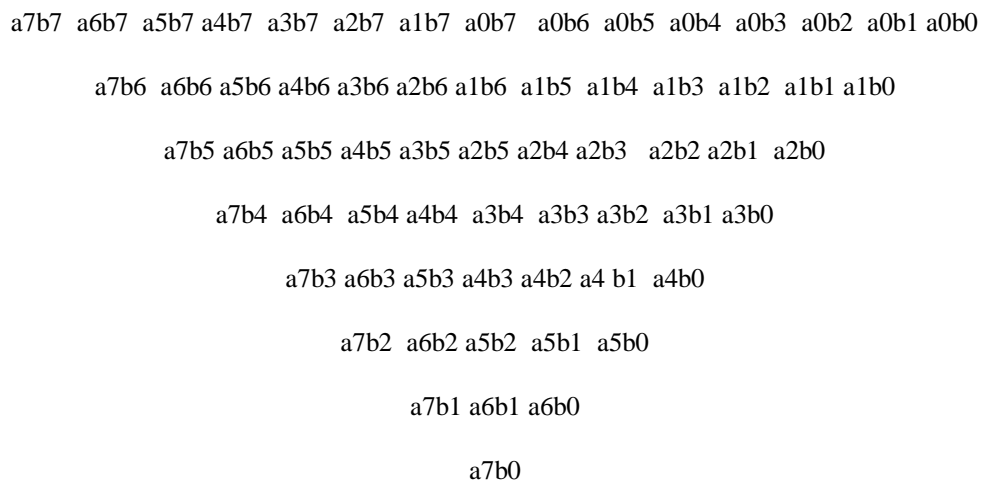


Fig. 6. 8X8 Wallace tree multiplier Structure

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

The calculation time of the multipliers can be minimized by using different compressor structures. The computation time of the Wallace tree has achieved the lower bound of $O(\log_3/2 N)$. The maximum number of steps required is $(\log_3/2(n/2) + 1)$ for an n-bit Wallace multiplier. Fig. 6 shows how partial products are arranged in the form of tree structure according to weights.

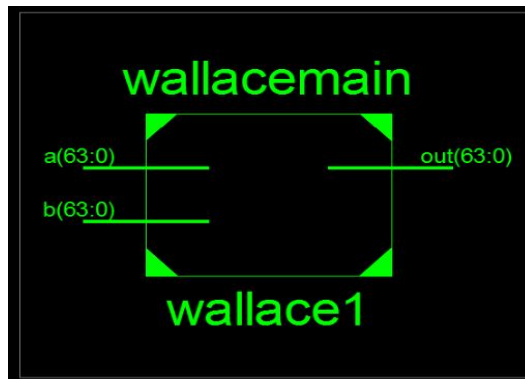


Fig. 7. Black box view of Wallace Tree Multiplier

Wallace tree has three steps:

1. Each bit of multiplier is multiplied with same bit position of multiplicand. The generated partial products have different weights depending on the position of multiplier bits
2. The number of partial products are minimized to two by using compressors.
3. After step 2 we get two rows of sum and carry. Add these two rows with half adders.

Explanation for second step:

- a. Take any three rows of same weight and align them into a full adder. The result will be an output row of same weight.
- b. If there are two rows of the same weight then input them into a half adder.
- c. If there is just one row then connect it to the next layer.

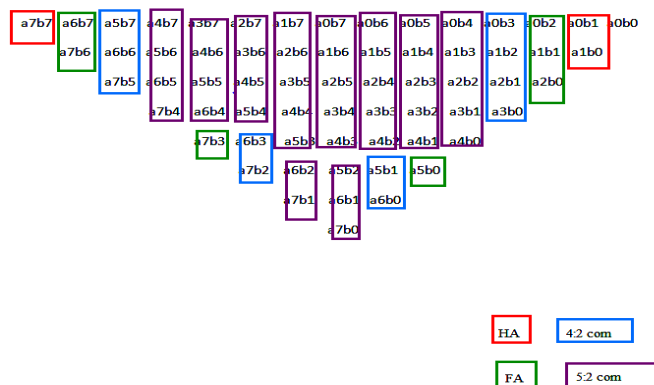


Fig. 8. 8X8 Wallace tree Multiplier using Compressors

The above figure explain about how the partial products are arranged according to their weights and how they are added using different compressors.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

V. COMPRESSORS

Compressors are arithmetic components similar to parallel counters, but having two different differences:

1. They have explicit carry-in and carry-out bits.
2. There may be some redundancy among the ranks of the sum and carry output bits.

4:2 COMPRESSORS:

A 4-2 compressor consists of five inputs and three outputs. So it compresses four partial products into two output bits. This can be implemented with two stages of full adders (FA) which are connected in series as shown in Fig.9.

