ISSN: 0976-3104
ISSUE: Engineering and Technology

ARTICLE

A STATE-OF-THE-ART STUDY ON MULTIPLIERS: ADVANCEMENT AND COMPARISON

Kush Kumar[1], Vishesh Tyagi[1], Himanshu Kukreja[1], Shaveta Thakral[1], Mohit Verma[1,2*]

[1]*Department of Electronics and Communication Engineering, Faculty of Engineering & Technology, Manav Rachna International Institute of Research and Studies, Faridabad, INDIA*

[2]*Accendere KMS-CL Educate Ltd., New Delhi, INDIA*

ABSTRACT

In modern era digital signal processor are the crucial components of the communication system setup. The essential part of the digital signal processors are multipliers, which helps to control the communication speed and also plays key role in other various applications such as Image Processing. Real time multimedia applications necessitate high speed computations. The key arithmetic operation i.e. Multiplication process depletes most of the time and hardware resources of a processor among all the arithmetic operations. Therefore, it necessitates a fast multiplier to be designed for enhancing the system performance. Procedure such as multiply, accumulate and inner products are the frequently used computation intensive arithmetic functions. These functions are applied to process many computations such as Fast Fourier Transform (FFT), filtering and convolution. These multiplications based calculations determines the instruction cycle time of the most algorithms and dominates the execution time of the digital signal processor. Currently, high speed processing devices are one of the primary demands which developed the necessity of higher throughput operational devices. Hence fast, reliable and efficient multiplier design is essential. Also the multipliers are more power consuming devices. As the portable, battery operated systems are the necessity nowadays due to the mobility, the power consumption is one of the major design constraints. Though multiplier is also a complex circuit designed device and consumes greater area. Therefore, it is imperative to design compact and efficient multipliers with less power dissipation. Again, the multiplier performs multiplication operation on unsigned numbers only. Thus, modern computer requires a committed and rapid multiplier unit, which can operate both types of numbers i.e. signed and unsigned. This paper presents a comprehensive study on different multipliers specifically, Array multiplier, Booth multiplier, Modified booth multiplier, Wallace tree multiplier, Modified Booth-Wallace tree multiplier and Vedic multiplier based on their operational procedures and working principals along with the advantages and limitations. A comparative analysis is also takes place on various performance parameters of these multipliers such as speed, area, power utilization and circuit complexity.

KEY WORDS
Array multiplier, Modified booth multiplier, Wallace tree multiplier , modified Booth-Wallace tree, Vedic multiplier

Received: 9 February 2018
Accepted: 30 March 2018
Published: 5 April 2018

***Corresponding Author**
Email:
mohit.verma@accendere
.co.in
Tel.: +91-9661910380

INTRODUCTION

Nowadays, digital signal processing (DSP) systems are essential to improve the quality of digital signals and one of the major components used in DSP systems are multipliers. As reflected, multipliers contributed for the multiplication process. Since it contains a large amount of computation, therefore high speed and greater efficiency are highly required. There are several approaches such as decrease the delay time, lessen the number of partial products, reduce the processing time of accumulation of the partial products, decrease the number of stages to enhance the overall speed of the computation, which directly improves the efficiency of the device along with to advance the multiplication performance [1], it is also essential to take care off some important factors such as compactness, consumption of power speed, area, regularity of layout etc. Every multiplier has worked on a set of defined instructions: Algorithm. There are various algorithms such as Add & Shift, Booth algorithm, Modified booth algorithm, Wallace tree algorithm, Basic hardware algorithm etc. This paper presents the comprehensive study, analysis and comparison of several multipliers such as Array multiplier, Booth multiplier, Modified Booth multiplier, Wallace multiplier, modified booth Wallace multiplier and Vedic multiplier, on the basis of Power, Speed and Area.

ARRAY MULTIPLIER

Array multiplier is a regular shaped multiplier based on “add & shift” algorithm. This algorithm follows standard ‘add and shift’ operation for computation. Multiplicands are multiplied to get the partial products and in each turn, a single bit of multiplier is shifted as per their bit order and finally added at the last stage. Here the number of partial products and multiplier bits are same; hence the number of components and computation stages increases. Due to this, Array multipliers are large in size and having more delay time along with the power consumption, which make them less efficient and complex structured multipliers [2] [3].

Array multiplier’s working procedure is discussed with an appropriate example of 4-bit array multiplier as follows:

Assume two 4-bit data for the multiplication are ‘A3 A2 A1 A0’ and ‘B3 B2 B1 B0’. The process of multiplication is shown in [Fig.1].

4x4 array multiplier needs16 AND gates, 4 half adders and 8 full adders. It requires 12 addresses. In general ‘mxn’ array multiplier requires the product of ‘m and ‘n’ of AND gates, ‘n’ numbers of half adders, product of ‘(m-2)’ and ‘n’ full adders and product of ‘(m-1)’ and ‘n’ addresses to complete the operation. The schematic illustration of 4x4 array multiplier is shown in [Fig. 2].

IIOAB JOURNAL
ISSN: 0976-3104

				A3	A2	A1	A0	
			X	B3	B2	B1	B0	
+			C B1xA3	B0xA3	B0xA2	B0xA1	B0xA0	
+		C B2xA3	sum B2xA2	sum B2xA1	sum B2xA0	sum		
+	C B2xA3	sum B2xA2	sum B2xA1	sum B2xA0	sum			
C	sum	sum	sum	sum				
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	

Fig.1: Array multiplication structure of two 4-bit data

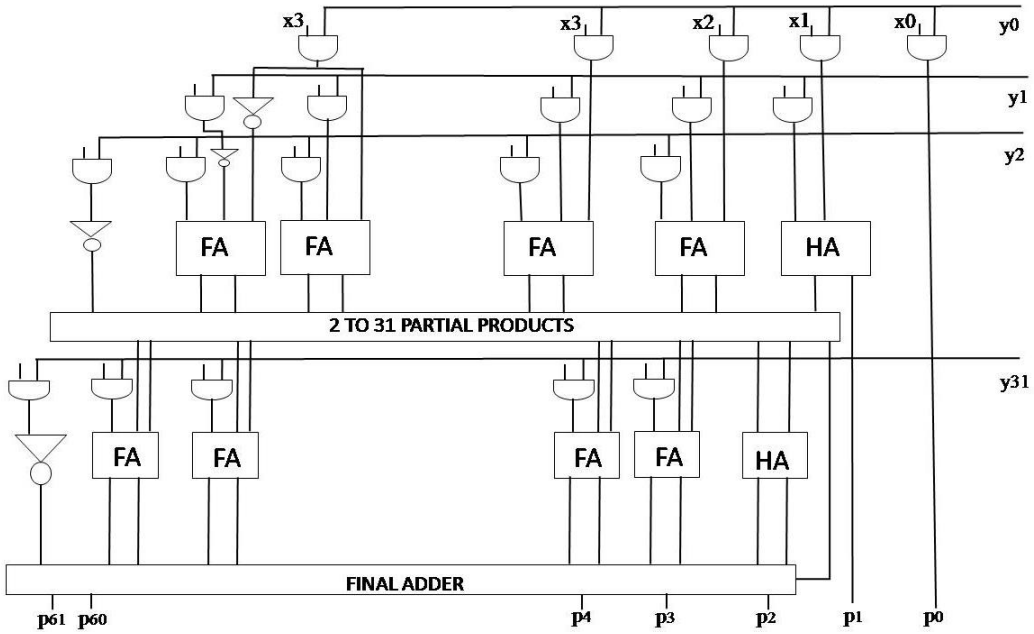

Fig. 2: 32-bit array multiplier [63]

