International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 404-412 June 2018-Dec 2018 www.csjournals.com

Simulation &Comparative Analysis of BoothMultiplier

[1]Ekta Gupta, [2]Aniket Kumar*, [3]R.K. Jain
[1]Research Scholar,[1,3]Dept. of Physics, School of Basic & Applied Sciences,
[2]Dept. of Electronics & Electrical Engineering,
Shobhit Institute of Engineering & Technology, Meerut, U.P., India
(ANAAC Accredited Deemed to-be University)
*aniket.kumar@shobhituniversity.ac.in

Abstract: As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip. These signal processing applications not only demand great computation capacity but also consume considerable amount of energy. Multiplication is a fundamental operation in most signal processing applications. Multipliers consumes considerable power, have large area & longlatency. Therefore low-power multiplier design has been an important part in low- power VLSI system design. There has been extensive work on low-power multipliers at technology, physical, circuit and logic levels [1]. A system's performance is generally determined by the multiplier performance as it is generally the slowest element in the system. Furthermore, it is generally the most area consuming hence, optimizing the speed and area of the multiplier is a major design issue. The basic motive of our manuscript is to study and develop an Efficient Fast and Low Power Multiplier.
Keywords:VLSI, CSA, Efficient, Power Multiplier, Array, Vedic multipliers, LUTs, Fan Out, Delay.

Introduction

In many real-time DSP applications, high performance is a prime target. However, achieving this may be done at the expense of area, power dissipation and accuracy. Attempts have been made to use alternative number systems to optimize the realization of arithmetic blocks, maintaining high performance without incurring prohibitive area and power increases [1]. Fourier transforms play an important role in many digital signal processing applications including speech, signal and image processing. However, direct computation of Discrete Fourier Transform (DFT) requires on the order of N^2operations where N is the transform size. Parallel-pipelined FFTs are preferred for both high throughput and low power consumption.

Inthebinarynumbersystemthedigits,calledbits,arelimitedtotheset[0,1].Theresult of multiplying anybinary number by a single binary bit iseither 0, or the originalnumber. This makes forming the intermediatepartial-productssimple and efficient. For multiplier summingthesepartial-productsis time consumingtask. Oneofthe logicalapproachistoformthepartial-productsoneatatimeandsumthemastheyaregenerated.Forapplicationswherethisapproachdoes not provide sufficient performance, multipliers canbe implemented directly in hardware.

The seriesofbitshiftsandseriesofbitadditions is binary multiplication,in whichthe two numbers, the multiplicand and the multiplier are combined into the result. Consideringthebitrepresentationofthemultiplicand $x = x_{n-1} \ldots . x_1 \; x_0$ andthemultipliery=$y_{n-1} \ldots . y_1 y_0$inordertoformtheproductuptonshiftedcopiesofthemultiplicandareto be added.

Research is going on to design multipliers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier. But area and speed are two conflicting constraints. So improving speed results always in larger areas. In this manuscript we try to find out the best trade off solution among the both of them.

ADDERS

Addition is often used arithmetic operation on microprocessor, DSP, especially digital computers. Also, it serves as a building block for synthesis all other arithmetic operations. Therefore, regarding the efficient implementation of an arithmetic unit, the binary adder structures become a very critical hardware unit [2].

A. *Ripple Carry adder*

In Ripple Carry Adder cascaded full adders are used ,carry generated in previous full adder works as input carry for next stage full adder. Fig 1 shows that N bit Ripple Carry Adder require N full adders.

