International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 5, May 2020||

Design and Implementation of Multiplier using Different Techniques

Bharath Kumar L [1], Bharat Hegde [2], Arvind L [3], Manjula B B [4]

U.G. Student, Department of Electronics and Communication Engineering, East West Institute of Technology, Bangalore, India [1,2,3]

Associate Professor, Department of Electronics and Communication Engineering, East West Institute of Technology, Bangalore, India[4]

ABSTRACT: Multiplier is an important building block in the design of many digital circuits, since, multiplication is extensively used in systems such as calculators, microprocessor, digital filters,Digital signal processing and in many communication applications.With the help of VLSI domain, cost effective, speed efficient and less power consuming multiplier can be designed.There are various technical methods for constructing a multiplication system such as, Wallace Tree Multiplier, Booth Encoded Multiplier, Vedic Multiplier, Dadda Multiplier etc. This paper reveals performance analysis and comparison of famous multipliers. The proposed designs are implemented using Verilog code and simulated using ModelSim. The Design efficiency deciding parameters are calculated using Xilinx. It is found that Modified Booth Multiplier is best for implementation with more benefits.

KEYWORDS*:* Radix-4 Booth Encoded Multiplier, Wallace Tree Multiplier, UT Multiplier Partial Product reduction, lesser time delay.

I. INTRODUCTION

Multiplication is an important operation that is extensively used in many applicative systems such as Discrete Fourier Transform, Correlation, Convolution, Range measurement etc. Multiplication is a key factor in many fields. There are plenty of system outputs, whose efficiency depends on speed of multiplier, since, multiplier is the more time consuming unit . As the technology is advancing, the demand for high speed and efficient multiplier is been increasing. However, the term multiplication sounds easy but complexity increases with increase in number of digits since, each and every bit of multiplicand should be multiplied with the digits of the other term. Therefore it is a time consuming process. There are plenty of methods for designing multiplication system but the system such as Dadda Multiplier takes more time since, their operation is based upon simple add and shift operation. To minimize computational delay, main technique is to reduce the number of partial products. Here is the design and performance analysis of Wallace Tree Multiplier, Modified Booth Encoding Multiplier using Radix-4 method, Vedic Multiplier using Urdva Triyak Sutra technique.

II. METHODOLOGY

A. WALLACE TREE MULTIPLIER

Several popular and well-known schemes, with the objective of improving the speed of the parallel multiplier, have been developed in past. Wallace introduced a very important iterative realization of parallel multiplier . This advantage becomes more pronounced for multipliers of bigger than 16 bits. In this section we discuss about the design and development of a 32-bit Wallace tree multiplier. In Wallace tree architecture, all the bits of all of the partialproducts in each column are added together by a set of counters in parallel without propagating any carries. The block diagram of 32-bit Wallace Tree Multiplier is shown in Fig 1.

AfastprocessformultiplicationoftwonumberswasdevelopedbyWallace.Usingthismethod,a three step process is used to multiply two numbers; the bit products are formed, the bit product matrix is reduced to a two row matrix where sum of the row equals the sum of bit products, and the two resulting rows are summed with a fast adder to produce a final product.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 5, May 2020||

