ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

Design and Development of Approximate Multiplier Using Urdhva Algorithm

N.Mageshwari[1], G.A.Sathish Kumar[2]

PG Scholar, Department of ECE, SVCE, Sriperumbudur, India

Professor, Department of ECE, SVCE, Sriperumbudur, India

ABSTRACT*:* In VLSI design, the performance of any system is mainly based on the performance of the multiplier. Multiplier is the one of the slow element in the system. The speed of the multiplier depends on the multiplication technique and type of adder used. This paper proposes the architecture of 128x128 multiplier using Vedic Mathematics. In Vedic mathematics several sutras are present. In this paper 'UrdhvaTriyakbhyam' sutra is used because the increase in delay and power consumption with the increase in number of input bits is at a slower rate when compared with the other algorithms. .The partial product addition in Vedic multiplier is realized using Ripple Carry Adder. This paper presents the detailed study of different multipliers like Booth Multiplier, UrdhvaTriyakbhyam multiplier based on vedic mathematics. Vedic multiplier is coded in Verilog HDL and simulated in Xilinx ISE Simulator 14.1 and synthesized using Cadence EDA tool. The multipliers are then compared based on path delays. Results show that UrdhvaTiryakbhyam multiplier is the fastest Multiplier with least path delay.

KEYWORDS-Vedic Mathematics, Delay, VLSI, Ripple Carry adder, UrdhvaTiryakbhyam Sutra, Verilog HDL

I. INTRODUCTION

A multiplier is one of the major blocks in most of the applications such as encryption and decryption in cryptography, digital signal processing and in other logical computations. With emerging technology, many researchers tried to design multipliers which offer either high speed, low power consumption, or less area. Multiplier is the core component of any Digital Signal Processing (DSP) applications and hence speed of the processor depends mostly on the multiplier design. Since multiplication dominates the execution time of most DSP algorithms, there is a necessity of designing high speed multiplier. Currently, multiplication time is the main factor in shaping the instruction cycle time of a DSP chip. The need of the fast multiplication had given rise to a multiplier called Booth multiplier using modified Booth algorithm. Although Modified Booth Algorithm (MBA) is the most successful method yet it is more time consuming. The conventional mathematical algorithms can be simplified and optimized by the use of Vedic mathematics. By using this Vedic algorithm the computational speed of the processor can be improved to perform fast arithmetic operations.Vedic multiplication is built on Vedic mathematics which is further extracted from the ancient vedas by Sri Bharathi Krishna Trithaji in between 1911-1918. The speciality of vedic mathematics is that it gives simple way to solve the calculations which can be easily understood by the human minds. This vedic mathematics is divided into 16 sutras which gives different rules for the simplication of the problems related to the trigonometry, algebra, geometry etc.

A. Vertically Crosswise Technique

The Sanskrit name of vertically crosswise technique is 'UrdhvaTriyakbhyam' which is a universal algorithm for multiplication. This method multiplies the digits vertically and crosswise and finally adds them using appropriate adder. This rule is applicable to both binary numbers and integers. The best feature of this method is that the partial products needed for the multiplication are already generated in advance and this leads to decrease in delay and power consumption. The increase in number of bits however increases area and critical delay [3].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

1) For Decimal number

There will be 2N-1 steps for a n digit decimal number. The result is stored after the digits on the line are multiplied. Then the previous carry is added to it. The least significant digit of resultant coming from addition is kept as sum bit of the multiplication and all other digits are considered as previous carries for the next step. When the number of lines becomesmore than one, then all the outcomes are accumulated with the previous carries. Initial carry in this case is taken as zero[3].

2) For Binary number

The same way is extended to the binary multiplication. One bit multiplication is a simple AND gate operation ofthe operands. For the higher bit multiplication, the following method is used for the evaluation of the equations for the resultant bits of multiplication. This method of multiplication is known as square table method. In this technique each small square block inside the table is partitioned into two identical parts. The multiplier and multiplicand values are written on the successive sides of the table as shown in Figure.1.