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 404-412 June 2018-Dec 2018 www.csjournals.com

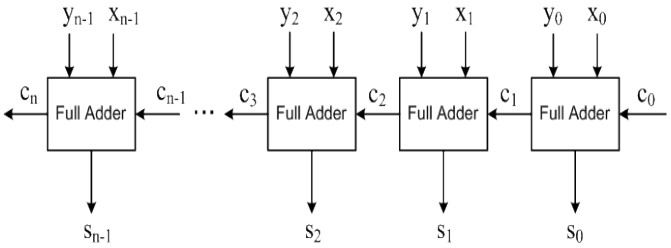


Figure 1 Block diagram of RCA

It can be represented in equation as shown below

$$C_i = x_i \& y_i \quad (1)$$

$$Pi = x_i \wedge y_i \quad (2)$$

$$Si = P_i \wedge Ci \quad (3)$$

B. *BITWISE Carry Select Adder*

The underlying strategy for Bitwise Carry Select Adder is that its uses two blocks of full adder or Ripple Carry Adder and generates two sets of sum bits and an outgoing carry. One set consider input carry as 1 and another set considers input carry as 0 and generates the corresponding sum and carry bit.

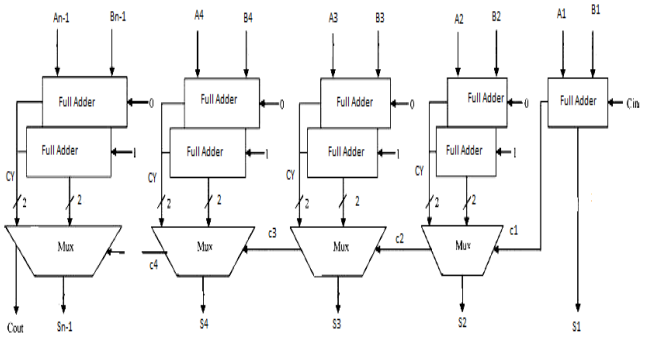
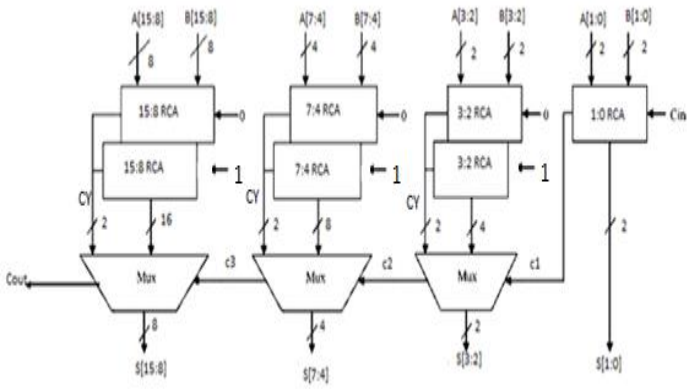


Figure 2Block diagram of Bitwise CSA

When the incoming carry into the group is assigned, its final value is selected out of the two sets. Hence in this we are reducing the time that next stage encounters while waiting for previous carry and then performing the addition operation. In Fig 2we have already added the two bits taking both input carry possibility and on the basis of previous carry we are only making the selection.

C. √ CARRY SELECT ADDER

The strategy used for Square Root Carry Select adder is



International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 404-412 June 2018-Dec 2018 www.csjournals.com

Figure3 Block diagram of Square Root CSA

Same as that of Bitwise Carry Select Adder(CSA) it also comprises of two blocks one with input carry as 1 and another block with input carry as 0 as shown in Fig. 3. But the difference lies in the size of blocks used in this. In this we areusing Ripple Carry Adder of more than 1 bit e.g. for 16-bit input we are using 2,2,4,8 bit Ripple Carry Adder.

D. Proposed Carry Select Adder

The strategy used for Modified Carry Select Adder is same as Square Root Carry Select Adder but instead of using block of Ripple Carry Adder with input carry as 1 in order to reduce the area and power consumption of the regular CSLA. Modified Carry Select Adder uses a block of Binary to Excess-1 converter. For n-bit ripple carry adder, n+1 bit of Binary to Excess-1 converter is used. This is done in order to reduce the area and power requirement of the previous Carry Select Adders.