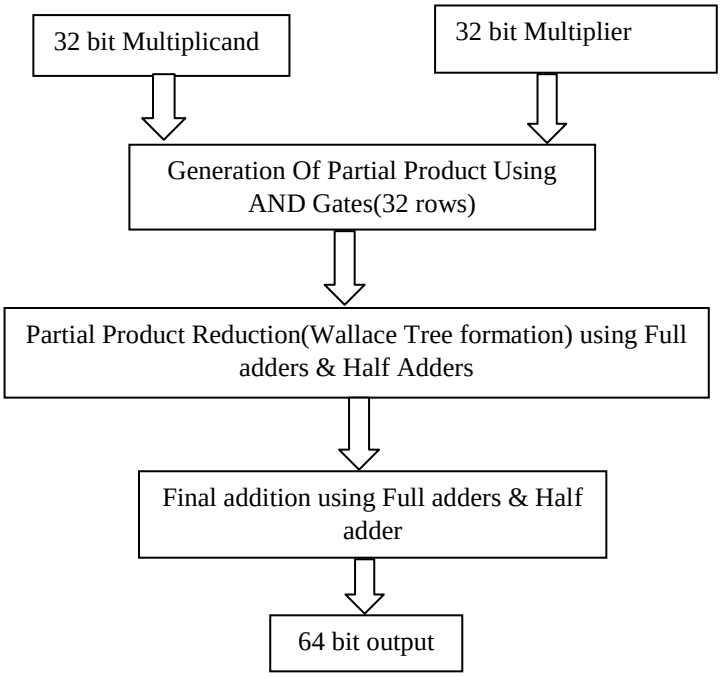


Fig 1. Block Diagram of 32-bit Wallace Tree Multiplier

B. URDHVA-TIRYAKBHYAM SUTRA

The Urdhva Tiryakbhyam (UT) sutras are very efficient sutras among 16 sutra's of Vedic mathematics which is applicable for all cases of multiplication. The formulae consists of only one compound word and means "vertically and crosswise". This is very unique and ancient Indian Technique and still one of the dominant method for implementing multiplier. The use of UT sutra for multiplication reduces the delay of a multiplier unit by introducing parallel computing of partial products but it requires more power. The Multiplication for 3-digit decimal numbers is illustrated in Fig 3.

- The partial product generation is indicated by the vertical and crosswise lines which uses AND operation.
- The 2-bit multiplication with two inputs A [1:0] and B [1:0] the following expressions are obtained.

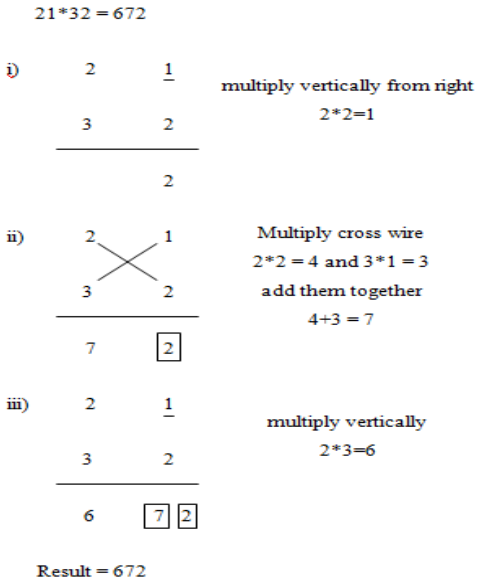
P0= B0A0 →(1)
C1P1= B0A1 + B1A0 → (2)
C2P2= C1 + B1A1 →(3)

- Where, BiAj represents the partial product terms for corresponding bit positions i and j of the operands.
- From partial product addition C1 and C2 are the carries generated and C2P2P1P0 is the result of the multiplication.
- The 4-bit Multiplication, as shown in the line diagram of Fig. , can be carried out by using 2-bit Vedic multipliers
- Any 4-bit binary number A can be represented as AhAl where Ah denotes the most significant 2 bit positions and Al is the least significant 2 bit positions.
- A similar concept is used for the number B which is represented as BhBl.
- Following the same process, two 32 -bit binary numbers A and B, represented as AhAl and BhBl respectively,
- where Ah and Bh are the higher nibbles and Al and Bl are the lower nibbles, can be multiplied using 16-bit multiplier blocks.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 5, May 2020||

Multiplication of two 2-bits using Urdhva-Tiryakbhyam sutra:



C. MODIFIED BOOTH MULTIPLIER

This technique has one great advantage of capable of multiplying both signed and unsigned numbers. The Radix-4 method is capable to reduce the partial products by half by a unique technique. Proposed Booth algorithm uses grouping of the bits and reduces partial products. Hence, it reduces delay. TABLE 1 shows the Digit Grouping.

n-1	n	n+1	Cases
0	0	0	0
0	0	1	+1
0	1	0	+1
0	1	1	+2
1	0	0	-2
1	0	1	-1
1	1	0	-1
1	1	1	0

TABLE 1. Digit Grouping

Operation for Cases:

Case 0: Arithmetic shift of 'X' by 2 bits.

