EFFICIENT AND COMPATIBLE VLSI ARCHITECTURE OF PARALLEL MULTIPLIER AND ACCUMULATOR

PratibhadeviTapashetti
Research Scholar
Department of ECE
NIMS University Rajasthan
Email:pratibhat702@gmail.com
Dr. Praveenkumar
Professor, NIMS University Rajasthan

Abstract

In the developing demand of todays life there is a requirements for high speed operations of digital signal processing and other similar applications. This need high speed low power, less delay and compact accurate circuits. Multiplication and addition are the major required operations for digital signal processing. This need is fulfilled by using new architectures for modified booth algorithm ,carry save adder and Wallace tree. Modified booth algorithm increases the speed by reducing the number of partial products and carry save adder performs the sum of partial products which further reduces the delay and power. New architectures are designed, simulated and synthesized using XILINX ISE.Delay and power analysis are shown in the result.

Key words_CSA carry save adder,CLACarry Look ahead Adder,MAC Multiplier and accumulator.

INTRODUCTION

The demand for high speed and efficient processing has been mounting as a result of growing computer and signal processing applications.

The core of every processing system is its data path. Available statistics [3] has given clear indications that more than 70% of the instructions usually perform arithmetical and logical operations mainly consist of addition and multiplication in the data path of RISC and CISC machines. Multiplication based computation, which involve operations like Multiply and Accumulate and inner product most intensive arithmetic functions, currently implemented in many signal processing applications such as convolution, fast Fourier transform, filtering and in microprocessors in its arithmetic and logic unit. Since multiplication dominates the execution time of most signal/ instruction processing algorithms, so there is a need of speed efficient multiplier. Also, low power consumption and area efficiency are among the most important criteria for the fabrication of any processing and high performance systems. However, area and speed are usually contradictory, so that improving speed results mostly in larger areas also.

In our study, we propose the best solution to this problem by introducing a new efficient VLSI architecture of parallel Multiplier-and Accumulator (MAC) using hybrid approach for high-speed arithmetic operations. Since the accumulator that has the largest delay in MAC, the Carry Save Adder (CSA) and compressor techniques are used as one of the processing element to improve the overall performance.

As per the previous work, Booth algorithm and CLA’s are used in MAC operation for getting the efficient output results through pipelinescheme. To filter the problems theproposed methodprefers modified Booth algorithm and CSA.

This study presents an efficient implementation of high speed multiplier. The radix 2 and radix 4 modified Booth multiplier does the operations using fewer adders and only few iterative steps.

However, the fact remains that the area and speed are two conflicting performance constraints. Hence, innovating increased speed always results in larger area. In this, we arrive at a better trade-off between the two, by realizing a marginally increased speed performance

The proposed MAC will show the better properties to the existing standard design in many ways and performance twice as much as the previous research in the similar clock frequency.

The power reduction techniques adopted in this work, and we expect that the proposed MAC can be adapted to various fields requiring high performance such as the signal processing areas.

RELATED WORK

MAC showed theproperties to the design in different ways andperformanceimprovement twice as that of the previous work with same clock frequency [7].

A new MAC architecture to execute the multiplication accumulation operation, for digital signal processing and multimedia information processing efficiently was proposed[8]. By removing the independent accumulation process has the huge delay and merging it to the compression process of the partial products, the overall MAC performance has been improved almost twice as that of the previous work.

A new MAC architecture to execute the multiplication - accumulation operation, which is the key operation, for digital signal processing and multimedia information processing efficiently, was proposed. By removing the independent accumulation process which has the highest delay and merging it to the compression process of the partial products[9].

An efficient implementation of high speed multiplier using the shift and add modified Booth algorithm [10] is presented. The adder used is look ahead carry adder. The compression tree along with the carry look ahead adder has reduced the hardware overhead and power consumption.

A new multiplier Accumulator architecture based on high accuracy modified Booth algorithm [11]. In this paper, a new MAC architecture is developed for high speed performance. Multipliers with high speed are essential for digital applications. A new architecture of MAC was proposed for high speed arithmetic in which multiplication and accumulation togetherness improves the performance [12].