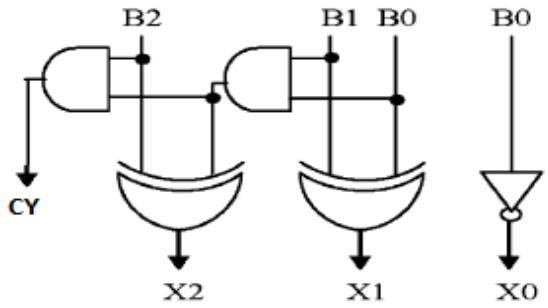


Figure4 Binary to Excess-1 logic diagram

It can be represented in equation as shown below

$$X_0 = \sim B_0 \quad (4)$$

$$X_1 = B_1 \wedge B_0 \quad (5)$$

$$X_2 = B_2 \wedge (B_1 \& B_0) \quad (6)$$

$$CY = B_2 \& (B_1 \& B_0) \quad (7)$$

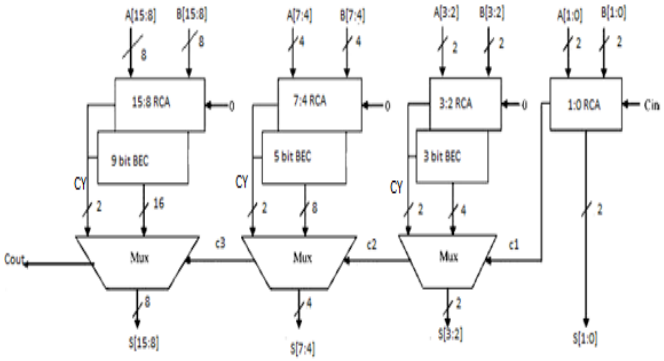


Figure 5 Block diagram of Proposed CSA

In order to replace the n-bit ripple carry adder (RCA), n+1-bit Binary to Excess-1 (BEC) is required. This produces the two possible partial results in parallel and the multiplexer is used to select either the BEC output or the direct inputs according to the control signal Cin.

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 404-412 June 2018-Dec 2018 www.csjournals.com

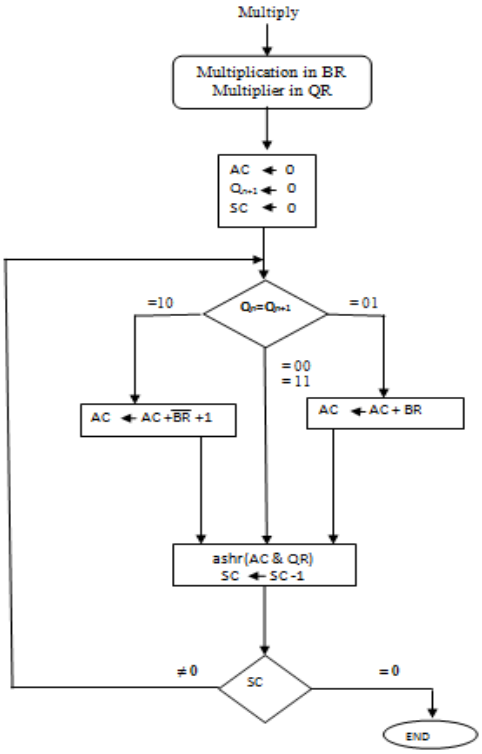


Figure 6 Complete flow chart of normal radix-2 booth multiplier

The LSB is added using conventional RCA, while other blocks are added in parallel along with the given incremented. Once all the interim sums and carries are calculated, the final sums are computed using multiplexers having minimal delay. The multiplexer block as shown in Fig 2.5, receives the two sets of input and selects the final sum based on the select input from the previous stage [2].

The significance of the binary to excess-1 logic stems from the large silicon area reduction when the CSLA with large number of bits are designed. The use of BEC logic with multiplexer, achieves fast incrementing action with reduced device count. Thus, the proposed CSA excels the conventional CSA circuit in terms of area and power.

