

An Efficient VLSI Design of 32X32 bit Multiplier using Wallace Tree Algorithm in Vivado HLS and Xilinx ISE Software using VHDL



Sunkana Rama Krishna, Arun Kishore Voleti, Jagadeesh Pasumarthi, Yelusuri Deepika, Rachakonda Sai Kiran

Abstract: Multiplier is the most basic component present in any digital system. These multipliers are mainly used in Digital Signal and Image Processing applications. In applications like image detection latest sophisticated algorithms like CNN are used which contains MAC units in their design. The multiplier used in MAC unit requires huge memory, offers high latency and consumes more power. There are many algorithms such as Combinational, Sequential and Array Multiplication Algorithms which helps in designing Multiplier. The major drawback in all designs is circuit complexity. The problem of latency and power dissipation are also present. Considering all the drawbacks present in those algorithms this paper proposes the usage of Wallace Tree Algorithm which consumes less power and has low latency. Also, there are many ways to add the final stage of partial products generated such as Carry Look Ahead adder, Carry Select Adder etc. This paper uses both Carry Select Adder and Ripple Carry Adder for performing final addition of partial products. All previous partial products are added using Half adders and Full adders. The Multiplier is designed using VHDL in Xilinx ISE and Vivado Platform.

Keywords: Carry Select Adder, Multiplier, Ripple Carry Adder, Vivado HLS, Wallace Tree Algorithm, Xilinx ISE.

I. INTRODUCTION

In the past decade, there is a rapid growth in the field of low power VLSI design. There is a huge research carried out to make a design offers low latency. Multiplication is a very common and vital operation performed in many high-end applications. The most possible option is to design a system with all components which are individually efficient in design. Starting from 1947 onwards a huge research is carried out on designing an efficient multiplier.

Revised Manuscript Received on May 30, 2020.

* Correspondence Author

Arun Kishore Voleti*, Department of ECE, LIET, Vizianagaram, India, Email: arunkishorevoleti@gmail.com

Jagadeesh Pasumarthi, Department of ECE, LIET, Vizianagaram, India, Email: jagadeeshpc5@gmail.com

Deepika Yelusuri, Department of ECE, LIET, Vizianagaram, India, Email: deepika4y@gmail.com

Sai Kiran Rachakonda, Department of ECE, LIET, Vizianagaram, India, Email: sk9160530227@gmail.com

S. Rama Krishna, Associate Professor, Department of ECE, LIET, Vizianagaram, India, Email: slkrishna444@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

[6] Different multipliers have different power consumption values and different delay. A Multiplier is a vital component in the design of any Digital Signal Processing and Digital Image Processing units. The fast-growing field in present technological world is Space Exploration. In order to perform image detection sophisticated algorithms like Convolution Neural Networks are used. The vital unit present in CNN is MAC unit. Each layer of CNN is made up of numerous MAC units. A MAC unit stands for Multiplier and Accumulator. The entire device efficiency depends completely on MAC unit performance.

Multiplier governs the performance of MAC unit. The efficiency of entire MAC unit depends on how efficient the Multiplier is. So, in order to improve efficiency of MAC unit which implicitly increases the efficiency of many high-end applications, this paper describes the efficient implementation of a Multiplier unit. The focus of this paper is to optimize a multiplier in terms of power and delay. Starting from the design of Full Adder, everything matters a lot to decrease latency.

There are many Multiplication algorithms [4], some of them are: Combinational Multiplication Algorithm, Sequential Multiplication Algorithm, Array Multiplication Algorithm, Baugh Wooley Multiplication Algorithm, Booth Algorithm, Wallace Tree Algorithm, Vedic Multiplication Algorithm etc. This paper used Wallace Tree Algorithm to design efficient circuit to achieve desired values.

Any Multiplier follows three basic steps. Different multipliers follow different approaches to perform these three stages. The three basic stages are Acquiring Partial Products, Adding Partial Products to reduce stages and Final Addition of Partial Products.