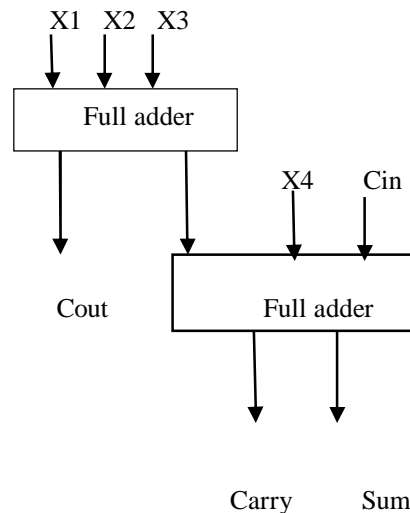


Fig. 9. 4:2 Compressor using Full Adders

Fig 9. represents the architecture of 4:2 compressor by using two serially connected full adders. The four inputs X1, X2, X3 and X4 and the output Sum are from same column of the partial product. Cin is the output carry of the previous stage and Cout is the carry output of the current stage compressor fed as Cin to the next compressor stage [2]. The compressor is characterized by the Equation (2).

$$X1+X2+X3+X4+Cin= Sum+2(Carry + Cout) \quad (2)$$

TABLE I. Truth Table for 4:2 Compressors

X1	X2	X3	X4	Cin	Cout	Sum	Carry
0	0	0	1	1	0	0	1
1	1	1	0	0	1	1	0
0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0
1	0	0	0	1	0	0	1
0	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

5:2 COMPRESSORS:

Fig 10. represents 5:2 compressor having five inputs X1, X2, X3, X4, X5 and two carry inputs Cin1 and Cin2 and produces 4 outputs like Sum, Carry, Cout1 and Cout2 [3] . The input carry bits are the outputs from the previous block of compressor and the output carry's are given to the successive stage of compressor. This can be implemented by using three stages of full adders which are connected in series as shown in fig. 10.

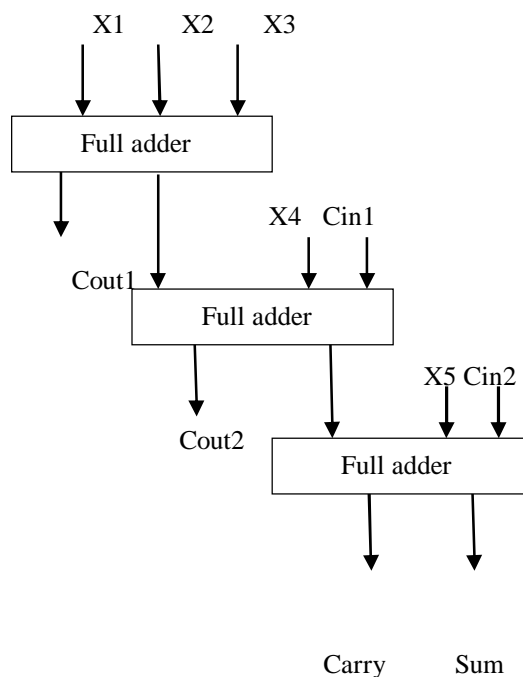


Fig. 10. 5:2 Compressor Using Full Adders

5:2 compressor operations is expressed by Equation (3)

$$X1+X2+X3+X4+X5+Cin1+Cin2=Sum+2. (Carry+Cout1+Cout2) \tag{3}$$

TABLE II. Truth Table for 5:2 Compressors

X1	X2	X3	X4	X5	Cin1	Cin2	Cout1	Cout2	Sum	Carr y
0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	1	1	1	0
1	1	1	0	1	0	0	1	1	0	0
1	1	1	0	0	0	0	1	0	1	0
1	1	0	0	0	1	0	1	0	1	0
1	1	1	0	0	1	1	1	0	1	1
1	1	1	1	1	1	1	1	1	1	1

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

TABLE III. Parameters for Wallace tree Multiplier

Parameters	Wallace tree multiplier
Area(LUT's)	6319
Delay(ns)	121.154
Throughput(1/ns)	0.00825

The multiplication operations is implemented by using wallace tree multiplier which yields different parameters like area, delay and Throughput which are shown in above table.