This conventional architecture consumes much power and time. Also it is complex in structure due to more number of components is used to make it. Various architectures have presented to enhance the efficiency and response time. In the year of 1993, Min C Park et. al. proposed a new design using dual tree structure. Due to this dual tree structure technique, the multiplier processing speed is increased twice as compared to the conventional one. However, the silicon area increases by 30% [4]. It was also suggested by the researchers that the performance of the multiplier could be improved by using low power array multiplier techniques in terms of delay & power dissipation [5]. Junghwan Choi et. al. in 2000, minimize the power consumption of the array multiplier with the help of Partially Guarded computation technique by 44%. This technique also helps to reduce the delay and overhead area by 3% and 30-36% respectively. Several researchers contributed their works for the development of the Array multiplier, which has been coated in the following table [Table 1].

BOOTH MULTIPLIER

The deficiency with the Array multiplier is its execution speed. Therefore, a new multiplier named Booth multiplier is developed to improve the performance by reducing the number of iterations. This multiplier is worked on Booth algorithm, which was invented by Andrew Donald Booth in 1950 at Birkbeck College in Bloomsbury, London. In this multiplier three bits are scanned at the same time. Out of these three bits,

present pair is made, which includes two bits and the higher bit of an adjacent lower order pair belongs to the third bit. After examined these three bits, booth logic is used, which converts the triplets into a set of five control signals. These five control signals are then used to perform the operations and controlled by using adder cells [11] in the array. The operation procedure signed numbers by booth multiplier is shown below with the help of flowchart [Fig. 3]

Table 1: Summary of findings of array multiplier

Sl.No.	Author Name	Technique Used	Objectives	Advantages	Limitations
1	Joseph Whitehouse et. al. [6]	FinFET models with array multiplier topology and low power Predictive Technology Models (PTM)	To Investigate the static power and delay using low power Predictive Technology Models (PTM)	Enhanced static power reduction in delay and feature size	Further reduction in channel length haven't effect on significant change
2	Zhong-ye Yanga et. al. [7]	Pipeline Techniques	Analysis of time complexity of two's complement	Improvedspeed of systematic performance is approximately twice as compared to non-pipelined multiplication.	Serial adder must be needed
3	S.K. Sahoo et. al. [8]	Delay optimization inter connection	To enhance the operation speed	Less delay time up to 12 bit multiplication	Consume more time for higher bit multiplication
4	S.Srikanth et. al. [9]	Using multiplexers	To reduce the power consumption	Average reduction of power consumption, area and delay by 35.45%, 40.75% and 15.65% respectively	Notsuitable for high power and large area applications.
5	PriyankaSrivastava et. al. [10]	New hybrid adder	Low power and high speed array multiplier	Works on low power and less delay approximately by 24% and 56%	Suffer from voltage swing problem

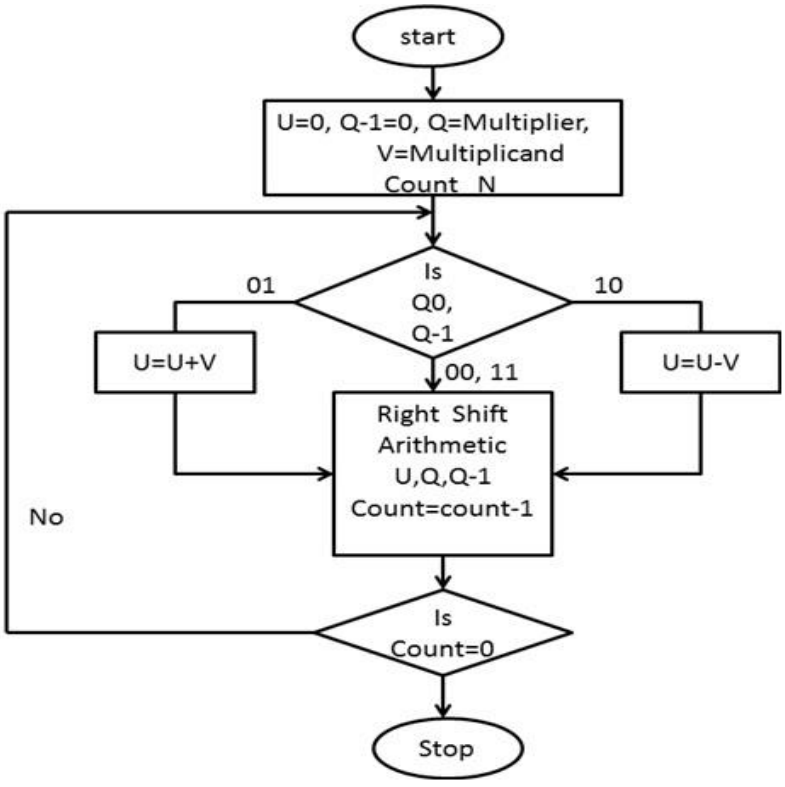

Fig.3: Flowchart of Booth Multiplication process

IIOAB JOURNAL
ISSN: 0976-3104

In the above flowchart, three conditions are applied for multiplication:
1) If Q0=0 and Q-1=1 then add V in U and perform the right shift.
2) If Q0=1 and Q-1=0 then add V, subtract U and perform the right shift.
3) If Q0= Q-1=0 or Q0=Q-1=1 then perform right shift only.

At first, this multiplier examined the two least significant bits with various conditions then performed the multiplication and repeated the same step up to 4 times. Then the sum of two partial products is accumulating, taking the product register as an accumulator. The power consumed of the multiplier is more due to the involvement of large number of adder cells to perform the multiplication operation with the help of this method. Therefore, efficiency is a crucial issue for this system.

Table 2: Summary of findings of booth multiplier

Sl.NO	Author Name	Technique Used	Objectives	Advantages	Limitations
1	Wen-QuanHe.et al. [12]	Probability and computer simulation (PACS).	To developed a high accuracy dynamic error-compensation circuit for fixed-width Booth multipliers	Highly accurate and area effective	Power consumption of proposed PACS multiplier is higher
2	A N Nagamani et. al. [13]	Garbage Cost and Ancillary inputs	To present a design for a Reversible Radix-4 Booth Multiplier for DSP application	The proposed design is capable of both signed and unsigned multiplication. Lower heat dissipation. The circuit area reduces by almost half.	High Quantum cost and delay
3	Daichi Okamoto et. al. [14]	Ring Oscillator	To make a Serial Booth Multiplier	High working frequency and low power consumption	Booth encoding is needed to reduce partial products
4	Rahul Shrestha et. al. [15]	Additional clock gating and resource sharing,	To present an area-efficient low-power architecture for configurable booth multiplier.	The proposed multiplier architecture requires 43.12% of lower area and consume 75.65% of less power as compared to previous one	Slight increase in latency
5	Jakia Sultana et. al. [16]	Reversible mode	To develop a design methodology for the realization of Booth's multiplier	Both signed and unsigned multiplications can be done	Need to examine this proposed logic on Redix-4 approach