II. LITERATURE SURVEY

Reference [1] has discussed about pipelining and parallel processing, retiming, folding, folding and Systolic Architecture Design in VLSI. Chapter-13 in this book has made a clear discussion on Different Multiplication Techniques. A power and time efficient designs were introduced in the in this book. Appendix E has made a brief discussion on Fast Binary Adders and Multipliers in which we have Wallace Tree Multiplier Algorithm.

Reference [2] has discussed about problems in large binary multiplications. This paper concluded that for large binary values, sequential multipliers shows more performance over combinational multipliers. A detailed comparison on characteristics of five different multipliers were shown in this paper.

III. ADDER CIRCUITS

Adders are very essential components of a Multiplier. The multiplication of two binary numbers cannot be done directly. First, each bit in number-1 must be multiplied with each bit in number-2. Assume that the two numbers are of N-bit size, then this bitwise multiplication will yield N² bits which are known as Partial Products. This bitwise multiplication is carried out using AND Gate. The optimization of logic can be done while reducing the number of partial products or the number of stages in partial products. Adders are used to add these partial products to obtain result. If there are two bits then Half Adder is used for addition, while Full Adder is used to add three bits.

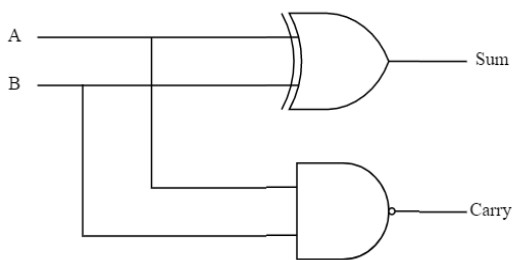
A. Half Adder

A Half Adder is a 1-bit adder i.e., used to perform addition of two 1-bit numbers. Half Adder takes two bits as input and generates sum and carry by adding the two bits. The sum and carry can be directly represented in the form of equations.

For any two input bits A and B,

$$\text{Sum} = A \text{ XOR } B$$

$$\text{Carry} = A \text{ AND } B$$



Half Adder

Fig-1: Block Diagram of Half Adder

Table-I: Truth Table of Half Adder

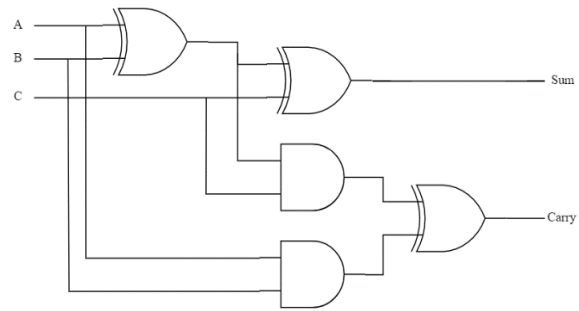
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

B. Full Adder

A Full Adder is also a 1-bit adder which is going to add three 1-bit numbers to generate sum and carry. The sum and carry are directly given by equations. This Full Adder can be realized using either two Half Adders or directly by solving the equations. After designing and simulating Full Adder in both ways, it is observed that design of Full Adder using two Half Adders is less efficient than the later method. For any three inputs A, B and C the output equations are:

$$\text{Sum} = A \text{ XOR } B \text{ XOR } C$$

$$\text{Carry} = (A \text{ AND } B) \text{ XOR } (C \text{ AND } (A \text{ XOR } B)).$$



Full Adder

Fig-2: Block Diagram of Full Adder

Table-II: Truth Table of Full Adder

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

C. Ripple Carry Adder

Half Adder and Full Adder are single bit adders. In order to add more than 1-bit numbers, special adder circuits are required. The point here is any adder will have Half Adders and Full Adders internally in its design. For example, a 4-bit Ripple Carry Adder requires four Full Adders to perform complete addition.