In this paper [13], a new architecture of MAC to increase the speed of arithmetic was proposed. Combining multiplication with accumulation with carry save adder CSA,increases the performance. Ithas CSA tree with 1's complementradix-2 basedmodified Booth's algorithm which has the modified array for extending the sign.

IMPORTANCE OF THE PROPOSED WORK

By showing the performance improvement of the Proposed MAC more than twice than the earlier research it can have the importance as below.

- It is seen that this MAC may beused in various fields requiring high speed and performance like the signal processing areas.
- MAC unit designed can be used in efficient filter realization for high speed DSP applications.
- Efficient MAC can have extension to design floating point arithmetic, signed arithmetic and Decimal arithmetic.
- Multimedia processing is the one of the major area requires high speed multiplier.
- Further the designed MAC can be used in many areas like numerical coprocessors, calculators filtering, modulation and demodulation etc.

In this work an architecture is proposed for efficient multiplication with Booths radix-4 algorithm, Modified Booths multiplier with bit pair recoding , 16 bit Wallace tree multiplier to improvethe speed.

In the proposed architecture, partial product generation and reduction is accomplished by the use of booth algorithm, 3:2, and 4:2, 5:2 compressor structures.

PROPOSED IMPLEMENTATION

The proposed MAC is analyzed and implemented. Then it is compared with some previous results. First the percentage of used hardware in implementing the hardware is analyzed theoretically and experimentally then the delay of the hardware is analyzed. Then the pipeline stage is defined and performance is analyzed based on the pipelining scheme. Implementation result from each section will be compared with standard design [4] and Equibaly's design [3], each of which has the most representative parallel MBA architecture.

Overview of MAC:

In this MAC operation a multiplier can be divided into 3 parts. The first is radix – 8 booth encoding in which the partial product is generated from the multiplicand (x) the multiplier (y).The second is adder array or partial product compression. The last is the final addition in which final multiplication result is produced by adding the sum & the carry.

General hardware architecture of this MAC is below in fig. -1

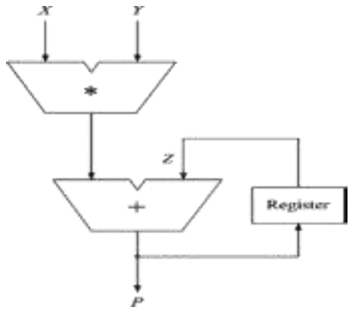

Fig.1General Hardware architecture of MAC

General hardware architecture of the MAC executes the multiplication operation by multiplying the input multiplicand X and multiplier Y. This is added to the previous multiplication result Z as the accumulation step if accumulation is needed.

The N-bit 2's complementbinary number can be expressed as

$$X = -2^{N-1}x_{N-1} + \sum_{i=0}^{N-2} x_i 2^i, \qquad x_i \in 0,1.$$

……..(1)

If (1) is expressed in base-4 type redundant sign digit form in order to apply the radix-8 Booth's algorithm.

$$X = \sum_{i=0}^{N/2-1} d_i 4_i$$

……..(2)

Where

$$d_i = -2x_{2i+1} + x_{2i} + x_{2i-1}.$$

……… (3)

If (2) is used, multiplication can be expressed as

$$X \times Y = \sum_{i=0}^{N/2-1} d_i 2^{2i} Y.$$

……… (4)

The multiplication - accumulation result are then

$$P = X \times Y + Z = \sum_{i=0}^{N/2-1} d_i 2^i Y + \sum_{j=0}^{2N-1} z_i 2^i.$$

……. (5)

The MAC architecture implemented by (5) is called the standard design [4].

BOOTH ALGORITHM

Booth Algorithm

Booth multiplication algorithm was named after the inventor Andrew Donald Booth. It is defined as an algorithm of multiplying binary numbers in two's complement notation. It is a method to multiply binary numbers where multiplication is carried out by repeated addition operations. Further this booth algorithm for multiplication is modified to reduce the number of partial products through bit pair recoding and known as modified booth algorithm.