Multiplication is normally done in two steps- Partial product generation and addition. In booth multiplication, partial product generation is done based on radix 2 encoding. Bits of multiplicand (Y) are grouped from left to right and corresponding operation onmultiplier (X) is done to generate partial product. The addition of partial products is carried out by using CSA.

Parallel Multiplication using basic Booth's recoding algorithm technique based on the fact that partial product can be generated for group of consecutive 0's and 1's which is called Booth's recoding. This recoding is used to generate efficient partial product, which always have large number of bits than

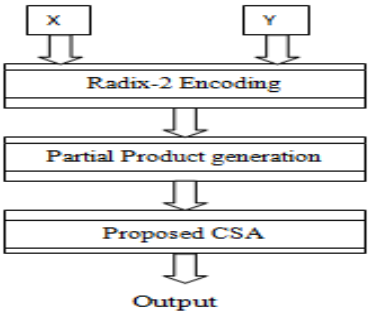


Figure 7 Complete flow chart of normal radix-2 booth multiplier

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 404-412 June 2018-Dec 2018 www.csjournals.com

A. *BOOTH Multiplier using* $\sqrt[3]{CSA}$

This multiplier is same as that using Bitwise CSA. It uses same logic as that used in booth multiplier using bitwise CSA, instead of using bitwise CSA we are using here Square root CSA. In this type of adder we are using blocks of ripple carry

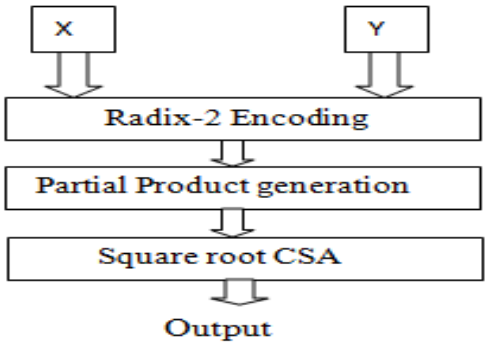


Figure8 Flow chart of normal radix-2 booth multiplier using square root CSA

adder with input more than one bits instead of single bit as in case of bitwise CSA. In this type of multiplier we are using Square root CSA in order to reduce the carry propagation delay of Normal booth multiplier. Partial product so generated is added using Square root CSA as shown in fig.8.

B. *Multiplier using Proposed CSA*

In this type of multiplier we are using proposed CSA in order to reduce the carry propagation delay of Normal booth multiplier. In this the basic concept used is same that of multiplier using square root CSA the only difference is that instead of using ripple carry adder block with the input carry as 1 we are using BEC. Partial product so generated is added using proposed root CSA

SIMULATION RESULTS

The VHDL code for all the adder & multiplier circuits are simulated on **Xilinx 14.4** & then implemented on Xilinx 14.4 with family Spartan6, device as XC6SLX45, package CSG324 with speed grade of -3 for comparative analysis.Fig. 10 Simulation waveform of normal booth multiplier.