Let A = A₃A₂A₁A₀ and B = B₃B₂B₁B₀ are given as inputs for a Ripple Carry Adder. The first Full Adder takes A₀ and B₀ along with another input C_{in} which is a carry (Initially 0 will be given as C_{in}). The first Full Adder generates S₀ and C_{out} as outputs. The S₀ is directly given to output line and C_{out} is given as C_{in} for next Full Adder. This process continues with all four Full Adders. The final output of Four-bit Ripple Carry Adder will have S₃, S₂, S₁, S₀ along with C_{out} of final Full Adder. These four sum variables along with carry variable gives the result of addition of given inputs A and B.

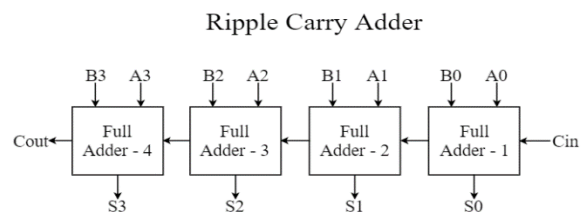


Fig-3: Block Diagram of Ripple Carry Adder

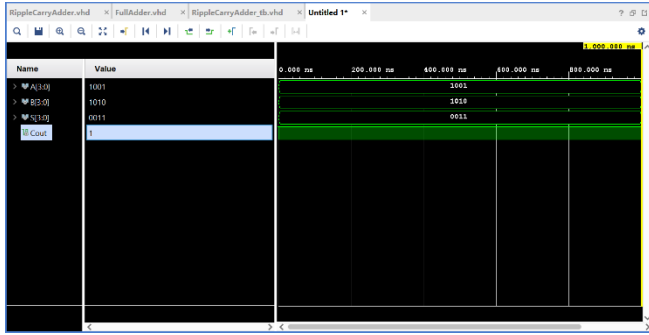


Fig-4: Simulated Output of Addition of two numbers using Ripple Carry Adder

D. Carry Select Adder

Carry Select Adder is one of the most efficient adder techniques. The basic problem in Ripple Carry Adder is there is a huge propagation delay as it must wait for the Carry every time. Carry is a 1-bit binary number. So, there are two possible options for carry bit i.e., either 1 or 0.

The complete working of Carry Select Adder lies in the above statement. For any two N-bit numbers, Carry Select Adder uses two N-bit Ripple Carry Adders along with N+1 2:1 Multiplexers. One of the two Ripple Carry Adders fed with '0' as Carry bit and other with '1' as Carry bit covering all two possibilities. Based on the original Carry bit, one of the two outputs got selected using 2:1 Multiplexer.

Carry Select Adder

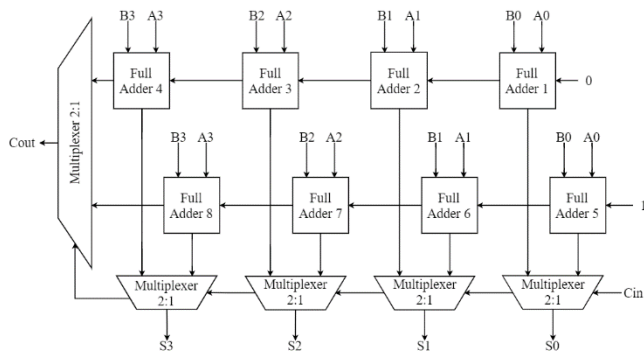


Fig-5: Block Diagram of Carry Select Adder

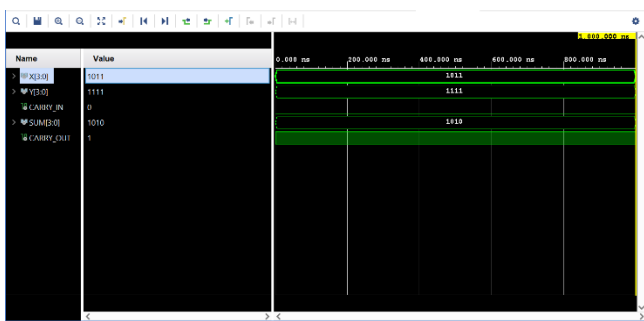


Fig-6: Simulated output of addition of two numbers using Carry Select Adder

IV. WALLACE TREE ALGORITHM

Out of all multiplication algorithms [4], Wallace Tree Algorithm is the efficient algorithm in terms of achieving low latency. It is a very simple algorithm developed by Chris Wallace in the year 1964. This algorithm mainly deals with the reduction of partial product layers. This Wallace Tree Algorithm [7] has three stages to yield final multiplication result. The Fig-7 shows the block diagram of Wallace Tree

Multiplier. Each block indicates a stage in the process of Wallace Tree Algorithm.