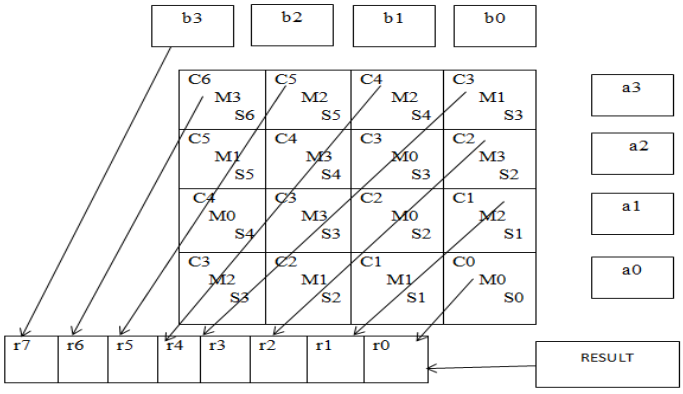

Figure.1: 4x4 Vedic Multiplication

B. Ripple Carry Adder

A Ripple Carry Adder is a digital circuit that is used to produce the arithmetic sum of two binary numbers. It can be constructed with full adders connected in series, with the carry output from each full adder stage can be connected as carry input of the next full adder in the chain. Bits A0 and B0 in the Figure.2 represent the least significant bits of the numbers to be added. The sum output is represented by the bits S0-S3 and the carry output is represented as C4.

1) Ripple Carry Adder Delays

In the ripple carry adder, the output is known after the carry generated from the previous stage. Thus, the sum of the most significant bit is only available after the carry signal has rippled through the adder stages from the least significant stage to the most significant stage. As a result, the final sum and the carry bits will be valid after a considerable delay.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

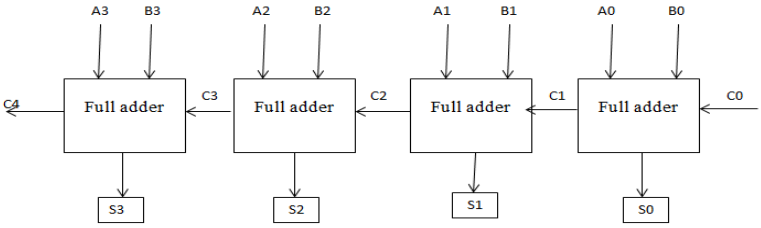

Figure.2: 4-bit full adder

II. BOOTH MULTIPLIER

Booth multiplier is a multiplier which can be used to multiply two binary integers with reduced delay. Even though modified Booth Algorithm is the most successful method it is more time consuming because the partial product generation and the addition of the partial product can takes place in two different stages. Hence the total path delay of the Booth multiplier is increased. In order toreduce the path delayUrdhva algorithm of Vedic mathematics is used in which the partial product generation and the addition of the partial products can be done in a single stage. Booth multiplication for a 5bit multiplier and multiplicand is explained in Table.1.

Table.1: Booth Algorithm for 5bit multiplier and multiplicand

Step	Multiplicand	Action	Multiplier upper 5-bits 0, lower 5-bits multiplier, 1 "Booth bit" initially 0
0	01110	Initialization	00000 11011 0
1	01110	10: Subtract Multiplicand	00000+10010=10010 10010 11011 0
		Shift Right Arithmetic	11001 01101 1
2	01110	11: No-op	11001 01101 1
		Shift Right Arithmetic	11100 10110 1
3	01110	01: Add Multiplicand	11100+01110=01010 (Carry ignored because adding a positive and negative number cannot overflow.) 01010 10110 1
		Shift Right Arithmetic	00101 01011 0
4	01110	10: Subtract Multiplicand	00101+10010=10111 10111 01011 0
		Shift Right Arithmetic	11011 10101 1
5	01110	11: No-op	11011 10101 1
		Shift Right Arithmetic	11101 11010 1

A. Booth multiplier

Booth multiplication algorithm gives a procedure for multiplying binary integers in signed -2's complement representation. Following steps used for implementing 5 bit booth algorithm is shown in Table.1.