Modified Booth Algorithm

Booth multiplication algorithm consists of three major steps that includes generation of partial product called as recoding, reducing the partial product in two rows, and addition that gives final product.

In modified Booth algorithm it includes three steps, firstly the bit pair recoding ,secondly reducing number of partial products and then finally these partial products are areadded.

Fast Multiplication

To speed p the multiplication process there are two techniques

- The first technique calculates the maximum number of summands that should be added is n/2 for n-bit operands.
- The second method reduces the time taken to add these summands.

Bit Pair Recoding Of Multipliers

These is a technique where maximum number of summands are reduced to half and is derived from Booths algorithms which is known as bit pair recoding.

Bit pair recoding with an example is shown below.

Multiplier:
signextention< - 1 1 1 0 1 0 0 -> implied zero
Booth recoding: 0 0 -1 +1 -1 0
Bit pair recoding :0 -1 -2

Table 1.multiplicand selection decisions.

y_{2i+1}	y_{2i}	y_{2i-1}	Generated partial products
0	0	0	$0 \times X$
0	0	1	$1 \times X$
0	1	0	$1 \times X$
0	1	1	$2 \times X$
1	0	0	$(-2) \times X$
1	0	1	$(-1) \times X$
1	1	0	$(-1) \times X$
1	1	1	$0 \times X$

An example of multiplication:

$$\begin{array}{rll}
0\;0\;1\;1\;1\;0 & (21)_{10} & \text{Multiplicand} \\
0\;1\;0\;1\;0\;1 & (14)_{10} & \text{Multiplier} \\
\times\;\underbrace{+1\;-1}_{+1}\;\underbrace{+1\;-1}_{+1}\;\underbrace{+1\;-1}_{+1} & \text{Booth recoded multiplier} & \\
 & \text{Bit pair recoded multiplier} & \\
\hline
0\;0\;0\;0\;0\;0\;0\;0\;1\;1\;1\;0 & (14 \times 1)_{10} & \\
0\;0\;0\;0\;0\;0\;1\;1\;1\;0\;0\;0 & (14 \times 4)_{10} & \\
0\;0\;0\;0\;1\;1\;1\;0\;0\;0\;0\;0 & (14 \times 16)_{10} & \\
\hline
0\;0\;0\;1\;0\;0\;1\;0\;0\;1\;1\;0 & (294)_{10} & \text{Product}
\end{array}$$

In the above example using Booths algorithm with bit pair recoding the number of partial products are reduced from six to three.

IMPLEMENTATION

A multiplier design comprises of three major steps. The first is generation of partial products using Booth encoding from the multiplicand X and the multiplier Y. The second isadder array or partial product compression which adds partial products and generates sum and carry. Finally thel addition is done in which the multiplication result isproduced by adding the sum and the carry.

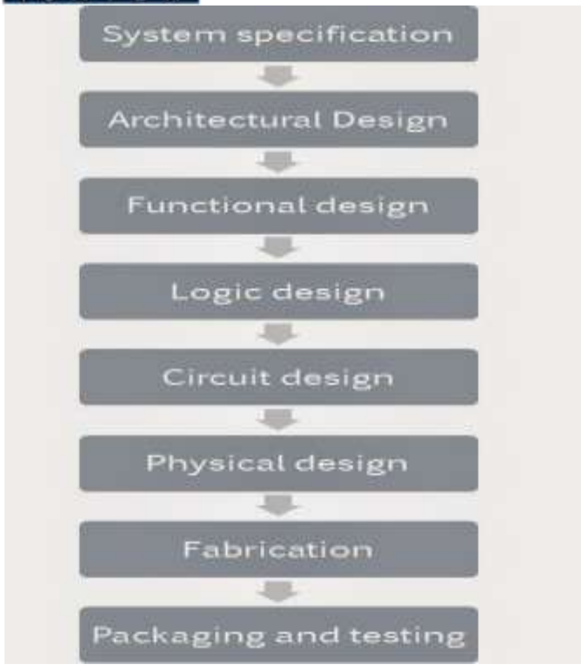

Fig.2.Flow diagram of VLSI design

Today, VLSI design flow is a very solid and mature process. The overall VLSI design flow and the various steps within the VLSI design flow have proven to be both practical and robust in multi-millions VLSI designs until now.