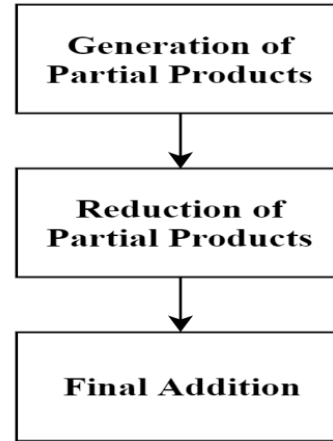


Fig-7: Block Diagram of Wallace Tree Multiplier

A. Stage-1: Generation of Partial Products

Assuming the multiplication between two 4-bit numbers $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$, the generation of partial products will be as follows:

	B3	B2	B1	B0	X	A3	A2	A1	A0
						A0B3	A0B2	A0B1	A0B0
						A1B3	A1B2	A1B1	A1B0
						A2B3	A2B2	A2B1	A2B0
						A3B3	A3B2	A3B1	A3B0

These partial products can be generated using AND gate.

B. Stage-2: Reduction of Layers of Partial Products

Wallace Tree Algorithm is popular because of its technique in reducing partial products. This algorithm follows reduction of three layers into two layers. This can be done using a ladder of Half Adders and Full Adders. If two bits are to be added then Half Adder is used, similarly Full Adder for three bits. Both Half Adder and Full Adder gives Sum and Carry as outputs. The sum and carry values are stored in signals where they can be used in solving the next layer. These signals will be given numbering alternatively based on their usage in the code while reducing the number of layers. The main objective here is to reduce N layers into two layers so that they can be added easily. The reduction will be as follows:

Table-III: Reduction table of 4-bit Multiplication

				A0B3	A0B2	A0B1	A0B0
				A1B3	A1B2	A1B1	A1B0
				A2B3	A2B2	A2B1	A2B0
				A3B3	A3B2	A3B1	A3B0
	A2B3	S3	S2	S1	S0	A0B0	
	C3	C2	C1	C0			
	A3B3	A3B2	A3B1	A3B0			
	C7	S7	S6	S5	S4	S0	A0B0
	A3B3	C6	C5	C4			

In the above reduction, if there are two terms, they will be given as input to Half Adder. If there are three terms, they will be given to Full Adder. The obtained sum and carry are used in next layer. For any N-bit multiplication there will be N-1 reduction stages before final addition. After the complete reduction there will be only two layers which contribute to the final product. The two layers are given as input to N-bit adders for final addition.

C. Stage-3: Final Addition of Partial Products.

The most crucial stage of any multiplier design is Final Addition Step. There are many adder circuits which can add two numbers of N-bits each. This paper already discussed two adder circuits i.e., Ripple Carry Adder and Carry Select Adder. The last layer in the Wallace reduction tree is given to N-bit adder circuits to obtain final output. Therefore, it is very much necessary to use fast adder circuit to yield faster output. This paper used both adders for multiplier design and compared the characteristics of multiplier with both adders. The comparison was made in terms of latency, number of slices used etc.