X=X >>> 2;

Case 1: Add 'A' to 'X' fallowed by Arithmetic shift of 'X' by 2 bits.

X=X+A; X=X >>> 2;

Case 2: Add 'A' to 'X' fallowed by Arithmetic shift of 'X' by 2 bits.

X=X+A;X=X >>> 2;

Case 3: Logical left shift by 1 bit (multiplication by 10), then Add 'A' to 'X' fallowed by Arithmetic shift of 'X' by 2 bits.

A=A<<1;X=X+A;X=X >>> 2;

Case 4: 2's complement of 'A' , : Logical left shift by 1 then Add 'A' to 'X' fallowed by Arithmetic shift of 'X' by 2 bits.

A=~A+1;A=A<<1;X=X+A;X=X >>> 2;

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 5, May 2020||

Case 5: 2's complement of 'A' , then Add 'A' to 'X' fallowed by Arithmetic shift of 'X' by 2 bits.

A=~A+1;X=X+A;X=X >>> 2;

Case 6: : 2's complement of 'A' , then Add 'A' to 'X' fallowed by Arithmetic shift of 'X' by 2 bits.

A=~A+1;X=X+A;X=X >>> 2;

Case 7: Arithmetic shift of 'X' by 2 bits.

X=X >>> 2;

The Procedure for implementing the Booth Algorithm are as follows:

Step1:Formation of Booth recoding table Consider the multiplier in block of three, such that each block overlaps the previous block by one bit and form the table using Table I.

Step 2: Booth Algorithm the multiplicand may be added to the partial product, subtracted from the partial product, or left unchanged according to the Booth rule and then shifted.