Step1: Consider 5bit multiplier = 11011 and 5 bit multiplicand = 01110. Initial value in the accumulator register is 00000 and booth bit is 0.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

Step2: Check the LSB bit of the multiplier and the booth bit. Here it is 10 hence subtract multiplicand from the accumulator value and right shift the result obtained with '0' fill or '1' fill depending upon the MSB bit of the accumulator.

Step3: Now the LSB bit of the multiplier and the booth bit is 11. Therefore right shift the result with no operation.

Step 4: Now the condition is 10 hence subtract the multiplicand value from the accumulator value and right shift the result with '1' fill.

III. VEDIC MULTIPLER

The Vedic multiplier is based on the Vedic multiplication formulae (Sutras). These Sutras are used for the multiplication oftwo decimal numbers. Here, we apply the similar procedure to the binary number system to build the proposed algorithm compatible with the digital hardware. Vedic Sutras are applied to almost every branch of Mathematics. They even apply to complex problems relating to a large number of mathematical operations. The Sutras are helpful in saving a lot of time and reducethe effort in solving the problems though the solutions appear like supernatural, but it is perfectly logical and rational. The Sutras provide not only the methods of estimate, but also the ways of thinking their application.

A. UrdhvaTriyakbhyam Sutra

UrdhvaTriyakbhyam sutra is also called as vertically crosswise technique. The increase in area and delay with the increase in number of input bits is at slower rate when compared with other Vedic algorithms. Previously UT algorithm is used only for the multiplication of decimal numbers. Now UT algorithm can also be used for the multiplication of two binary numbers. Figure.3 explains the 4 bit binary Vedic multiplication.

1) 16-bit Vedic multiplier

Implementation of 16 bit Vedic multiplier requires 16 bit multiplier a15a14……a0 and 16-bit multiplicand b15b14……..b0. 16-bit multiplier requires four 8-bit Vedic multiplier and three 16-bit Ripple Carry Adders. The sum bit obtained for 16-bit Vedic multiplier is [31:0]s and one carry bit namely c3. The block diagram for 16-bit Vedic multiplier is shown in Figure.4

Step1: The 16-bit multiplicand and the multiplier are divided into two 8bit operands.
Step2: The LSB bits of the multiplier and the multiplicand are multiplied first and the process is repeated to N-1 steps of multiplication for a N-bit multiplier.
Step3: 16-bit multiplier has 32 sum bits and 1 carry bit.