Each and every step of the VLSI design flow has a dedicated EDA tool that covers all the aspects related to the specific task perfectly. And most importantly, all the EDA tools can import and export the different file types to help making a flexible VLSI design flow that uses multiple tools from different vendors.

VLSI design flow is not exactly a push button process. To succeed in the VLSI design flow process, one must have: a robust and silicon-proven flow, a good understanding of the chip specifications and constraints, and an absolute domination over the required EDA tools (and their reports!).

This article covers the VLSI design flow in very high level. We will provide a more detailed articles in the future explaining more about the activities within each phase. Let's start with the first step.

VLSI System Design

Assuming your VLSI specifications are completed and approved by the different parties, it's time to start thinking about the architectural design. In VLSI system design phase, the entire chip functionality is broken down to small pieces with clear understanding about the block implementation. For example: for an encryption block, do you use a CPU or a state machine. Some other large blocks need to be divided into subsystems and the relationship between the various blocks has to be defined. In this phase the working environment is documentation.

Register Transfer Level (RTL)

For digital VLSIs or for digital blocks within a mixed-signal chip, this phase is bVLSIally the detailed logic implementation of the entire VLSI. This is where the detailed system specifications is converted into VHDL or Verilog language. In addition to the digital implementation, a functional verification is performed to ensure the RTL design is done according to the specifications.

Synthesis

In this phase the hardware description (RTL) is converted to a gate level netlist. This process is performed by a synthesis tool that takes a standard cell library, constraints and the RTL code and produces an gate-level netlist.

Synthesis tools are running different implementations to provide best gate level netlist that meets the constraints. It takes into account power, speed, size and therefore the results can vary much from each other. To verify whether the synthesis tool has correctly generated the gate-level netlist a verification should be done.

Layout

In this stage, the gate level netlist is converted to a complete physical geometric representation. The first step is floorplanning which is a process of placing the various blocks and the I/O pads across the chip area based on the design constraints. Then placement of physical elements within each block and integration of analog blocks or external IP cores is performed. When all the elements are placed, a global and detailed routing is running to connect all the elements together.

Also after this phase a complete simulation is required to ensure the layout phase is properly done.

The file produced at the output of the layout is the GDSII (GDS2) file which is the file used by the foundry to fabricate the silicon. The layout should be done according the silicon foundry design rules.

RESULTS

The simulation and synthesis results are shown below with timing diagrams.

Table 2 Synthesis results for Wallace tree

HDL Synthesis Report
Synthesis Results for the Wallace Tree Multiplier

Logic Utilization	**Used**	**Available**
Slice LUTs	359	63400
LUT FF pairs	0	359
Bonded IOBs	210	320