III. EXPERIMENTAL RESULTS

Example:INPUT→(1EW16EC016) $_{16}$ * (1EW16EC014) $_{16}$

OUTPUT→= (1D819E63A9338000) $_{16}$

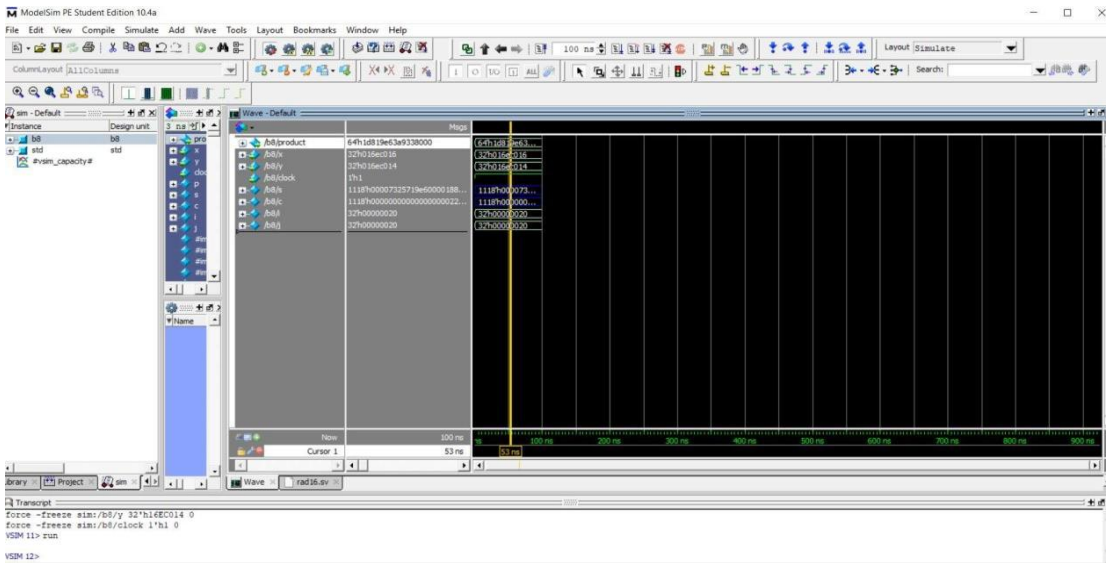


Fig 2 Simulation result of Wallace Tree multiplier

Example: INPUT→$(FFFFFFFF)_{16}$ *$(80000001)_{16}$

OUTPUT→=$(800000007FFFFFFF)_{16}$

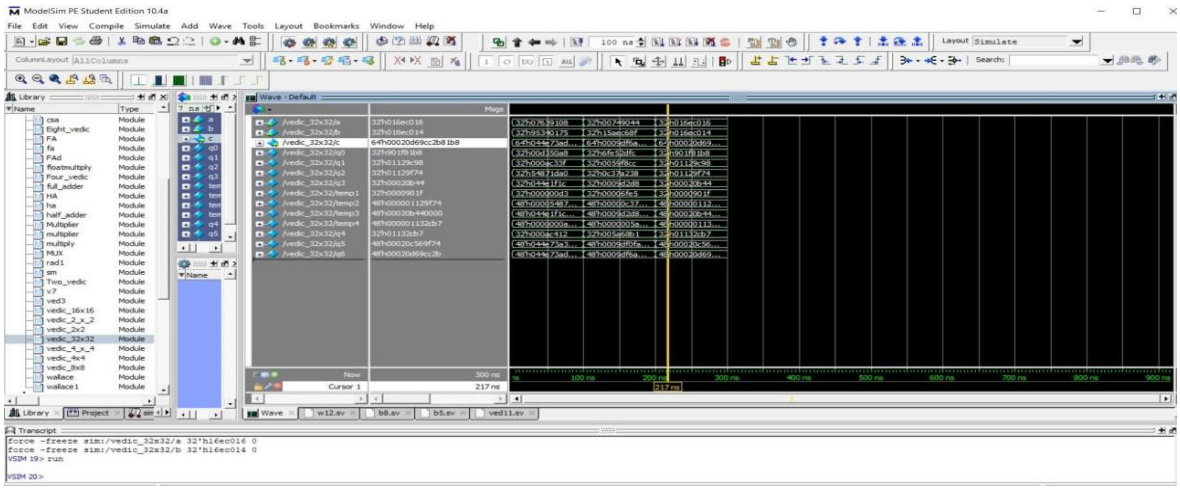


Fig 3. Simulation result of Vedic (U.T Sutra) Multiplier

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 5, May 2020||

Example: INPUT→$(2523092534)_{10}$ * $(2575722496)_{10}$

OUTPUT→= $(6498786199313444864)_{10}$

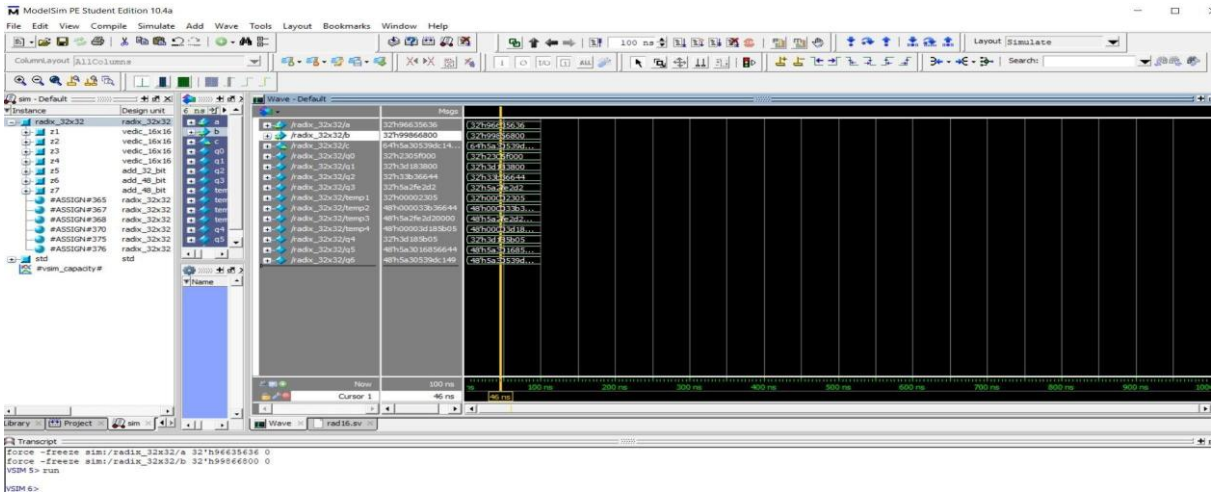


Fig 4. Simulation result of Radix-4 Modified Booth Multiplier

Fig2 and 3 gives the simulation result of wallace and vedic(U-T sutra) 32 bit multiplier. Using Modelsim software And fig 4 gives simulation result of Radix-4 Modified Booth multiplier of 64 bit output.

Multiplier	**Speed (ns)**
Wallace Tree	70.15
Vedic	45.38
Modified Booth	23.66

Table 2 Delay comparison of Multipliers

Names	**Used**	**Total**	**Percentage**
Number of Slice	1595	3584	44
Number of 4 inputs LUTS	2724	7168	38
Number of bounded IOBs	128	141	90

Table 3 Area used in Wallace Tree Multiplier