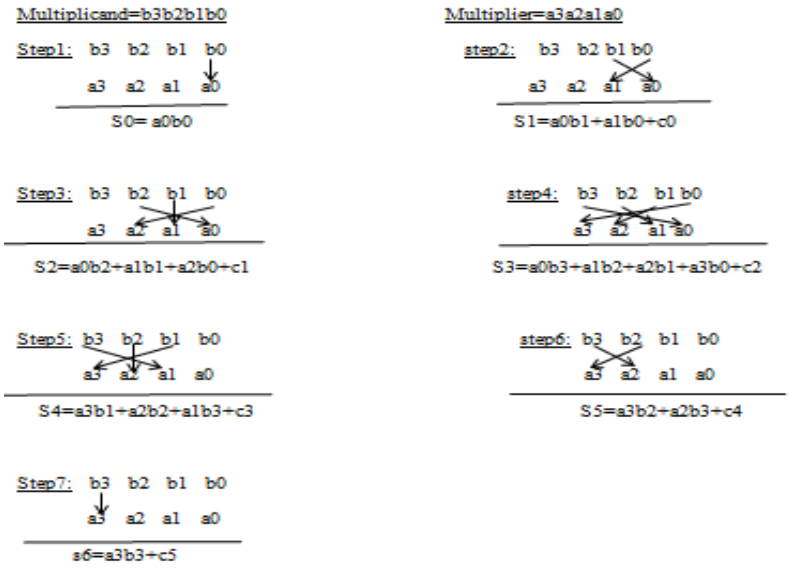

Figure.3: 4 bit binary Vedic multiplication

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

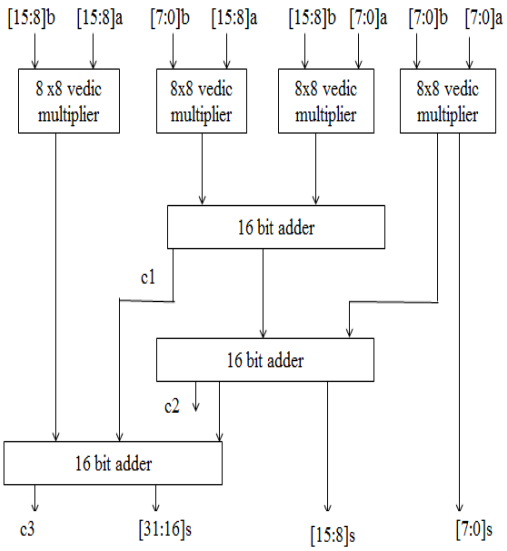

Figure.4: Block diagram for 16 bit Vedic multiplier

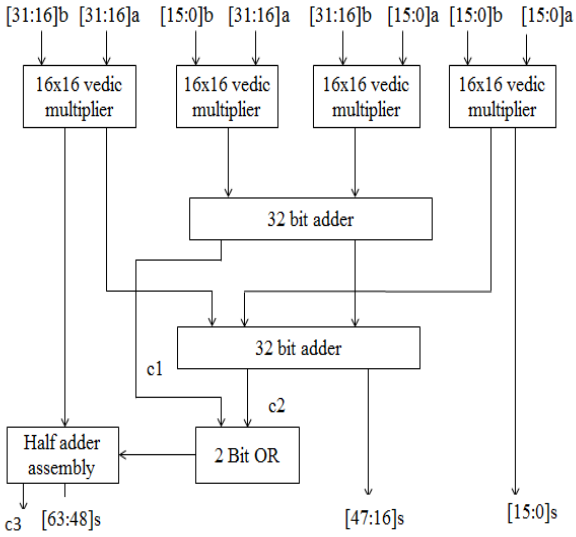

Figure.5: Block diagram for 32 bit Vedic multiplier

2) 32-bit Vedic multiplier

The block diagram for 32-bit Vedic multiplier is shown in Figure.5. The 32-bit Vedic multiplier requires four 16-bit multipliers and three 32-bit ripple carry adders. Ripple carry adders are used here for their high speed performance and accuracy. The multiplicand required is 32-bit and the multiplier required is also 32-bit.The sum bit obtained for 32-bit multiplier is [63:0]s and single carry bit obtained called c3. It also requires an assembly of half adders in order to produce the MSB bits of sum.

3) 64-bit Vedic multiplier

The 64-bit Vedic multiplier requires four 32 bit multiplier, three 64-bit Ripple Carry Adders, one OR gate and an half adder assembly. If the half adder assembly is replaced by a 16-bit ripple carry adder then the speed of operation of the 64 bit multiplier can be increased and the circuit complexity can also be reduced.Since the ripple carry adder utilizes the full adders the accuracy can also be increased in the multiplication process. Ripple carry adders are used here for their high speed performance and accuracy. The multiplicand required is 64-bit and the multiplier required is also 64-bit.The sum bit obtained for 64-bit multiplier is [127:0]s and single carry bit obtained called c3. It also requires an assembly of half adders in order to produce the MSB bits of sum. The block diagram for 64-bit Vedic multiplier is shown in Figure.6.