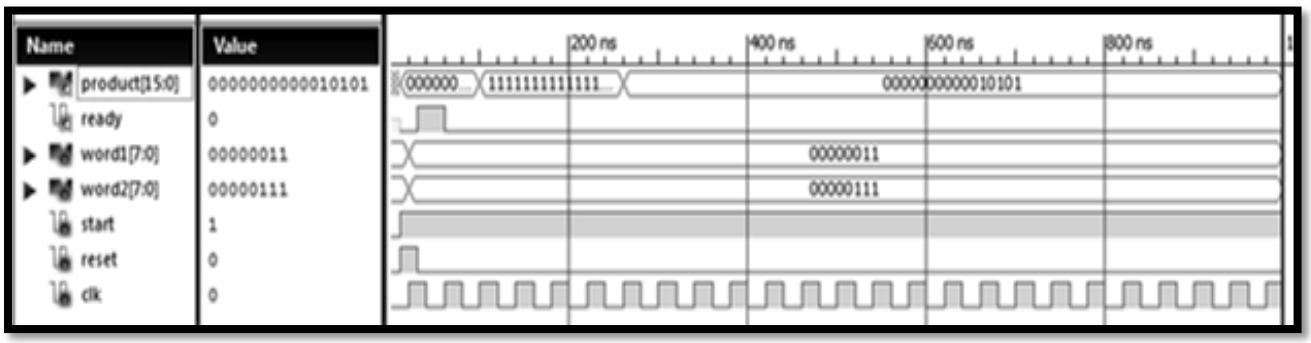


Fig. 9 Simulation waveform of normal booth multiplier

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 404-412 June 2018-Dec 2018 www.csjournals.com