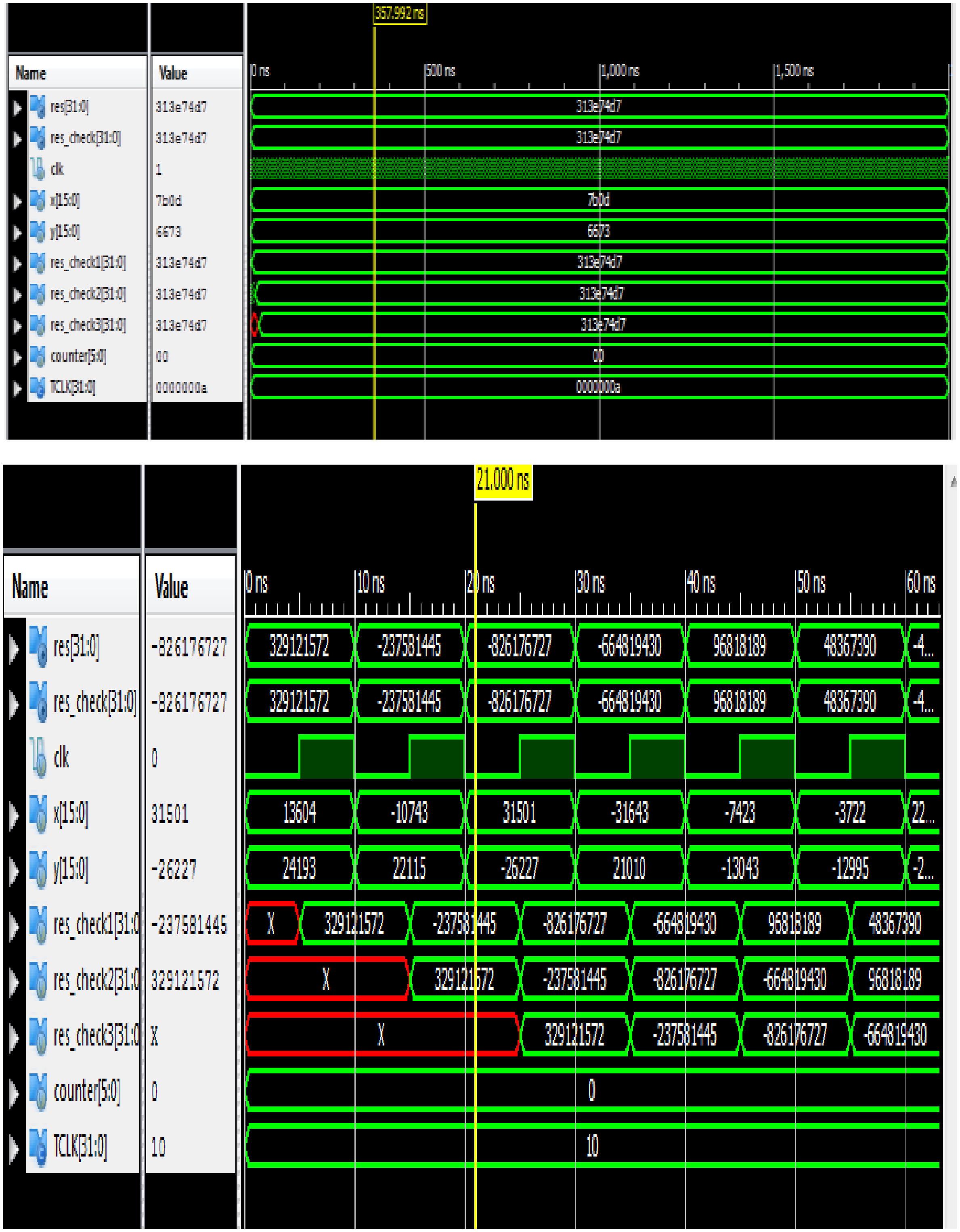
357.992 ns
Name
Value
0 ns
500 ns
1,000 ns
1,500 ns
res[31:0] 313e74d7
res_check[31:0] 313e74d7
clk 1
x[15:0] 7b0d
y[15:0] 6673
res_check1[31:0] 313e74d7
res_check2[31:0] 313e74d7
res_check3[31:0] 313e74d7
counter[5:0] 00
TCLK[31:0] 0000000a
21.000 ns
Name
Value
0 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
res[31:0] -826176727 329121572 -237581445 -826176727 -664819430 96818189 48367390 -4...
res_check[31:0] -826176727 329121572 -237581445 -826176727 -664819430 96818189 48367390 -4...
clk 0
x[15:0] 31501 13604 -10743 31501 -31643 -7423 -3722 22...
y[15:0] -26227 24193 22115 -26227 21010 -13043 -12995 -2...
res_check1[31:0] -237581445 X 329121572 -237581445 -826176727 -664819430 96818189 48367390
res_check2[31:0] 329121572 X 329121572 -237581445 -826176727 -664819430 96818189
res_check3[31:0] X X 329121572 -237581445 -826176727 -664819430
counter[5:0] 0 0
TCLK[31:0] 10 10