MODIFIED BOOTH MULTIPLIER

To enhance the efficiency of the Booth multiplier, several modifications has done on it such as modified booth encoder and selector technique to rearrange and reduce partial products [17]. The booth encoder presents here, performs various steps simultaneously, therefore, the speed of the multiplier increases and due to this method the number of gate count reduces and hence the multiplier's performance is improved. It also uses modified radix4 booth algorithm, when the operands are greater than or equal to 16 bits. The area of the multiplier circuit is also get shortened by using this algorithm. In this algorithm the number of partial products, those are to be added is reduced by encoding 2's compliment. In this same algorithm, the multiplier bits are divided into 3 blocks and it is divided into 4 blocks, when radix8 algorithm is applied. These divided blocks are rearranged in such a way that each block overlaps the other by 1 bit. Also the computation time and the logarithm of the word length of operands are proportional to each other and due to this way the number of partial products reduces. The partial products are generated from the multiplied and encoded multiplier with the help of PPG (partial product generator). Then partial product reduction tree is used to add these partial products. Again the results are added using carry propagate adder (CPA). [Fig. 4] represents the block diagram of this process/multiplier.

From time to time several modifications have been done to improve the performance of this multiplier. At year 2000, Wen Chang Yehet.al. developed a design using a new developed booth encoding scheme (MBE) [18-19]and enhanced the speed of the multiplier up to 25%. At the same year Fayej Elguibaly [20] developed a parallel multiply accumulate hardware using the modified booth algorithm [19] which is three times quicker operator as compared to other standard parallel MAC units. Again at 2007, Zhou Shun et al. (2007) [21] designed a multi precision reconfigurable Radix -4 booth multiplier which can be cascaded to comply with the different input length which improved the performance in terms of delay & area. To increase the performance of FAM (fused add multiply) Kostas Tsoumanis et al. in the year of 2014 [22]

incorporated structured and efficient modified booth recording technique to reduce power consumption, hardware complexity and critical delay. Some of the progressive development of this multiplier is listed below in [Table 3].

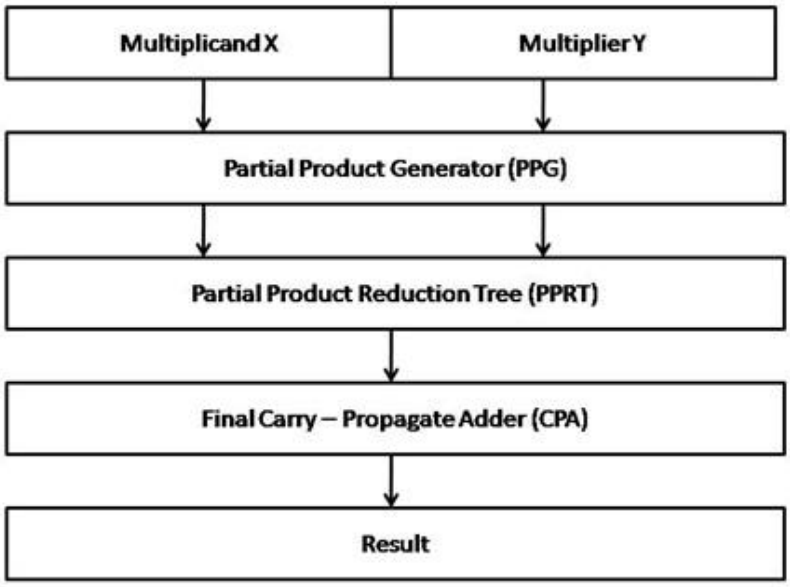

Fig. 4: Block diagram of Modified Booth multiplication process.

.....

Table 3: Summary of findings of modified booth multiplier

Sl.No.	Author Name	Technique used	Objectives	Advantages	Limitations
1	Kei-Yong Khoo et. al. [23]	Modified Booth encoding Algorithm	Improved booth encoding for low power multipliers	Reduce the power dissipation	Unit delay of canonic signed digit is more in proposed multiplier
2	Ravindra P Rajput et. al. [24]	Modified booth encoding (MBE) technique	To design a signed-unsigned Modified Booth Encoding (SUMBE) multiplier	Less hardware and chip area reduces the overall cost and also power dissipation	Low speed due to more number of partial products
3	Babu M. Pranay SrivatsavaJandhyala [25]	Booth encoder and booth decoder	A new architecture design is proposed for an accuracy configurable modified Booth multiplier (ACMBM) with two types of approximate adders, which or can be configured in terms of error during run time	Delay reduction achieved of 15.3% and 15.8% for type I and type II adder. Again power is decreased by 5% in type I and increased by 2% in type II	Error configuration depends on the approximate adders working at adding operation on partial products
4.		2's complement and 1's complement representation	To design efficient 1's Complement Modified Booth multiplier	This multiplier is efficient than modulo 2(power n) -1 modified booth multiplier.	Aarea and power complexity
5	Bipinlikhar MsSakshi [27]	Modified booth algorithm, carry select adder, ripple carry adder	To propose an efficient technique to find 2's compliment for generation of regular partial products	Reduced power modified booth encoding (MBE) multipliers with less occupied area	To reduce the amount of delay a Carry select adder is required

IIOAB JOURNAL
ISSN: 0976-3104

WALLACE TREE MULTIPLIER

Statics show that 70% of instructions and algorithm performs addition and multiplication in microprocessors [28-29]. Therefore the important challenge is to establish a highly quick multiplier for shortening the entire operation time period. In 1964 C. S. Wallace proposed fast parallel multiplying scheme to reduce the partial products/ intermediate steps, which in turns introduced Wallace multiplier. It is a hardware implementation for multiplying two binary numbers. In this multiplier, at first it generates set of bit by bit multiplications for each bit and assigned it with a particular weight. These weights maintain the track of binary digits corresponding to the obtained partial products. These partial products are now divided into set of rows, named reduction layers. The partial products having same weight are combined in a series of reduction layers as shown in [Fig. 5]. Thereafter, it is added with the help of full and half adders. Then the repetition of the process occurs until to get the two last rows. Finally, the final result is achieved with the help of Carry Look Ahead adder.

This multiplier requires many numbers of gates to operate, which make these multipliers bulky and slow. To overcome with this slow processing, Wallace multiplier configured with the parallel AND gates that make partial products simultaneously, which in turns helpful to reduce the execution time and enhance its speed.

In the year 1998, Moises E. Robinsonet.al. modified its design. He inserted counterat the first layer,to manage the reduction process and reduce the delay without increasing the structure complexity [30]. Again Ron S. Waters et. al. reduces the number of half adder and S. Rajaram et. al used parallel prefix adders to fix final adders, which decreased the delay. In 2014, Damarala Paradhasaradhi et al. [31] presented a new structure of Wallace multiplier based on square root carry select adder. In this design common Boolean logic is shared to remove the duplicate adder cells, hence decreases the number of gates which in turn reduced the delay and power consumption of the multiplier. These multipliers are highly used in 3-d computer graphics and high speed floating point processing.