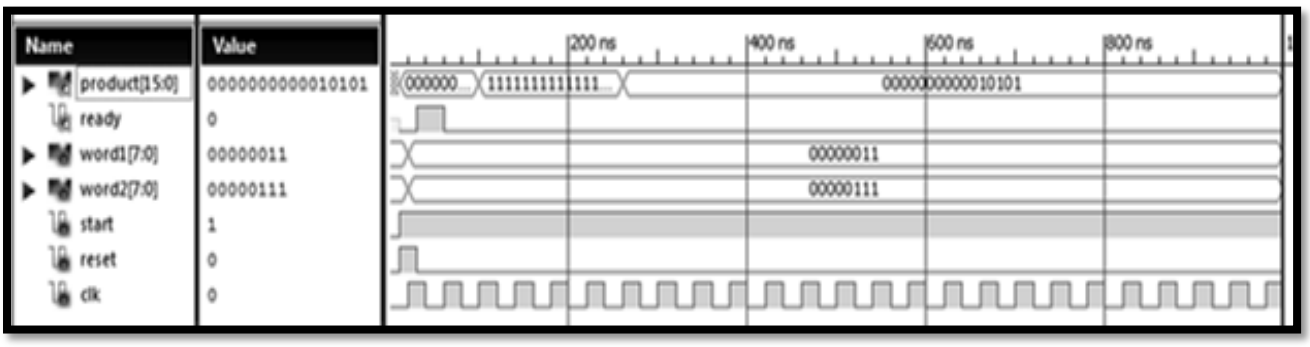


Figure 10Simulation waveform of booth multiplier using bitwise CSA

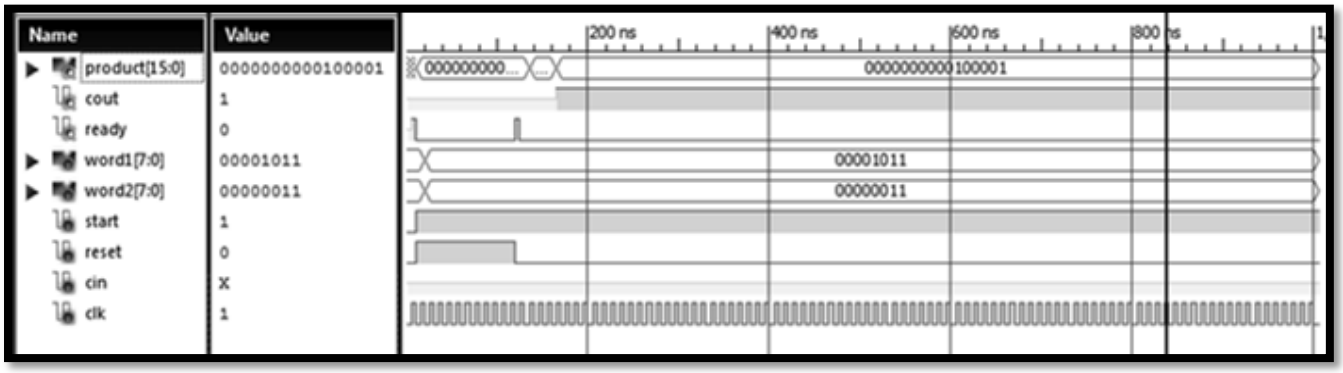


Figure 11 Simulation waveform of booth multiplier using square root CSA

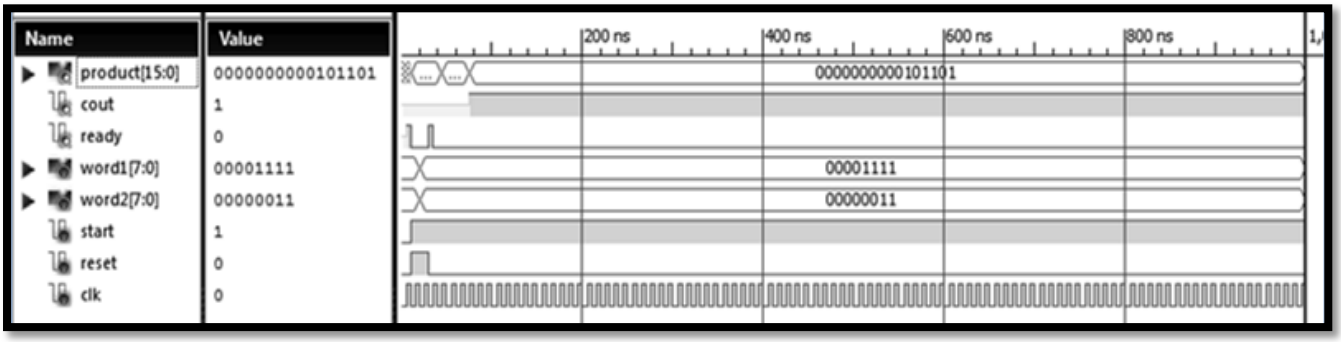


Figure 12 Simulation waveform of booth multiplier using proposed CSA

COMPARARTIVE ANALYSIS

The performance analysis of implementation& simulation is show in tabular & graphical form respectively.

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 404-412 June 2018-Dec 2018 www.csjournals.com

TABLE 1. Comparison of Adders

Parameters		Simple ripple Carry Adder	Bitwise CSA	Sqrt. CSA	Proposed BEC adder
Path Delay(ns)		8.303	15.456	12.891	10.125
On chip power utilization(mW)		81.18	81.18	81.18	80.98
Memory (MB)		147	145	92	92
Device Utilization	No. of Slice Flip Flops	0%	1%	1%	1%
	No. of occupied Slices	0%	1%	1%	1%
	Total Number of 4-input LUTs	1%	1%	1%	1%

TABLE 2. Comparison of Booth Multipliers

Parameters		Simple multiplier	Bitwise CSA multiplier	Sqrt . CSA multiplier	Proposed BEC multiplier
Path Delay (ns)		7.055	7.0	6.89	6.56
Memory (KB)		186360	103584	100032	79772
On chip power utilization (nW)		80.98	80.1	75.37	69.45
Number of bonded. IOBsS (utilization)		15%	15%	15%	20%
Device Utilization	No. of Slice Flip Flops	0%	1%	1%	1%
	No. of 4 input LUTs	1%	1%	1%	1%
	No. of bonded. IOBsS	15%	15%	15%	20%

Fig. 9 – 12 shows simulation waveform of multiplier viz. normal booth multiplier, booth multiplier usingbitwise CSA , booth multiplier using square -root CSA&booth multiplier using proposed CSA.
Table no. 1 shows comparative analysis of different adders. From this table it can said that delay in proposed BEC adder is more than normal adder but power consumption & memory usage is comparatively lower.
Table no. 2 shows comparative analysis of different multiplier viz. normal booth multiplier, booth multiplier using bitwise CSA, booth multiplier using square -root CSA & booth multiplier using proposed CSA. Analysis

says that path delay in proposed multiplier is lowest then reaming types booth multiplier, therefore it can be concluded that response time of multiplier has increased . On chip power utilization of proposed multiplier is lowest among all.