Names	Used	Total	Percentage
Number of Slice	45	3584	1
Number of 4 inputs LUTS	62	7168	1
Number of bounded IOBs	96	141	68

Table 4 Area used in Vedic Multiplier

Names	Used	Total	Percentage
Number of Slice	101	3584	2
Number of 4 inputs LUTS	181	7168	2
Number of bounded IOBs	96	141	68

Table 5 Area Comparison in Modified Booth Multiplier

Table 2 gives the comparison of delay for wallace Tree, vedic, Modified Booth multiplier. And Table 3 , Table 4 ,Table 5 gives the information of area used in Wallace Tree,Vedic multiplier,Modified Booth Multiplier in terms of Number of slice,Number of 4 input LUTS and Number of bounded IOBs.

On-Chip	Used	Available	Utilization (%)
Logic	2732	7168	38
IOs	128	141	91

Table 6 On-Chip power summary for Wallace Tree Multiplier

On-Chip	Used	Available	Utilization (%)
Logic	62	7168	1
IOs	96	141	68

Table 7 On-Chip power summary for Vedic Multiplier

On-Chip	Used	Available	Utilization (%)
Logic	3161	7168	44
IOs	128	141	91

Table 8 On-Chip power summary for Modified Booth Multiplier

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 5, May 2020||

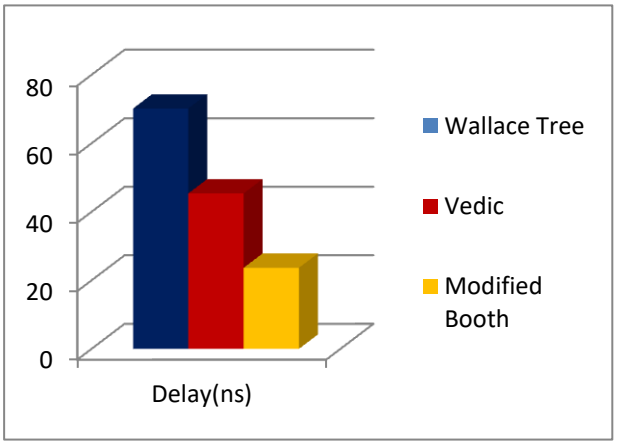


Fig 5 comparison graph of delay(ns)