Fig. 3.Simulation waveforms

Table 3 Synthesis results for Carry save adder

Below given table 4 presents the cumulative results for the overall proposed architecture

Synthesis Results for Carry save adder

Logic Utilization	**Used**	**Available**
Slice LUTs	58	63400
LUT FF pairs	0	359
Bonded IOBs	98	320
Delay	8.795 ns	

Table 4 Overall architecture synthesis result

Logic Utilization	**Used**	**Available**
4 input Slice LUTs	878	1536
Occupied Slices	473	768
Bonded IOBs	64	320
Delay	12.728ns	

Table 5. Power Results generated by Xilinx ISE

Power Supply Currents			
Supply Source	Supply voltage	Total Current(mA)	Quiescent Current (mA)
Vccint	1	16.58	16.58
Vccaux	1.8	13.14	13.14
Vcc018	1.8	1.00	1.00
Vccbram	1	0.35	0.35

The simulation output is -826176727 with th e inputs X= -31501 and Y= -26227 which is shown in fig 3

CONCLUSION

In this paper, we propose Aefficient and compatible VLSI architecture of multiplier and accumulator which is a most needed for digital signal processing and multimedia communication applications . The proposed models consists of modified booth algorithm, carry save adder and Wallace tree. The importance of the existing design shows the reduction of the partial products by using the modified booth algorithm for parallel processing. The proposed design shows the improved results in terms of power consumption, area and delay which are the parameters to be bend for any circuit.

References

[1] S. Waser and M.J. Flymn, "*Introduction to arithmetic for Digital system designers*," New York: Holt, Rinechart and Winston, 1982.

[2] J J F Cavanagh, *Digital computer arithmetic*. New York: McGrawHill 1984.

[3] F. Elguibaly, "A fast parallel multiplier–accumulator using the modified Booth algorithm," *IEEE Trans. Circuits Syst.*, vol. 27, no. 9, pp. 902–908, Sep. 2000.

[4] A.R. Omondi, *Computer arithmetic systems*, Eagle wood cliffs. NJ: Printice Hall. 1994.

[5] J.FadaviArdekani, " Booth encoded multiplier generator using optimize Wallace tree," *IEEE Transactions on VLSI Systems*, vol.1, pp.120-125, 1993.

[6] A. Fayed and M. Bayoumi, "A merged multiplier-accumulator for high speed signal processing applications," *Proc. ICASSP*, vol. 3, pp. 3212–3215, 2002.

[7] P. Zicari, S. Perri, P. Corsonello, and G. Cocorullo, "An optimized adder accumulator for high speed MACs," *Proc. ASICON 2005*, vol. 2, pp. 757–760, 2005.

[8] Young Ho seo, Dong wookkim, "A new VLSI Architecture of parallel multiplier Accumulator based on Radix 2 Modified Booth algorithm," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 18, no.2, pp. 201-208 , Feb.2010.

[9] M.V. Sathish, M. Sailaja, "Vlsi Architecture Of Parallel Multiplier– Accumulator Based On Radix-2 Modified Booth Algorithm," *International Journal of Electrical and Electronics Engineering (IJEEE)*, vol.1, Issue-1, 2011

[10] P. SasiBala, S. Raghuvendra, "A new VLSI architecture of parallel multiplier – Accumulator based on Radix – 2 modified Booth algorithm," *International Journal of Instrumentation, control and Automation*, vol.1, no.2, pp.91-97, 2011.

[11] SankeyGoel, R.K. Sharma, "Parallel MAC based on Radix 4 & Radix 8 Booth encodings," *International Journals of Engineering & Technology*, vol.3, no.8, pp.6692-6697, 2011.