4) 128-bit Vedic multiplier

The 128-bit Vedic multiplier requires 128-bit multiplier [127:0]a and 128 bit multiplicand [127:0]b. Three 128 bit Ripple Carry Adders and a 64-bit Ripple Carry Adder is used. The block diagram for 128-bit Vedic multiplier is shown in Figure.7.The combinational path delay obtained for 128-bit Vedic multiplier is about 246.377 ns. The combinational path delay obtained for existing 16-bit Booth multiplier is 77.08ns and for existing 32-bit Booth multiplier the combinational path delay obtained is 149.940 ns whereas for the proposed 32 bit Vedic multiplier the combinational path delay obtained is only about 73.426ns. For 64 bit Vedic multiplier the combinational path delay is 136.945 ns. Thus the speed of the Vedic multiplier gets increased thereby the power consumption can be reduced.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

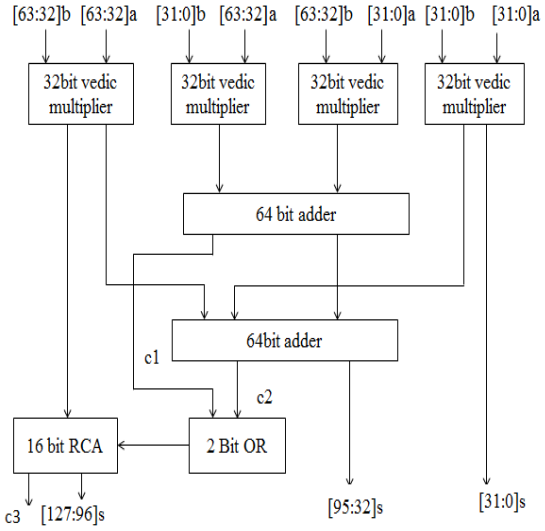

Figure.6: Bock diagram for 64 bit Vedic multiplier

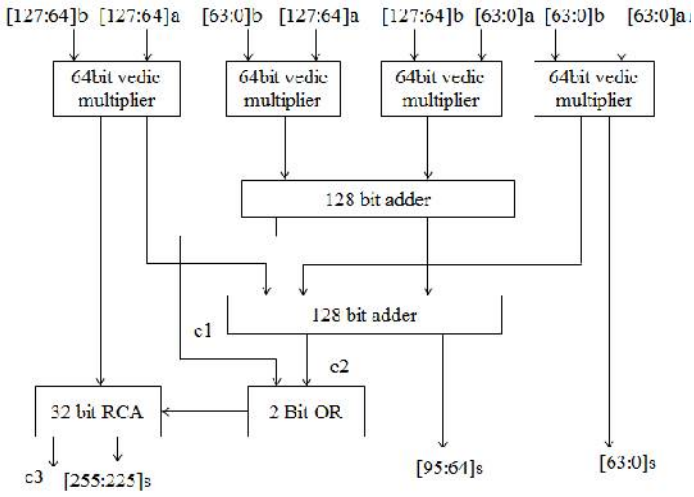

Figure.7: Block diagram for 128 bit Vedic multiplier

IV. RESULTS AND DISCUSSION

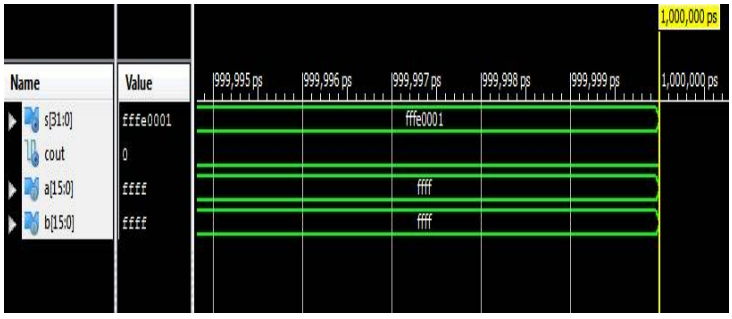

Figure.8: Simulation result for 16 bit Vedic multiplier

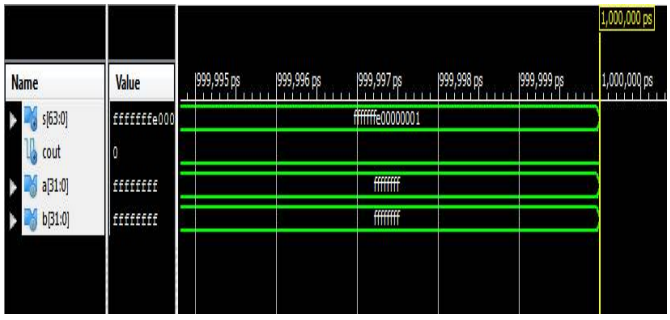

Figure.9: Simulation result for 32 bit Vedic multiplier

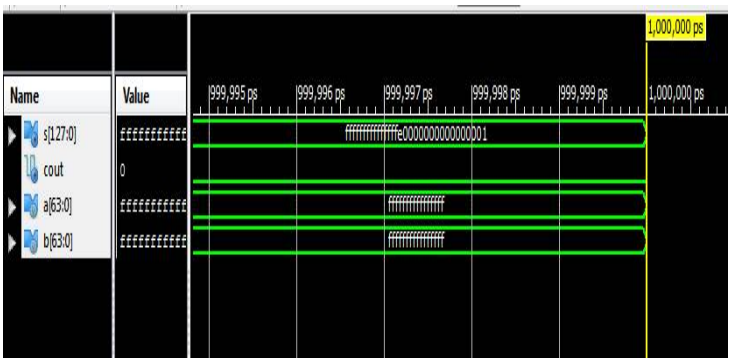

Figure.10: Simulation result for 64 bit Vedic multiplier

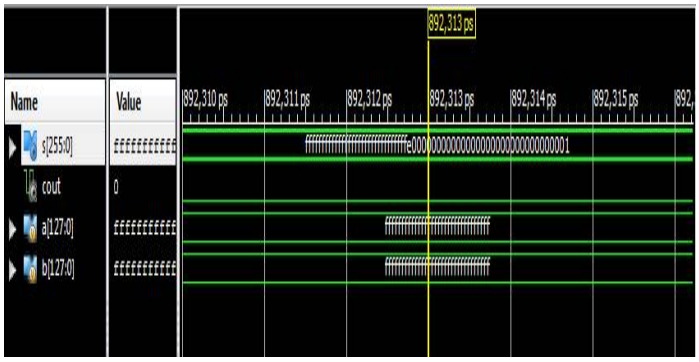

Figure.11: Simulation result for 128 bit Vedic multiplier

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

V. PERFORMANCE COMPARISON

A. *Maximum Combinational path delay obtained using Xilinx ISE simulator*

The maximum combinational path delay is obtained by the simulation of Vedic multiplier using Xilinx ISE simulator 14.1. Delay comparison of the Vedic multiplier with the existing booth multiplier is shown in the Table.2. On comparing the two multipliers the delay obtained for the Vedic multiplier is smaller than the delay obtained for the booth multiplier using Modified Booth Algorithm.The combinational path delay obtained for the proposed 128 bit Vedic multiplier is 246.377 ns.