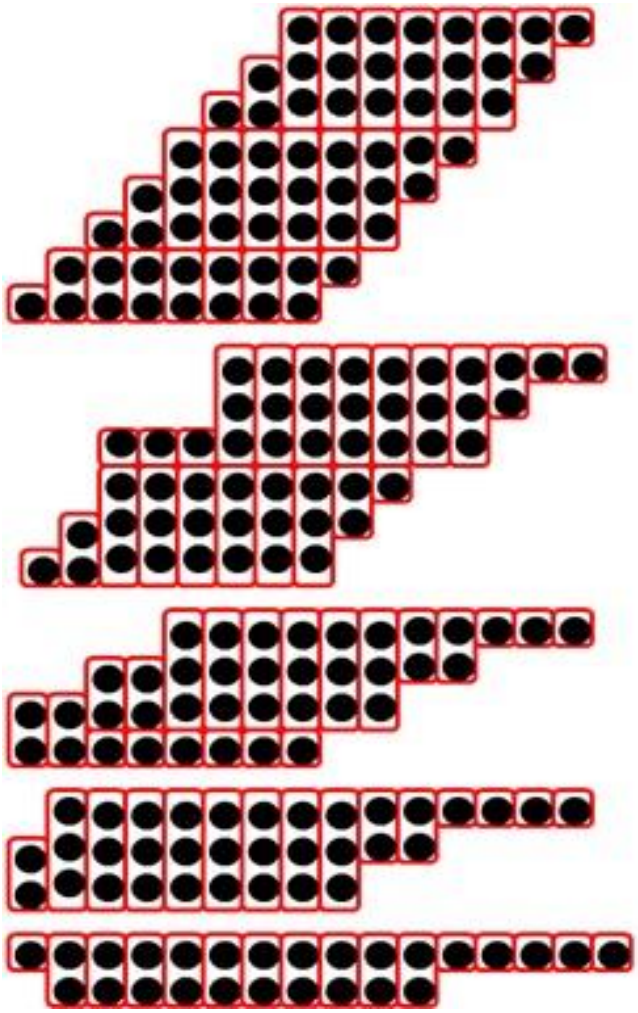

Fig. 5: Multiplication of 8-bit wallace tree

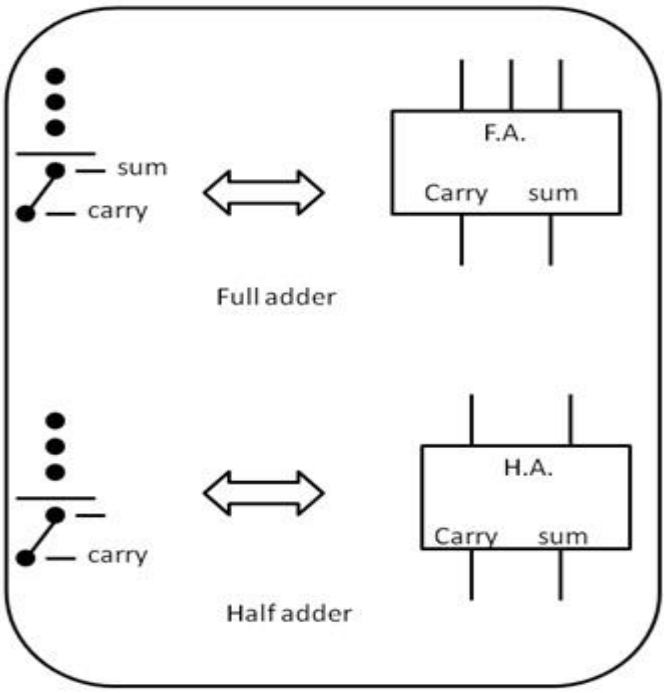

Fig. 6: Representation of half adder and fulladder

Table 4: Summary of findings of wallace tree multiplier

Sl.No.	Author Name	Technique Used	Objectives	Advantages	Limitations
1	Shahabaz Khan et. al. [32]	Energy Efficient CMOS based full adder	Reducing the intricacy of Wallace Multiplier	Reduced area and power.	No improvement in terms of delay as compared to the standard Wallace tree multiplier
2	ShahzadAsif et. al. [33]	algorithm uses high speed 7:3, 6:3, 5:3, and 4:3 counters	to construct the counter based Wallace tree multipliers for higher speed	Speed enhanced up to 22% as compared to the traditional Wallace multiplier	Not suitable for low speed applications
3	DamarlaParadhasaradhi et. al. [34]	Modified Square Root Carry Select-Adder (MCSLA), Square Root Carry Select Adder using RCA and Carry Select Adder (CSLA)	To proposed an area efficient Wallace tree multiplier	Reduction of delay and area	Delay enhances
4	R. BalaSaiKesava et. al. [35]	Carry select adder(CSLA),Binary to excess one converter (BEC),Square root carry select adder(SQRTCSLA)	To established compact Wallace tree multiplier with the help of CSLA technique	CSLA based Wallace tree multiplier having BEC occupies Less area and memory. It also able to works on low power	CSLA based Wallace tree multiplier having BEC has higher delay than CSLA based Wallace tree multiplier without BEC
5	Kazuteru NAMBA and Hideo [36]	Bit-slice reconfiguration design	To design a Defect Tolerant Wallace Multiplier	Defects can be tolerated through this multiplier	Not applied to barrel shifter

IIOAB JOURNAL
ISSN: 0976-3104

MODIFIED BOOTH WALLACE MULTIPLIER

For larger multiplier such as 32-bit, the performance of Booth algorithm is limited. To overcome this problem, the Modified Booth Wallace multiplier introduced with the help of Wallace multiplier. It is a combination of both Booth and Wallace multiplier. This new designed multiplier contains four key components i.e. booth encoder, partial product generator, Wallace tree and carry look ahead adder [37]. Booth encoder is dedicated to the encoding of multiplier bits by using Radix-4 and Radix-8 algorithm. Then the partial products are produced with the help of multiplicand and encoded multiplier by the partial product generator. Thereafter the Wallace tree operates on these partial products (see section Wallace Tree multiplier). Finally, the result is achieved by the carry look ahead adder. This multiplier contributes to minimize the consumed power and circuitry area as compared to Booth and Wallace multiplier.