CONCLUSION

In this manuscript different adders & multipliers has been simulated & then implemented . Experimentally it was found that adder using binary to excess-1 logic when used in booth multiplier , performance of multiplier increases in terms of speed , power consumption & memory usage .

FUTURE WORK

By utilizing the findings of this research work , similar analysis can be done for Vedic & Array multiplier for different bit lengths using different types adder & purposed adder . Then after comparative analysis may be preformed for Booth , Array &vedic multiplier after implementation on Xilinx 14.4 with family Spartan6, device as XC6SLX45, package CSG324 with speed grade of -3.

References

[1] Jasbir Kaur, Sumit K, "Analysis and Comparison of Different Multiplier,"International Journal on Recent and Innovation Trends in Computing andCommunication ISSN: 2321-816975-78. Volume: 4 Issue: 6. June 2016.

[2] Kavitha Priya N, and Karthikeyan KV, "Analysis of Mac Unit Using VedicMultiplier and Sklansky Adder", Research Journal of Pharmaceutical, Biologicaland Chemical Sciences, ISSN: 0975-8585, May – June 2016.

[3] Vishal Galphat1, Nitin Lonbale, "The High Speed Multiplier by using PrefixAdder with MUX and Vedic Multiplication," International Journal of Science andResearch (IJSR), ISSN (Online): 2319-7064, Volume 5 Issue 1, January 2016.

[4] Mahadevappa Mugalihal, Dr. T. C. Thanuja, "Design and Implementation of anEfficient 32×32 Bit Vedic Multiplier Using Kogge Stone Adder," InternationalJournal of Advanced Research in Computer Science and Software Engineering,Volume 6, Issue 6, June 2016 ISSN: 2277 128X.

[5] Sudeep.M.C, "Design and FPGA Implementation of High Speed VedicMultiplier," International Journal of Computer Applications (0975 – 8887),Volume 90 – No 16, March 2014.

[6] Savita Nair & Ajit Saraf, "A Review Paper on Comparison of Multipliers based on Performance Parameters," International Conference on Advances in Science and Technology (ICAST-2014).

[7] Saji. M. Antony, "Design of High Speed Vedic Multiplier using Multiplexer basedAdder," International Conference on Control, Communication & Computing India(ICCC) 119-21 November 2015.

[8] K.Harika, "Analysis of Different Multiplication Algorithms & FPGAImplementation," IOSR Journal of VLSI and Signal Processing (IOSR-JVSP),Volume 4, Issue 2, Ver. I (Mar-Apr. 2014), PP 29-35, e-ISSN: 2319 – 4200, pISSN No. : 2319 – 4197.

[9] Parul Agrawal, Rahul Sinha, "Comparative Analysis and FPGA Implementationof Vedic Multiplier for various Bit Lengths using Different Adders," IJIRCCE(An ISO 3297: 2007 Certified Organization), Vol. 3, Issue 10, October 2015,ISSN(Online): 2320-9801, ISSN (Print): 2320-9798.

[10] Mahendra Tiwari & Aniket Kumar**,** Implementation of High Speed and Low Power Novel Radix 2 BoothMultiplier using 2248 BECConverter , International Journal of Engineering Science and Computing, Volume 7,Issue No. 3, pp. 4861-4863, March 2017.

[11] Bhavesh Sharma , Amit Bakshi, "Comparison of 24X24 Bit Multipliers forVarious Performance Parameters," 1st International Conference on Advent Trendsin Engineering, Science and Technology "ICATEST 2015", 08 March 2015.

[12] B. Ramkumar, H.M. Kittur, and P. M. Kannan, "ASIC implementation of modified faster carry save adder," *Eur*.J. Sci. Res.vol. 42, no. 1, pp. 53–58, 2010.

[13] K.S.Rao, M.Reddy, A.V.Babu and P.Srinivasulu, "Energy and area efficient carry select adder," *IJERA, vol. 34* issue 2, pp.436-440, Mar-Apr 2012.

[14] Y.Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," *Electron. Lett., vol. 37, no. 10, pp. 614–615*,May 2001.