V. RESULTS AND ANALYSIS

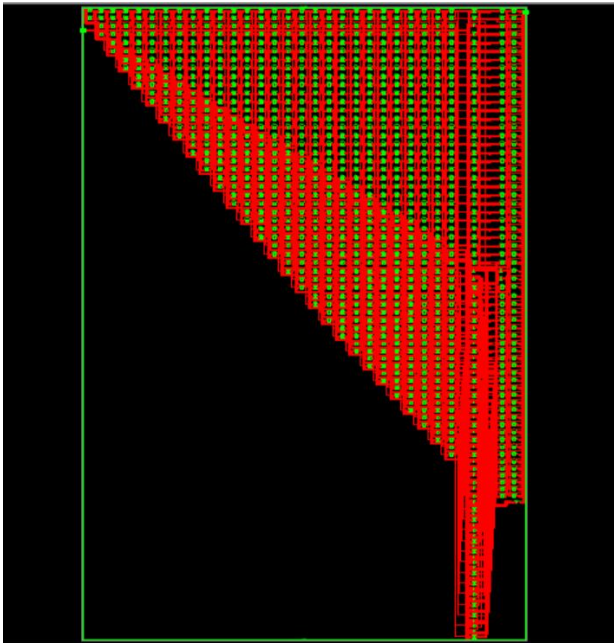


Fig-8: RTL Schematic of Wallace Tree Multiplier (RCA)

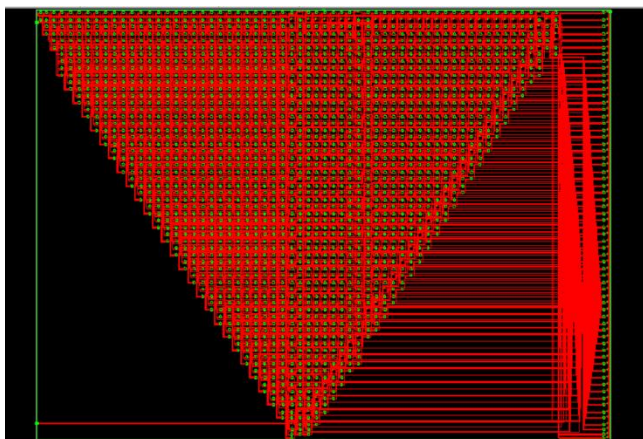


Fig-9: Technology View of Wallace Tree Multiplier (RCA)

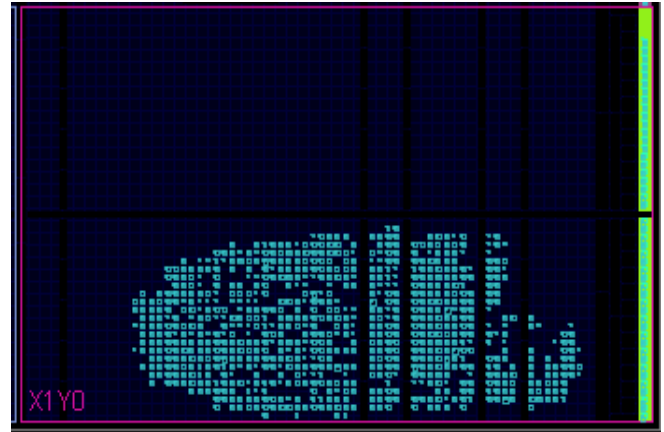


Fig-10: FPGA View of Wallace Tree Multiplier (RCA)



Fig-11: RTL Schematic of Wallace Tree Multiplier (CSA)

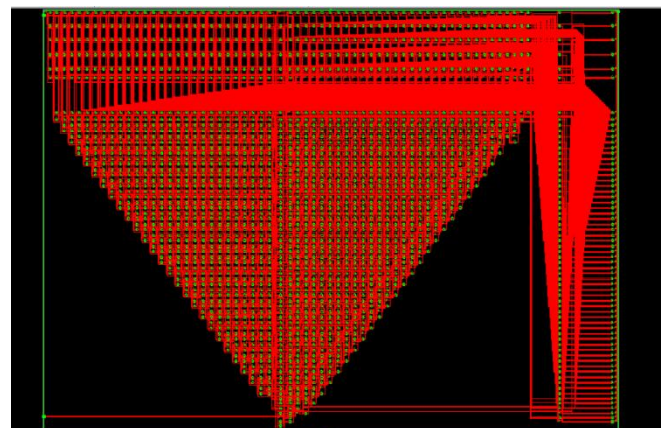


Fig-12: Technology View of Wallace Tree Multiplier (CSA)