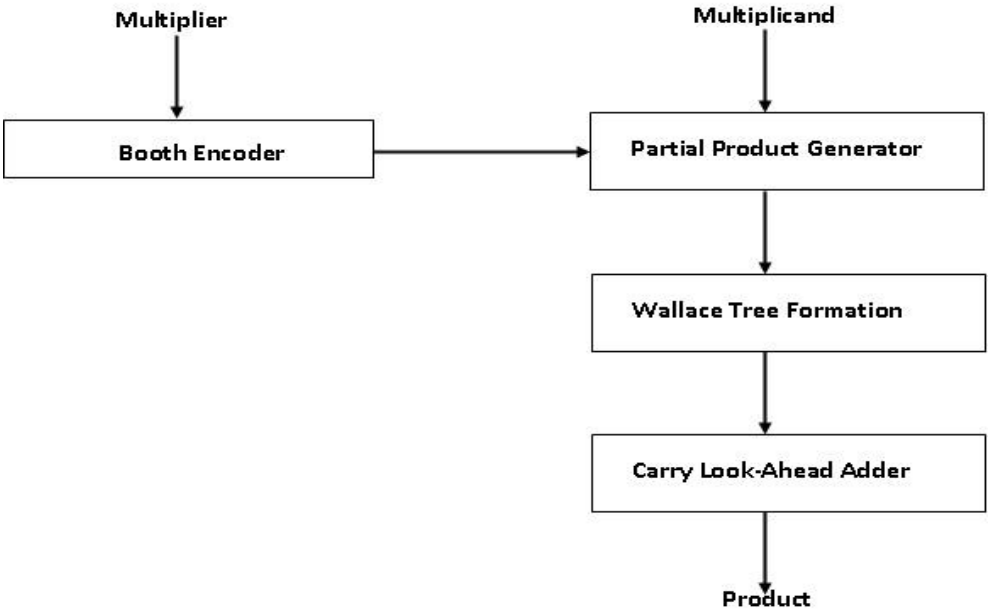

Fig 7: Block diagram of modified booth wallace tree multiplier [64]

In the year of 1993, Jalilfadaviardekani [38] developed a new architecture to optimized the partial product of booth encoded [11, 19] parallel multiplier. According to this, the inputs are encoded into booth equivalent and a Wallace tree [39] and then the partial product are added with the help of carry select adder. To reduce the delay, an algorithm was presented by M J liaoet al. [40] using portioning carry select adder. In this algorithm the carry select address are partitioned into the number of blocks. The algorithm reduces the delay of 9.125 with less than 1% overload. Also in the year of 2006, a parallel complex number multiplier was proposed by Rizalafandecheismail and Razaidi Hussin[41],based on Radix-4 modified booth algorithm and Wallace tree [40].Again M. Jagadeshwar Raoet. al. [42] proposed a new architecture of Wallace multiplier using booth recorder and compressor .This modified multiplier is 67% quicker than previous Wallace tree multiplier and 22% quicker than radix-8 booth multipliers. Recently a new structure of multiplier has developed [43], which divided the multiplier architecture into four different modules. This new multiplier consume less time for computation and also operation independent multiplier.

Table 5: Summary of findings of modified booth wallace multiplier

Sl.No.	Author Name	Technique Used	Objectives	Advantages	Limitations
1	M.-J. Liao et. al. [44]	Carry-select-adder partitioning algorithm	To enhance the performance of Booth-encoded Wallace-tree multiplier	The average delay and area overhead is reduced by 9.12% and 1% with the help of proposed algorithm for multipliers ranges from 16XI6-bit to 64x64-bit.	Power dissipation effects need to be realized
2	JalilFadavi-Ardekani [45]	Optimized Wallace Trees	To define and design the architecture of MxN bit Booth encoded parallel multiplier generator	Fast data paths are achieved using ASIC (standard cell based) designed multipliers	If the number of cells increases, then this will lead to increase in area
3	Rahul D Kshirsagar et. al. [46]	Pipelining	To introduce pipelining system for in-between nodes of the modules	Enhanced speed and computation.	Independent operation cannot be done in a given clock period

4	LiangyuQianet . al. [47]	Utilizing approximate modules in the Booth encoder	To design an approximate Wallace –booth approximate multiplier	Improved efficiency in terms of power consumption, delay and combined metrics	Moderate loss in accuracy
5	M Jagadeshwar Raoet. al. [48]	Booth algorithm, 5:2, 4:2, and 3:2 compressor adders	To reduce latency and power consumption of the Wallace tree multiplier	The proposed architecture is around 67% faster	No effect on power improvement

VEDIC MULTIPLIER

In the modern world the Vedic mathematics is based on 16 aphorisms and 12 corollaries. These formulations are selected from Atharva Ved by Swami Bharati Krishna Tirtha (1884-1960). Thereafter, the former Jagadguru Sankaracharya developed and presented the techniques to modify the principles in these selected sutras and sub-sutras. Among all of these sutras and sub-sutras, the Nikhilam Navatashcaramam Dashatah and Urdhva–Tiryagbhyam sutras are used for the multiplication purpose. These Vedic mathematic techniques when implemented for the multiplication, showed very good results in terms of saving computational time. Therefore it is concluded that the multiplier design integrated with Vedic mathematic techniques based upon "Urdhvatriyagbhyam" (vertical and cross wise algorithm) sutra [49] enhanced the speed of multiplication operation. The methodology for 4x4Vedic mathematics is given below to clarify the procedure:

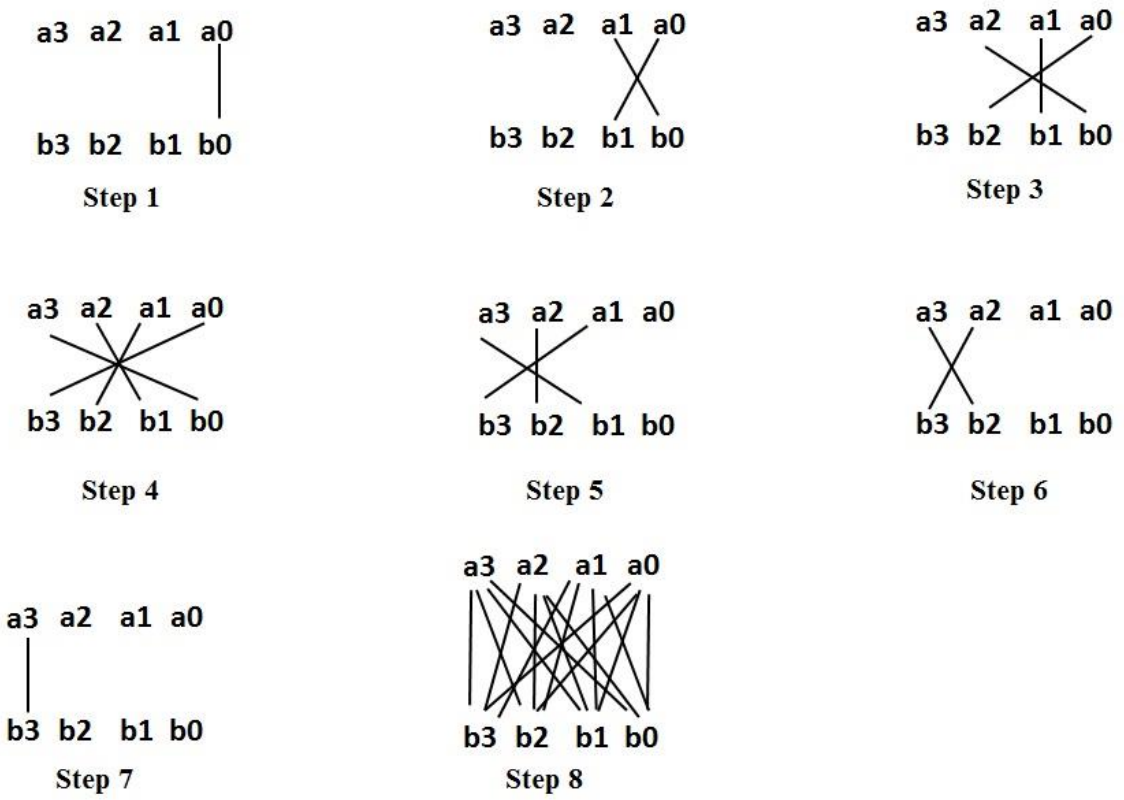

Fig. 8: steps of vedic multiplication.