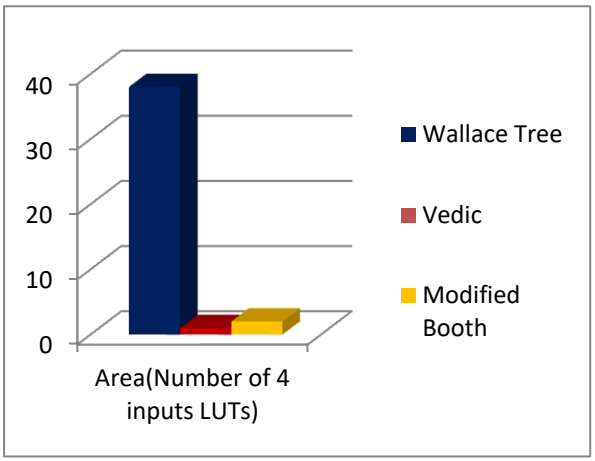


Fig 6 Comparison graph of Area in terms of number of Input LUTs

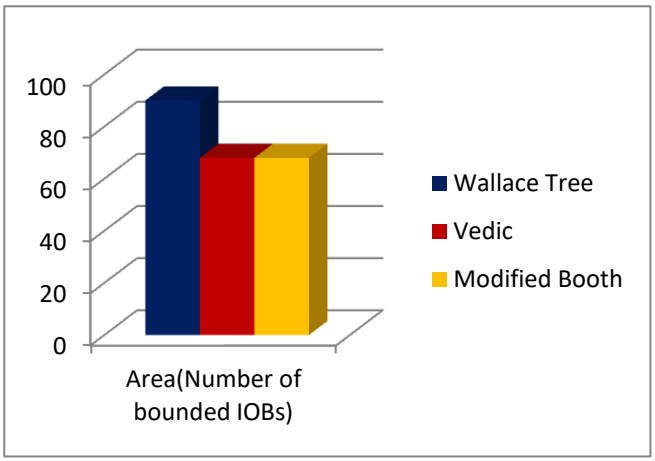


Fig 7 Comparison graph of Area in terms of number of bounded IOBs

Fig 6 and 7 gives the comparison graph of area in terms of number of 4 inputs LUTs and number of bounded IOBs. Due to the trade-off among delay, area and power, as delay is lesser in Modified Booth Multiplier, power and area are more when compared to Wallace Tree Multiplier and Vedic Multiplier.

Table 6, Table 7 and Table 8 gives the information about power utilization (percentage) in terms of Logic and Ios for Wallace Tree, Vedic(using U-T sutra) and Modified Booth(Radix-4) multiplier