[15] Z.Huang, "High-Level Optimization Techniques for Low-Power Multiplier Design," *PhD dissertation, Univ. of*California Los Angeles, June 2003

[16] M. Morris mano: Computer system architechture Prentice Hall Date: 1992-10-29,pp 344.

[17] Jagadguru Swami Sri Bharti Krishna Tirthaji Maharaja "Vedic Mathematics or Sixteen Simple Mathematicle Formulae from the Veda, Delhi(1965)", Motilal Banarsidas Varanasi, India.

[18] Ch. Harish Kumar, Implementation and Analysis of Power, Area and Delay of Array, UrdhvaandNikhilam Vedic Multipliers, International Journal of Scientific and Research Publications, 2013, 3 (1), 1-5.

[19] PushpalataVerma and KK Mehta, Implementation of an Efficient Multiplier based on Vedic Mathematics Using EDA Tool, International Journal of Engineering and Advanced Technology, 2012,1(5), 75-79

[20] G Kumar, Design of High Speed Vedic Multiplier using Vedic Mathematics Techniques, International Journal of Scientific and Research Publications, 2012, 2 (3), 1-5.

[21] PrabirSaha, Arindham Banerjee, ParthaBattacharyya andAnupDhandapat, High Speed Design of Complex Multiplier using Vedic Mathematics, Proceedings of IEEE Students Technology Symposium, IIT Kharagpur, India, 2011, 237-241.

[22] Aniket Kumar and Vishikha, "Comparative Analysis of Vedic & Array Multiplier", International Journal ofElectronics andCommunication Engineering and Technology, Vol. 8, Issue. 3, pp.17-27, May - June 2017.

[23] Shripad Kulkarni, Discrete Fourier Transform (DFT) by using Vedic Mathematics, Papers on Implementation of DSP Algorithms/VLSI Structures using Vedic Mathematics, www.edaindia.com, IC Design Portal, 2006.

[24] Multiplier Architectures, International Journal of Emerging Technology and Advanced Engineering,2012, 2(6), 292-295.

[25] Sunjoo Hong, TaehwanRoh and Hoi-Jun Yoo, A145w 8×8 Parallel Multiplier Based on Optimized ByPassing Architecture, Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Republic of Korea, Circuits and Systems (ISCAS), 2011 IEEE International Symposium on, 2011, 1175-1178.

[26] Vadivel Sandeep Shrivastava, Jaikaran Singh and Mukesh Tiwari, Implementation of Radix-2 Booth Multiplier and Comparison with Radix-4 Encoder Booth Multiplier, International Journal on Emerging Technologies,2011, 2(1), 14- 16.

[27] Wen-Chang Yeh and Chein-Wei Jen, High-Speed Booth Encoded Parallel Multiplier Design, IEEE Transaction on Computers, 2000, 49, 692-701.

[28] Ravindra P Rajput, MNShanmukhaSwamy, High Speed Modified Booth Encoder Multiplier for Signed and Unsigned Numbers, IEEE International Conference on Modelling and Simulation, 2012.

[29] Jagadguru Swami Sri Bharti Krishna Tirthaji Maharaja,Vedic Mathematics or Sixteen Simple Mathematicle Formulae from the Veda, MotilalBanarsidas Varanasi, India, 1965.

[30] Sumit Vaidya and Deepak Dandekar,Delay-Power Performance Comparison of Multipliers in VLSI Circuit Design,International Journal of Computer Networks and Communications, 2010, 2 (4), 47-56.

[31] AbhijitAsati and Chandrashekhar, A High-Speed, Hierarchical 16×16 Array of Array Multiplier Design,International Conference on Multimedia, Signal processing andCommunication Technologies, 2009,161-164.

Shamim Akhter, VHDL Implementation of Fast N×N Multiplier based on Vedic Mathematic, 18th European Conference on Circuit Theory and Design, Spain, 2007, 472-475.