Table.2: Delay comparison of Vedic and Booth multiplier

BOOTH MULTIPLIER	VEDIC MULTIPLIER
8 bit: 39.754ns	8bit: 24.04 ns
16 bit: 77.802ns	16bit: 41.228 ns
32 bit: 149.940 ns	32bit: 73.426 ns

B. *Delay summarization using cadence EDA tool*

The maximum combinational path delay obtained for the Vedic multiplier using Cadence EDA tool is shown in Table. 3. The proposed Vedic multiplier is coded using Verilog HDL and simulated using Xilinx ISE simulator version 14.1. Synthesis has been done using Cadence EDA tool.

Table.3: Delay summarization using Cadence EDA Tool

VEDIC MULTIPLIER	DELAY OBTAINED
16 bit	11400ps
32bit	20772.70ps
64bit	41268.20ps
128bit	81417.40ps

VI. CONCLUSION

A technique for the multiplication of 128 bit operands with the help of Vedic multiplier is described. The proposed architecture is based on 'UrdhvaTriyakbhyam' sutra of Vedic mathematics which is a general multiplication technique for multiplication. This algorithm makes the parallel generation of partial products and removes unwanted multiplication steps. Also with this algorithm the increase in delay and power consumption is less with the increase in number of bits. This proposed technique works in two steps. In the first step the equation for each bit of the resultant is computed. In the second step that equation is executed with the required multipliers and Ripple Carry Adders. A good result in terms of speed is seen in this work with the approximate combinational path delay of about 246.377 ns for

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

128 bit Vedic multiplier. This optimized Vedic multiplier design may prove to be of greater use in future Digital Signal Processing applications with stringent demands of speed and power consumption.