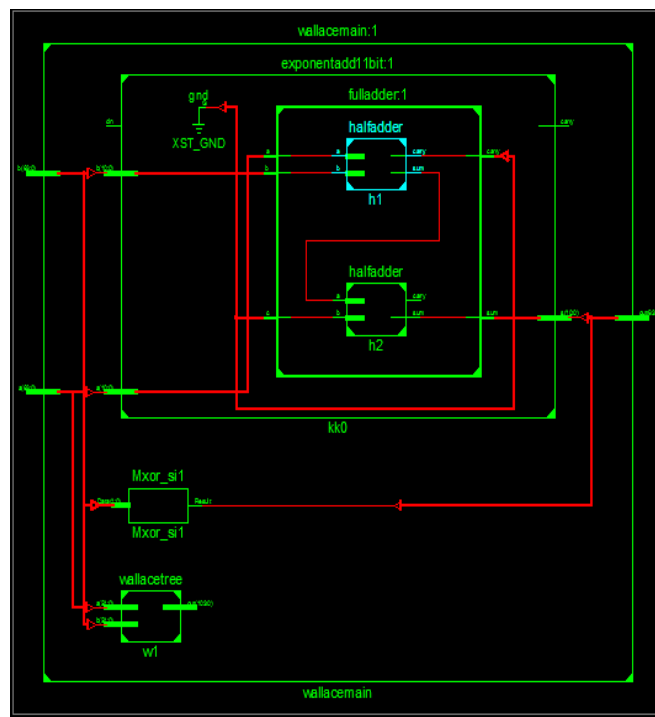


Fig. 11. RTL Schematic of Double Precision Floating Point multiplier using Wallace Tree

V. RESULTS

The previous sections have discussed design for floating-point Multiplier using Wallace tree multiplier. This is implemented in Verilog-HDL targeted on a Xilinx Virtex-6 device. The functionality of the implementations is verified by performing a simulation with some random input vectors. In order to evaluate designs, the area, critical path latency, throughput, and power consumption are verified. The result for the design in double precision implementation are shown in Table 3.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

Fig. 12 shows simulation results for double precision floating point Wallace tree multiplier .A and B are inputs of 64 bits each. Each input having sign bit ,11 bit exponent and 52 bit mantissa. Sign bit can be calculated by XOR operation and the result is stored in `si` and two exponent bits are added together and result is stored in `sumexp[11:0]`. The two mantissa's are multiplied by using Wallace tree multiplier and the result is stored in `prod[52:0]`.

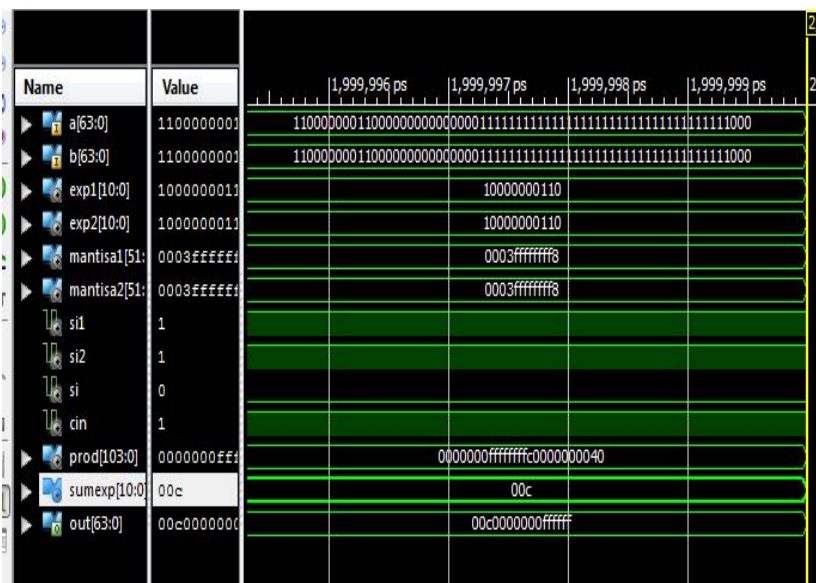


Fig.12. Simulation results of double precision floating point multiplier

VI. CONCLUSION

The implementation of Double Precision Floating point Multiplier using Wallace Tree multiplier is presented which is targeted on Xilinx Vertex-6 xc6vlx75t-3ff484 FPGA. It provides high speed and supports double precision. This design can handle the overflow, underflow, and truncation and rounding modes.

REFERENCES

- [1] Naveen Kr. Gahlan , Prabhat Shukla and Jasbir Kaur "Implementation of Wallace Tree Multiplier using Compressor", Vol 3 (3), 1194-1199 .
- [2] Riya Garg, Suman Nehra, B. P. Singh "Low power 4-2 Compressor for Arithmetic circuits", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-2, Issue-1, March 2013.
- [3] Jagadeshwar Rao M, Sanjay Dubey "A High Speed Wallace Tree Multiplier Using Modified Booth Algorithm for Fast Arithmetic Circuits", IOSR Journal of Electronics and Communication Engineering (IOSRJECE) ISSN: 2278-2834, ISBN No: 2278-8735 Volume 3, Issue 1 (Sep-Oct 2012).
- [4] Addanki Purna Ramesh, A. V. N. Tilak, A.M.Prasad "An FPGA Based High Speed IEEE-754 Double Precision Floating Point Multiplier using Verilog", IEEE 2013.
- [5] Addanki Purna Ramesh, Rajesh Pattimi "High Speed Double Precision Floating Point Multiplier", International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 9, November 2012.