[12] K. HimaBindu, K. BalaSouri, K. V. RamanaRao, " MAC architecture Accumulator based on Booth encoding parallel multipliers,"*International journals of soft computing and Engineering*,vol.1, no.5, pp. 385–388, Nov.2011.

[13] Ren-Der Chen; Yu-Cheng Chou; Wan-Chen Liu, "Comparative design of floating-point arithmetic units using the Balsa synthesis system," Integrated Circuits (ISIC), 2011 13th International Symposium on , vol., no., pp.172,175, 12-14 Dec. 2011

[14] RashmiRanjan et al., "A New Vlsi Architecture Of Parallel Multiplier Based On Radix-8 Modified Booth Algorithm Using Vhdl," *International Journal of Computer Science & Engineering Technology (IJCSET)*, Vol. 3 No. 4 April,2012

[15] AdalankiPurna Ramesh, Dr. A. V. N. Tilak, Dr. A. M. Prasad, "Efficient implementation of 16 bit multiplier, Accumulator using Radix – 2 modified Booth algorithm and SPST adder using Verilog," *International journal of VLSI design and communication systems*, vol. 3, no.3, pp.107-118, Jun. 2012.

[16] S. Jagadish, S. Venkatachary, "Design of parallel multiplier Accumulator based on Radix-8 modified

Booth algorithm with SPST," *International journal of Engineering Research and application*, vol.2, no.5, pp.425-435,Sept.Oct. 2012.

[17] B. Jyothirmai, M. Premalatha, "Design and implementation of parallel MAC by Booth algorithm," *International Journals of scientific Research,*vol.1,no.6, pp.78-79,Nov.2012.

[18] Harilal, Durga Prasad, "High speed arithmetic Architecture of parallel multiplier Accumulator based on Radix-2 modified Booth algorithm," *International journals of computational Engineering research*, vol.2, no.8, pp.28-38, Dec.2012.

[19] Naveen Kumar, Manu Bansal, AmandeepKaur, "Speed Power and Area Efficient VLSIArchitectures of Multiplier and Accumulator," *International Journal of Scientific & Engineering Research,*vol.4, Issue 1, Jan-2013

[20] J. Y. YahwanthBabu, P. Dinesh Kumar, "Anew multiplier Accumulator architecture based on high accuracy modified Booth algorithm," *International journal of advanced research in computer Engineering & Technology*, vol.2,no.3,pp.1036-1040,Mar 2013.

[21] Suriya, T.S.U.; Rani, A.A., "Low power analysisof MAC using modified booth algorithm," Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on , vol., no., pp.1,5, 4-6 July 2013

[22] Sathya, A.; Fathimabee, S.; Divya, S., "Parallel multiplier-accumulator based on radix-2 modified Booth algorithm by using a VLSI architecture," Electronics and Communication Systems (ICECS), 2014 International Conference on , vol., no., pp.1,5, 13-14 Feb. 2014

[23] Pratibhadevi Tapashetti, Dr.Rajkumar B Kulkarni, "Implementation of High Speed MAC VLSI Architectures, Based on High Radix Modified Booth Algorithm", International conference on advances in electronics ,computer and mathematical sciences(ICIREMPS-2016) Sagar Institute of Research and Technology(SIRT) Bhopal,MP,India,26 to 28 FEB 2016

[24] Pratibhadevi Tapashetti, Dr.Rajkumar B Kulkarni, "Efficient And Compatible VLSI Architecture Of Parallel Mac Based On High Radix Modified Booth Algorithm", 3rd International Conference onIET-IEEE-Electrical, Electronics, Engineering Trends, Communication, Optimization and Sciences (EEECOS)-2016, SASI Institute of Technology and Engineering ,Tadepalligudem, Andrapradesh,India, 2016/5.

[25] Pratibhadevi Tapashetti, Dr. Praveen Kumar, " Compatible Architecture of MAC, Based on Modified Booth Algorithm", International Journal of Emerging Technology and Advanced Engineering(IJETAE),ISSN-2250-2459, Volume 7, Issue 10, October 2017(UGC approved serial no:4677