Here, in the first row a3a2a1a0 represented the multiplicand bits row and multiplier bits are represented by b3b2b1b0 lies in a second row. At first in the step 1, the least significant bits having equal weights are added hence the addition of 0^{th}bits take place i.e., 'a0' and 'b0'. Then the cross adding are done as shown in steps 2, 3 and 4. Again the same process have done but from the most significant bits side as indicated in steps 5, 6 and 7. These whole processes (steps) are continuously repeated until the final output has come out.

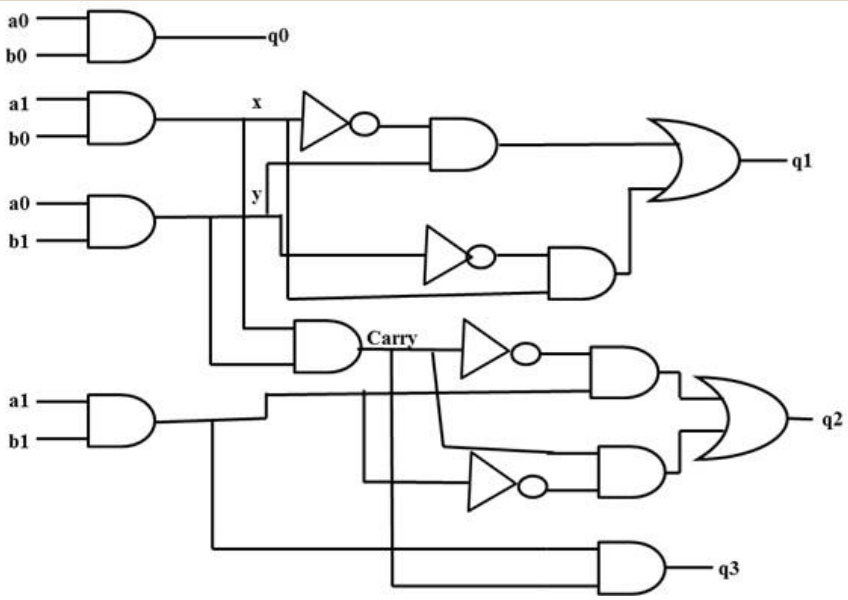

Fig. 9: Block diagram of 2x2 vedic multiplier

.....

The idea of implanting/using vedic mathematics as the key function of multipliers, attract the attention of all over the world due to its enhanced performance. In 2012, Vaijyanath Kunchigi et al. designed a pipeline architecture using Vedic mathematics. This proposed architecture consists of 3 stages:-

1st stage consist of 4 bits Vedic multiplier units,2nd stage is for the parallel products reduction and the 3rd stage is the addition of those parallel products .This projected multiplier shows high performance in the area of speed and power consumption as compared to other multipliers (Array multiplier, Booth multiplier etc.). Also this same architecture could be functional for larger word length input such as 16 bit, 32 bit, 64 bit etc. multiplier. In 2013, using another Vedic sutra named modified Nikmilamsutra Pavan Kumaret. al. [50] implemented a multiplier with the help of parallel shifter. This new modified Vedic multiplier enhanced the speed highly by reducing the delay up to 45% as compared to the Array and Booth or conventional Vedic multiplier. Thereafter R. Anganaet. al. [51] proposed and developed a new architecture of Vedic multiplier by combining it with the Kogge Stone adder [52], a parallel prefix form of carry look ahead adder. This new architecture provides one of the fastest multiplier. Again to minimize the consumed power, another architecture was proposed by Hardik Sangani et.al.[53]. This architecture is based on Vedic multiplication and adiabatic logic .They proposed a Vedic multiplier build-up on differential cascade pre-resolve adiabatic logic (DCPAL) and reduces the amount of consumed power by 57% and 68.5%, as compared to the conventionally designed Vedic and Array multiplier on traditional CMOS respectively. In the later years, various researchers are continuously working to enhance the performance of the device using vedic multipliers along with other techniques embedded in it. R. Katreepalli et. al. [54] introduced a new design of vedic multiplier which is more efficient in terms of power, delay and area using adaptable manchester carry chain adder. G. V. Nikhil et. al. [55] also proposed same by using kogge-stone adder and reversible logic gates. To enhance the speed of the vedic multiplier D. K. B. Kahar et. al. [56] implemented a new algorithm in the year 2017. At the same year ancient India Vedic mathematic is used to optimized the multithread for long digit multiplier [57] and the vedic multiplier is used to develop a processor for single-path delay feedback pipeline FFT which is highly-speedy [58].

Table 6: Summary of findings of vedic multiplier

Sl.No.	Author Name	Technique Used	Objectives	Advantages	Limitations
1	G.Challa Ram et. al. [59]	Vedic mathematics	To design a high speed Vedic multiplier	Provide minimum delay for multiplication for all numbers	Requirement of BEC(binary to excess code converter) is necessary for utilized memory reduction
2	EktaMasurkar PravinDakhole[60]	Urdhvatriyagbhyam sutra, Adiabatic logic	To optimize vedic multiplier design in terms of high speed and low power useing vedic sutra UrdhvaTriyagbhyam	Low power consumption	Requirement of Adiabatic logic is necessary
3	Ms. G. R. Gokhale et. al.	Carry select adder	To design an efficient multiplier	Requires less area	Delay is more

	[61]		in terms of area and delay		
4	Kunjpriya Morghade PravinDakhole [62]	Build-in self-test (BIST) Technique	To design and implement 4-bit Vedic multiplier along with build-in self-test (BIST)technique for testing multiplier circuit	More efficient Algorithm	
5	K Pranav et. al. [63]	Urdhvatriyagbhyam Sutra	To perform linear convolution	Enhanced speed	Pipelining is needed

All of these discussed multipliers are using efficiently as per the application requirements. From the discussion above a table [Table 7] is shown below to represent the comparison among the multipliers by taking into the account of some very important parameters such as: time delay, power consumption, circuit complexity and area required for simple understanding.

Table 7: comparison table

Multiplier	Speed	Area	Power consumption
Array	Low	Small	Most
Booth	Low but better than Array	Small	Less than Array
modified booth	High	Medium	Less
wallace tree	Higher	Larger	More
modified booth Wallace	Highest	Largest	More
Vedic	Higher than Wallace tree	Larger than Wallace tree	More than Wallace tree

CONCLUSION

To enhance the performance of the modern communication setup, multipliers with higher efficiency are extremely required. Fast multipliers improved the speed of computation, which also increases the performance of the other digital applications such as image processing. The main parameters which need to be take care off in case of multipliers are delayed time, power consumption, circuitry complexity and area requirement.