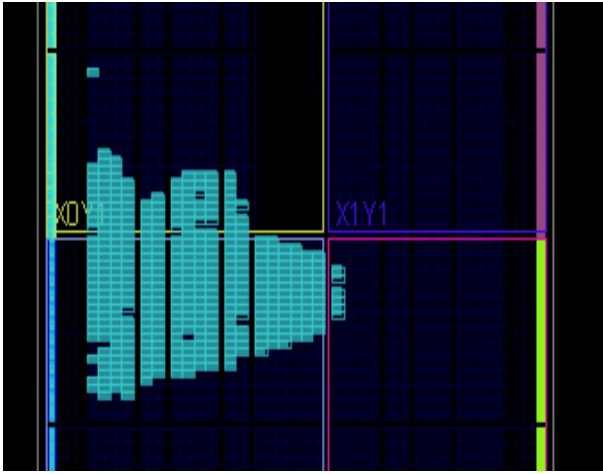


Fig-13: FPGA View of Wallace Tree Multiplier (CSA)

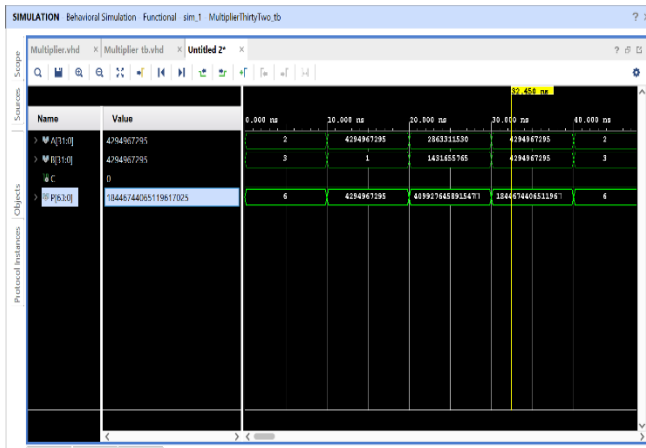


Fig-14: Simulated output of 32-bit multiplication using Wallace Tree Multiplication Algorithm

The two designs discussed in the paper are very much similar as the only change is in the adder circuit used. The final adder is used for one-time N-bit addition. So, the power consumption in both the designs is same. The power consumption obtained according to the simulated results from Xilinx ISE is 4.1 milli watts.

As Carry Select Adder is fastest adder compared to Ripple Carry Adder, a considerable amount of delay can be observed in the case of Ripple Carry Adder Design. Always there will be a trade-off between Latency and Memory. Low latency can be achieved when more memory is allocated for the device to process.

Table-IV: Latency and Memory Comparison of Wallace Tree Multiplier with RCA and CSA

Adder Circuit Used	Latency in Nano Seconds	Memory in Kilo Bytes
Ripple Carry Adder	117.884	4618924
Carry Select Adder	95.984	4628140

The number of Slices, LUT and Io used, ports below system noise margin are given by the below table:

Table-V: Device Utilization and Noise Margin Report

Criterion for Comparison	RCA	CSA
Number of Slices Used	1191	1222

Number of Input LUTs	2071	2134
Number of IOs	128	129
Number of Bonded IOBs	128	129
Ports below SSN	64 out of 64	64 out of 64

VI. CONCLUSION

The primary goal of this paper is to design an efficient implementation of Wallace Tree Multiplier. This task was achieved successfully for a 32-bit binary multiplication using an architectural change made at the stage of final addition i.e., change of adder circuit used. The results and comparison between designs was also presented in this paper. The aim of reducing latency has been achieved by using a Carry Select Adder for the 32-bit final addition. A comparison was made in terms of speed, memory, device utilization between the multiplier design with Carry Select Adder and Ripple Carry Adder.

FUTURE SCOPE

This paper has proposed an efficient way of designing a multiplier. [8] If an efficient adder circuit which has compatibility with this multiplier is designed, then this can be extended further by making an efficient design of MAC unit. Also, ALU can also be designed efficiently.