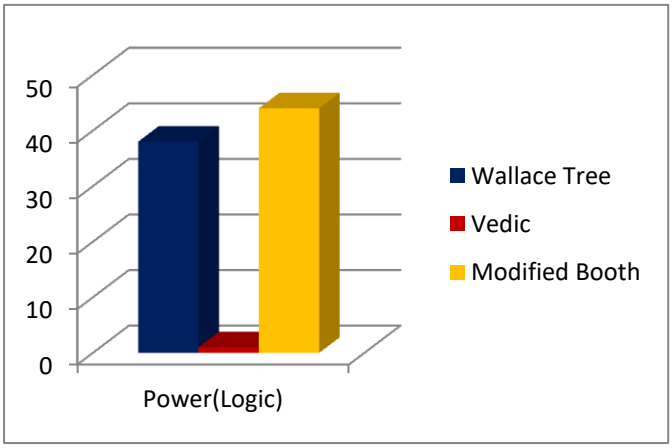


Fig 8. Comparison graph of Power in terms of Logic

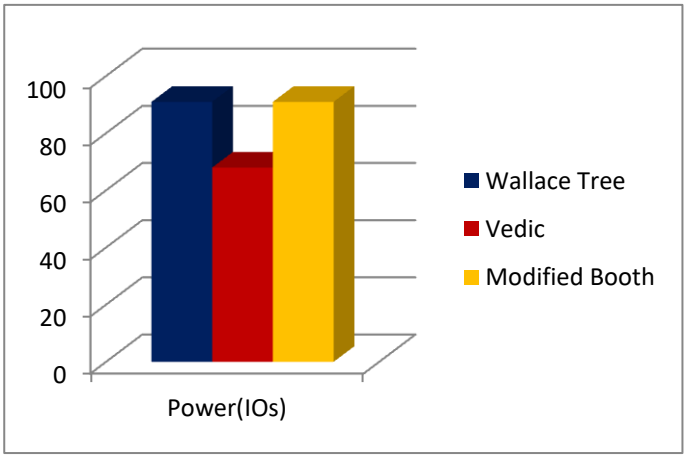


Fig 9. Comparison graph of Power in terms of IOs

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 5, May 2020||

Fig 8 and 9 gives the comparison graph of Power in terms of Logic and IOs.

IV. CONCLUSION

In this paper, three multipliers discussed above are compared based on delay, area and power consumption. Wallace Tree Multiplier is takes relatively more area because number of cells required for implementation is more. Modified Booth is well suited for lesser delay applications. However because of the trade-off factor.based on the application its wise to implement the required method. Vedic method using UT sutra is found to be relatively better but takes more delay compared to Modified Booth Multiplier where partial products were decreased to half. Thus, a multiplier should be selected depending of the performance requirements and the nature of the applications.

V . FUTURE SCOPE

Generally code is implemented for specified n*n multiplication. However a general architecture is required which can perform multiplication for all set of inputs. Existing methods fallows trade-off factor, a new method which can have control over all parameters.

REFERENCES

[1]https://www.researchgate.net/post/what_is_the_basic_need_of_Vedic_multiplier_in_the_field_of_Engineering_and_in_the_industry_if_it_be_then_suggest_the_associated_fieldsaccessed on23-10-2019

[2] Shahebaj Khan, Sandeep Kakde , Yogesh Suryawanshi's-VLSI Implementation ofReduced Complexity Wallace Multiplierhttps://ieeexplore.ieee.org/document/6724141

[3] Eppili Jayak and Chitambra Rao's- Power, Area And Delay Comparision Of Different Multipliers (Volume 5, Issue 6, June 2016)http://ijsetr.org/wp-content/uploads/2016/06/IJSETR-VOL-5-ISSUE-6-2093-2100.pdf

[4] Karthik Naregal, Chandu Y, pratan k hebbar's Design and Implementation of High Efficiency Vedic Binary Multiplier Circuit based on Squaring Circuits . 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)

[5] Siva.S.Sinthura, Assistant Professor, Afreen Begum, B.Amala , A.Vimala, V.Vidhya Aparna's Implemenation and Analysis of different 32-bit multipliers on aspects of Power, Speed and Area (11-12 may2018)https://ieeexplore.ieee.org/document/8553859

[6] Priyanka.Mand Balamurugan. V-Design and performance analysis of a High speed MAC using different multipliers(2-3 sep 2015)https://ieeexplore.ieee.org/document/7433795

[7] D.Kamaladevi and M.Saraswathi's - Design of an efficient high speed radix-4booth multiplier for signed and unsigned numbers, 04 October 2018

[8] "The Evolution of Forth" by Elizabeth D. Rather et al. [1][2]

[9] "Interfacing a hardware multiplier to a general-purposemicroprocessor"

[10] M. Rafiquzzaman (2005). Fundamentals of Digital Logic and Microcomputer Design.John Wiley & Sons. p. 251. ISBN978-0-47173349-2.

[11] KrishnaKant(2007).MicroprocessorsandMicrocontrollers:Architecture,Programmingand System Design 8085, 8086, 8051, 8096. PHI Learning Pvt. Ltd. p. 57. ISBN9788120331914.

[12] Krishna Kant (2010). Microprocessor-Based Agri Instrumentation. PHI Learning Pvt. Ltd. p. 139. ISBN 9788120340862.