VII.FUTURE SCOPE

Here design and development of 128 bit Vedic multiplier using UrdhvaTriyakbhyam algorithm has been proposed. The synthesis and simulation results for n bit multipliers has shown. The comparison result of Vedic multiplier with Booth multiplier showed that Vedic multiplier is more efficient in terms of delay. The performance of the appropriate multiplier can further be improved by using various adders (Ripple Carry Adder, Carry Save Adder) using reversible logic gates.

REFERENCES

[1] Chilton Fernandes, Samarth Borkar, "Application of Vedic Mathematics In Computer Architecture", International Journal of Research in Engineering and Science (IJRES), Volume 1 Issue 5, 2013, PP.40-453.
[2] Ganesh Kumar G, V.Charishma, "Design of High Speed Vedic Multiplier using Vedic Mathematics Techniques", International Journal Of Computer Applications, Volume 73-No.10, 2013.
[3] Yogita Bansal, Charu Madhu, "A novel high-speed approach for 16x16 Vedic multiplication with Compressor adders", Computers and Electrical Engineering 49(2016)39-49.
[4] Jagadguru Swami Sri Bharath, Krsna Tirathji, "Vedic Mathematics or Sixteen Simple Sutras From The Vedas", Motilal Banarsidas, Varanasi(India),1986.
[5]Krishnaveni.D,Umarani T.G., "VLSI Implementation of Vedic Multiplier with Reduced Delay", International Journal Of Advanced Technology & Engineering Research(IJATER,ISSN No. 2250-3536 Volume 2, Issue 4, 2012).
[6] N.G.Nirmal, Dr. D.T.Ingole, "Novel Delay Efficient Approach for Vedic Multiplier with Generic Adder Module", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 3, Issue 3, 2013, PP. 1394-1396.
[7] R.Priya and J.Senthil Kumar , "Implementation and comparison of vedic multiplier using Area Efficient CSLA Architectures", International Journal Of Computer Applications,Volume 73-No.10,2013.
[8] Pushpalatha Verma, K. K. Mehta "Implementation of an Efficient Multiplier based on Vedic Mathematics Using EDA Tool" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-1, Issue-5, 2012.
[9] Mrs. M. Ramalatha, Prof. D. Sridharan, "VLSI Based High Speed Karatsuba Multiplier for Cryptographic Applications Using Vedic Mathematics", IJSCI, 2007.
[10] Sandesh S. Saokar, R. M. Banakar. SarojaSiddamal, "High Speed Signed Multiplier for Digital Signal Processing Applications", IEEE 2012.
[11] Thapliyal H. and Srinivas M.B. "High Speed Efficient N x N Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics", Transactions on Engineering, Computing and Technology, 2004, Vol.2.
[12] J.-P. Wang, S.-R. Kuang, and S.-C. Liang, "High-accuracy fixed-width modified booth multipliers for lossy applications", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 1, pp. 52–60, 2011.
[13] Wallace, C.S., "A suggestion for a fast multiplier," IEEE Trans. Elec. Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.
[14] Anju Sunny, Binu K. Mathew., Dhanusha P.B, " Area Efficient High Speed Appropriate Multiplier with Carry Predictor", International Conference on Emerging Trends in Engineering, Science and Technology(ICETEST-2015).
[15] Botang Shao and Peng Li, Senior Member, IEEE, "Array-Based Approximate Arithmetic Computing: A General Model and Applications to Multiplier and Squarer Design", IEEE Transactions on Circuits and Systems – 1:Regular Papers, VOL.62, no.4, April 2015.