All the multipliers discussed above are efficient in terms of these all performance parameters. Out of all, the array multiplier is the simplest due to its simple circuitry, which leads to less space usage. Although, this multiplier suffers with low speed and maximum power consumption. The fastest multiplier among all is the modified booth Wallace tree multiplier by taking the advantages of both multipliers: modified booth multiplier and Wallace multiplier. In this multiplier the number of partial products is minimized to either half or one by third of the number of multipliers bit by using radix 4 algorithm and radix-8 algorithms respectively. The Wallace tree multiplier, where the overall speed of the accumulation increases due to using carry save adder (CSA) has occupies the largest area. Here by minimizing the number of partial products and examine more than one partial products at the same time, the speed is further enhanced and these techniques also make the system more accurate. One of the fastest and less power consuming multiplier is Vedic multipliers, which is based on the vedic mathematical formulations. It is proved by several researchers that the vedic multiplier reduces the delay time and power consumption by approximately 45% and 57% as compared to the array multiplier.

CONFLICT OF INTEREST
None

ACKNOWLEDGEMENTS

Authors would like to express the gratitude to the Research Mentors of Accendere Knowledge Management Services Pvt. Ltd. for their comments on an earlier version of the manuscript. Although any errors are our own and should not tarnish the reputations of these esteemed persons.

FINANCIAL DISCLOSURE
None

REFERENCES

[1] Parate PG, Patil PS, Subbaraman S. [2008] Asic implementation of 4 bit multipliers. In Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on (pp. 408-413). IEEE.

[2] Thapliyal H, Arabnia HR. [2004] A Novel Parallel Multiply and Accumulate (V-MAC) Architecture Based on Ancient Indian Vedic Mathematics. In ESA/VLSI (pp. 440-446)

[3] Morris Mano, Computer System Architecture, 1993. 3rd edition, PHI, 346- 347.

[4] Park MC, Lee BW, Kim GM, Kim DH. [993] Compact and fast multiplier using dual array tree structure. In Circuits and Systems, 1993, ISCAS'93, 1993 IEEE International Symposium on (pp. 1817-1820). IEEE

[5] Mahant-Shetti SS, Balsara PT, Lemonds C. [1999] High performance low power array multiplier using temporal tiling. IEEE Transactions on very large scale integration (VLSI) systems, 7(1):121-124

[6] Whitehouse J, John E. [2014] Leakage and delay analysis in FinFET array multiplier circuits. In Circuits and Systems (MWSCAS), 2014 IEEE 57th International Midwest Symposium on IEEE pp. 909-912.

[7] Yang ZY, Xiao JQ. [2011] The design and simulation of array multiplier improved with pipeline techniques. In Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on Vol. 8: 4326-4329. IEEE

[8] Sahoo SK, Shekhar C. [2011] Delay optimized array multiplier for signal and image processing. In Image Information Processing (ICIIP), 2011 International Conference on (pp. 1-4). IEEE

[9] Srikanth S, Banu IT, Priya GV, Usha G. [2016] March). Low power array multiplier using modified full adder. In Engineering and Technology (ICETECH), 2016 IEEE International Conference on (pp. 1041-1044). IEEE

[10] Srivastava P, Vishant V, Singh RK, Nagaria RK. [2013] Design and implementation of high performance array multipliers for digital circuits. In Engineering and Systems (SCES), 2013 Students Conference on (pp. 1-5). IEEE.

[11] Booth AD.[1951] A Signed Binary Multiplication Technique, Qrt J Mech App Math, 4(2): 236–240

[12] He WQ, Chen YH, Jou SJ. [2015] High-accuracy fixed-width Booth multipliers based on probability and simulation. IEEE Transactions on Circuits and Systems I: Regular Papers, 62(8): 2052-2061

[13] Nagamani AN, Nikhil R, Nagaraj M, Agrawal VK. [2016] Reversible Radix-4 booth multiplier for DSP applications. In Signal Processing and Communications (SPCOM), 2016 International Conference on (pp. 1-5). IEEE

[14] Okamoto D, Kondo M, Yokogawa T, Sejima Y, Arimoto K, Sato Y. [2016] A Serial Booth Multiplier Using Ring Oscillator. In Computing and Networking (CANDAR), 2016 Fourth International Symposium on (pp. 458-461). IEEE

[15] Shrestha R, Rastogi U. [2016] Design and Implementation of Area-Efficient and Low-Power Configurable Booth-Multiplier. In VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), 2016 29th International Conference on (pp. 599-600). IEEE

[16] Sultana, J, Mitra, S. K, & Chowdhury, A. R. (2015, January). On the analysis of Reversible Booth's Multiplier. In VLSI Design (VLSID), 2015 28th International Conference on (pp. 170-175). IEEE

[17] Chu T A. [2002] U.S. Patent No. 6,393,454. Washington, DC: U.S. Patent and Trademark Office.

[18] Yeh WC, Jen CW. [2000]. High-speed Booth encoded parallel multiplier design. IEEE transactions on computers, 49(7):692-701

[19] MacSorley OL. [1961] High-speed arithmetic in binary computers. Proceedings of the IRE, 49(1): 67-91

[20] Elguibaly F. [2000] A fast parallel multiplier-accumulator using the modified Booth algorithm. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 47(9): 902-908.

[21] Shun Z, Pfander O A, Pfleiderer HJ, Bermak A. [2007] A VLSI architecture for a run-time multi-precision reconfigurable Booth multiplier. In Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on (pp. 975-978). IEEE

[22] Tsoumanis K, Xydis S, Efstathiou C, Moschopoulos N, Pekmestzi K. [2014] An optimized modified booth recoder for efficient design of the add-multiply operator. IEEE Transactions on Circuits and Systems I: Regular Papers, 61(4): 1133-1143

[23] Khoo KY, Yu Z, Willson AN. [1999] Improved-booth encoding for low-power multipliers. In Circuits and Systems, 1999. ISCAS'99. Proceedings of the 1999 IEEE International Symposium, 1: 62-65. IEEE

[24] Rajput RP, Swamy MS. [2012] High speed Modified Booth Encoder multiplier for signed and unsigned numbers. In Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on (pp. 649-654). IEEE

[25] Pranay BM, Jandhyala S. [2015] Accuracy configurable modified Booth multiplier using approximate adders. In Nanoelectronic and Information Systems (iNIS), 2015 IEEE International Symposium on (pp. 281-285). IEEE

[26] Pekmestzi K, Efstathiou C. [2016] Design of Efficient 1's Complement Modified Booth Multiplier. In Digital System Design (DSD), 2016 Euromicro Conference on (pp. 238-243). IEEE

[27] Likhar B. [2013] Design and Comparison of Regularize Modified Booth Multiplier Using Different Adders. In Machine Intelligence and Research Advancement (ICMIRA), 2013 International Conference on (pp. 387-391). IEEE

[28] Hwang K. [1979] Computer arithmetic principles, architecture, and design

[29] Pezaris SD. [1971] A 40-ns 17-bit by 17-bit array multiplier. IEEE Transactions on Computers, 100(4):442-447

[30] Robinson ME, Swartzlander E. [1998] A reduction scheme to optimize the Wallace multiplier. In Computer Design: VLSI in Computers and Processors, 1998. ICCD'98. Proceedings. International Conference on (pp. 122-127). IEEE

[31] Paradhasaradhi D, Prashanthi M, Vivek N. [2014] Modified wallace tree multiplier using efficient square root carry select adder. In Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on (pp. 1-5). IEEE.