The outcome in which these efficient ALU and MAC units can be used is CNN. In CNN each layer will have these MAC and ALU. If these blocks are implemented with utmost efficiency, then eventually the high-end applications will be implemented with more efficiency.

One way of designing an efficient multiplier i.e., change in the adder used has been discussed in this paper. Another way of improving efficiency is changing the way of reduction of partial products. There are many ways such as combining Wallace Tree Algorithm with booth encoding [3][5] to reduce number of layers of partial products.

This paper implemented the design of a 32-bit unsigned multiplier architecture which can be extended further to a 64-bit architecture or a signed multiplication architecture or a floating-point multiplication architecture.

REFERENCES

1. VLSI Digital Signal Processing Systems by Keshab K. Parthi.
2. A Review of Different Type of Multipliers and Multiplier-Accumulator Unit by Soniya and Suresh Kumar.
3. A High-speed Unsigned 32-bit Multiplier Based on Booth-encoder and Wallace-tree Modifications by Xuan-Vy Luu, Trong-Thuc Hoang, and Trong-Tu Bui.
4. Analysis of Different Multiplication Algorithms and FPGA Implementation by K. Harika, B.V. Swetha, B.Renuka, D. Lakshman Rao and S. Sridhar.
5. High performance pipelined signed 64x64-bit multiplier using radix-32 modified Booth algorithm and Wallace structure by Manish Bansal, Sangeeta Nakhate and Ajay Somkuwar.
6. Implementation and Analysis of a different 32-bit multipliers on aspects of Power, Speed and Area by Siva.S.Sinthara, Afreen Begum, B. Amala, A. Vimala and V. Vishya Aparna.
7. A Proposed Wallace Tree Multiplier Using Full Adder and Half Adder by Swathi A.C, Yuvaraj T, Praveen J and Raghavendra Rao A.
8. High Performance Wallace Tree Multiplier Using Improved Adder by Meenali Janveja and Vandana Niranjana.

AUTHORS PROFILE



Sunkana Rama Krishna received the M.Tech degree from Jawaharlal Nehru Technological University Kakinada, B.Tech from BPUT, Rourkel India. He also Member in MIE, MISTE. Currently he is a Associate Professor of ECE in Lendi institute of engineering and Technology, vizianagaram, India. He has published over 20 papers in International and National Journal/Conferences. His research interest includes

Vlsi, Embedded, communication, 5G Wireless communications.



Arun Kishore Voleti is in his final semester in his Bachelor of Technology degree in the department of Electronics and Communication Engineering in the Lendi Institute of Engineering and Technology, Vizianagaram, Andhra Pradesh. He, in his final semester worked in a project in the stream of VLSI design. He has attended many national conferences in the fields of VLSI design, Signal and Image Processing, Advanced Computer Architecture and

5G Communications. He was an IEEE member in the year 2019-2020. His areas of interest include VLSI Design, Signal and Image Processing, Low Power VLSI Design, IoT and AI.



Jagadeesh Pasumarthi is currently pursuing his Bachelor of Technology degree from Lendi Institute of Engineering and Technology, Vizianagaram, Andhra Pradesh in the stream of Electronics and Communication Engineering. He is in his final semester. He worked in a project in the stream of VLSI design. His areas of interest are VLSI Design, Embedded systems and AI. He has participated in many workshops and attended conferences in the

fields of communications and IoT.



Deepika Yelusuri is currently pursuing his Bachelor of Technology degree from Lendi Institute of Engineering and Technology, Vizianagaram, Andhra Pradesh in the stream of Electronics and Communication Engineering. Her areas of interest are VLSI & IOT. She had done mini projects like Line Follower Robot in IOT domain. She also attended many workshops.



Rachakonda Sai Kiran is currently pursuing his Bachelor of Technology degree from Lendi Institute of Engineering and Technology, Vizianagaram, Andhra Pradesh in the stream of Electronics and Communication Engineering. His areas of interest are VLSI, AI and Virtual Reality.