[32] Khan S, Kakde S, Suryawanshi Y. [2013] Performance analysis of reduced complexity Wallace multiplier using energy efficient CMOS full adder. In Renewable Energy and Sustainable Energy (ICRESE), 2013 International Conference on (pp. 243-247). IEEE

[33] Asif S, Kong Y. [2015] Design of an algorithmic Wallace multiplier using high speed counters. In Computer Engineering & Systems (ICCES), 2015 Tenth International Conference on (pp. 133-138). IEEE

[34] Paradhasaradhi D, Prashanthi M, Vivek N. [2014] March). Modified wallace tree multiplier using efficient square root carry select adder. In Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on (pp. 1-5). IEEE

[35] Kesava RB. S Rao, BL Sindhuri, KB & Kumar NU. [2016] Low power and area efficient Wallace tree multiplier using carry select adder with binary to excess-1 converter. In Advances in Signal Processing (CASP), Conference on (pp. 248-253). IEEE

[36] Namba K, Ito H. [2005] Design of defect tolerant wallace multiplier. In Dependable Computing, 2005. Proceedings. 11th Pacific Rim International Symposium,pp. IEEE

[37] Jan M Rabaey. [1995] Digital Integrated Circuits, A Design Perspective, Prentice Hall.

[38] Fadavi-Ardekani J. [1993]M* N Booth encoded multiplier generator using optimized Wallace trees. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1(2):120-125

[39] Wallace CS. [1964] A suggestion for a fast multiplier. IEEE Transactions on electronic Computers, (1):14-17

[40] Liao MJ, Su CF, Chang CY, Wu AH. [2002] A carry-select-adder optimization technique for high-performance booth-encoded Wallace-Tree multipliers. In Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium IEEE, 1(1-1)

[41] Ismail RC, Hussin R. [2006] High performance complex number multiplier using booth-wallace algorithm. In Semiconductor Electronics, 2006. ICSE'06. IEEE International Conference on (pp. 786-790). IEEE

[42] Rao MJ, Dubey S. [2012] A high speed and area efficient Booth recoded Wallace tree multiplier for Fast Arithmetic Circuits. In Microelectronics and Electronics (PrimeAsia), 2012 Asia Pacific Conference on Postgraduate Research in (pp. 220-223). IEEE

[43] Kshirsagar RD, Aishwarya EV, Vishwanath AS, Jayakrishnan P. [2013] Implementation of pipelined booth encoded wallace tree multiplier architecture. In Green Computing, Communication and Conservation of Energy (ICGCE), 2013 International Conference on (pp. 199-204). IEEE

[44] Liao MJ, Su CF, Chang CY, Wu AH. [2002] A carry-select-adder optimization technique for high-performance booth-

encoded Wallace-Tree multipliers. In Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium,. 1: I-I). IEEE

[45] Fadavi-Ardekani J. [1993] M* N Booth encoded multiplier generator using optimized Wallace trees. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1(2):120-125

[46] Kshirsagar, R. D, Aishwarya, E. V, Vishwanath, A. S, & Jayakrishnan, P. (2013, December). Implementation of pipelined booth encoded wallace tree multiplier architecture. In Green Computing, Communication and Conservation of Energy (ICGCE), 2013 International Conference on (pp. 199-204). IEEE

[47] Qian, L, Wang, C, Liu, W, Lombardi, F, & Han, J. (2016, May). Design and evaluation of an approximate Wallace-Booth multiplier. In Circuits and Systems (ISCAS), 2016 IEEE International Symposium on (pp. 1974-1977). IEEE

[48] Rao MJ, Dubey S. [2012] A high speed and area efficient Booth recoded Wallace tree multiplier for Fast Arithmetic Circuits. In Microelectronics and Electronics (PrimeAsia), 2012 Asia Pacific Conference on Postgraduate Research in (pp. 220-223). IEEE

[49] Huddar SR, Rupanagudi SR, Kalpana M, Mohan S. [2013] Novel high speed vedic mathematics multiplier using compressors. In Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on (pp. 465-469). IEEE

[50] Kumar UP, Goud AS, Radhika A. [2013] FPGA Implementation of high speed 8-bit Vedic multiplier using barrel shifter. In Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference on(pp. 14-17). IEEE

[51] Anjana R, Abishna B, Harshitha MS, et al. [2014] Implementation of vedic multiplier using Kogge-stone adder. In Embedded Systems (ICES), 2014 International Conference on (pp. 28-31). IEEE

[52] Kogg, PM, Stone HS. [1973] A parallel algorithm for the efficient solution of a general class of recurrence equations. IEEE transactions on computers, 100(8): 786-793

[53] Sangani H, Modi TM, Bhaaskaran VK. [2014] Low power vedic multiplier using energy recovery logic. In Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on (pp. 640-644). IEEE

[54] Katreepalli R, Haniotakis T. [2017] Power-delay-area efficient design of vedic multiplier using adaptable manchester carry chain adder. In Communication and Signal Processing (ICCSP), 2017 International Conference on (pp. 1418-1422). IEEE

[55] Nikhil GV, Vaibhav BP, Naik VG, Premananda BS. [2017] Design of low power barrel shifter and vedic multiplier with kogge-stone adder using reversible logic gates. In Communication and Signal Processing (ICCSP), 2017 International Conference on (pp. 1690-1694). IEEE

[56] Kahar DK, Mehta H. [2017] High speed vedic multiplier used vedic mathematics. In Intelligent Computing and Control Systems (ICICCS), 2017 International Conference on (pp. 356-359). IEEE

[57] Thongbai N, Tuwanuti P. [2017] Optimization of multithread for long digit multiplier: By using ancient India Vedic mathematic. In Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2017 14th International Conference on (pp. 652-655). IEEE

[58] Gupta T, Sharma JB. [2017] A high-speed single-path delay feedback pipeline FFT processor using vedic-multiplier. In Information, Communication, Instrumentation and Control (ICICIC), 2017 International Conference on (pp. 1-6). IEEE

[59] Ram GC, Lakshmanna YR, Rani DS, Sindhuri KB. [2016] Area efficient modified vedic multiplier. In Circuit, Power and Computing Technologies (ICCPCT), 2016 International Conference on (pp. 1-5). IEEE

[60] Masurkar E, Dakhole P. [2016] Implementation of optimized vedic multiplier using CMOS technology. In Communication and Signal Processing (ICCSP), 2016 International Conference on (pp. 0840-0844). IEEE

[61] Gokhale GR, Gokhale SR. [2015] Design of area and delay efficient Vedic multiplier using Carry Select Adder. In Information Processing (ICIP), 2015 International Conference on (pp. 295-300). IEEE

[62] Morghade K, Dakhole P. [2016] Design of fast vedic multiplier with fault diagnostic capabilities. In Communication and Signal Processing (ICCSP), 2016 International Conference on (pp. 0416-0419). IEEE

[63] Pranav K, Pramod, P. [2015] Pipelined convolution using Vedic multiplier. In Intelligent Computational Systems (RAICS), 2015 IEEE Recent Advances in (pp. 33-38). IEEE.

[64] Choi J, Jeon J, Choi K. [2000] Power minimization of functional units partially guarded computation. In Proceedings of the 2000 international symposium on Low power electronics and design (pp. 131-136). ACM