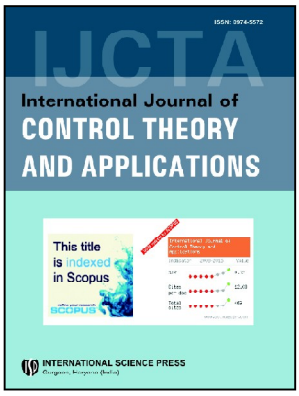

International Journal of Control Theory and Applications

ISSN : 0974-5572

Special Issue, 2016

High Speed and Low Power 16x16 bit Vedic Multiplier

[1]Y. Rama Lakshmanna, [2]G. V. S. Padma Rao, [3]N. Udaya Kumar and [4]K. Bala Sindhuri

[1] *Associate Professor, ECE Department, S.R.K.R. Engineering College, Bhimavaram, India, E-mail: yrljohnson@gmail.com*
[2] *Professor of ECE Department, S.R.K.R. Engineering College, Bhimavaram, India, E-mail: gvspadmarao@gmail.com*
[3] *Professor, ECE Department, S.R.K.R. Engineering College, Bhimavaram, India, E-mail: Nuk_07@yahoomail.com*
[4] *Asst. Professor, ECE Department, S.R.K.R. Engineering College, Bhimavaram, India, E-mail: k.b.sindhuri@gmail.com*

Abstract: This paper describes the techniques of Vedic mathematics used to improve the performance of high speedVedic multiplier. Especially the efficiency of UrdhvaTiryagbhyam it is also known as vertical and crosswise method. Urdhva -Tiryagbhyam is the most effective algorithm that gives least delay for multiplicationof all types of numbers, either small or large.It enables parallel generation of intermediate products, eliminates unwanted multiplication steps with zeros. Vedic multiplier is coded in Verilog HDL and stimulated and synthesized by using XILINX software 12.2 on Spartan 3E kit. The design of Vedic multiplier using different adder architectures arecompared in terms ofmemory, delay and power utilized.

Keywords: Urdhva-Tiryagbhyam, Binary Ripple Carry Adder (RCA), Binary to Excess-1 Code Converter (BEC), Carry Select Adder (CSLA) Power Dissipation, Propagation Delay.

1. INTRODUCTION

The desire for Very High speed and high-performance digital systems using VLSI design has become one of the greatest task in in recent years. For this cause, more techniques and algorithms have been described and illustrated to minimize the power consumption and to enhance speed of the digital systems. Most of these algorithms focused on the speed and power utilization during normal mode of operation. But it has been identified that the power utilization during testing mode is much greater than during normal operation. Because most utilized power obtained from the switching activity at the circuit nodes under test. In this paper a 16x16 bit Vedic multiplier is developed by using Vedic Mathematics related Urdhva Tiryakbhyam method. The advantage of Urdhva Tiryakbhyam method is that the power dissipation of Vedic multiplier architecture using CSLA adder is 87.14mW& total delay of the CSLA architecture is 32.34 ns. These results are improvements over power dissipations and delays obtained in different Vedic Multiplier architectures using other adders, such as RCA Adders, BEC adders are some of the standard approaches used in implementation of Vedic multiplier which are suitable for VLSI implementation [1]. The organization of paper as follows. Section II deals with Preliminaries, Section III deals with introduction of the method will be discussed, with the description of the Sutra, multiplication steps. Section IV deals with Design Steps For 16x16 Bit Vedic

Multiplier implementation, the comparison of different multipliers. Sections V and VI deals with results and conclusion [2].

2. PRELIMINARIES

(A) Basic Adders

Half adder, full adder are the fundamental building blocks to perform addition, subtraction, multiplication and divison. Half adder needs XOR gate and e AND gate.

(B) Binary Ripple Carry Adder (4 Bit)

Binary Ripple Carry Adder (4 Bit) is intended by utilizing one Half adder (2 Bit) and three Full adders (3 Bit).

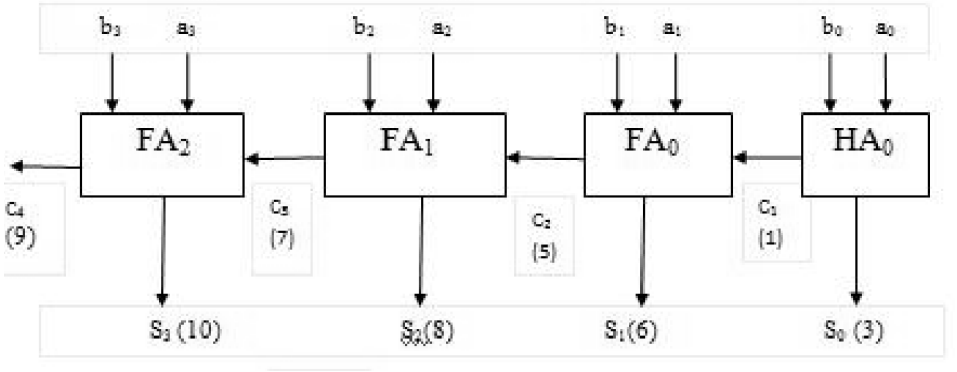

Figure 1: Ripple carry adder(4 bit) with input carry = 0

The inputs for HA are a0, bo and outputs are so, c1. Generation of s0 needs three units of delay and carry c1 needs one unit of delay.

The inputs for FA are a1, b1 and outputs are s1, c2. Generation of s1 needs six units of delay and carry c2 needs five units of delay.

The inputs for FA are a2, b2 and outputs are s2, c3. Generation of s2 needs Eight units of delay and carry c3 needs seven units of delay.

The inputs for FA are a3, b3 and outputs are s3, c4. Generation of s3 needs ten units of delay and carry c4 needs nine units of delay.

(C) 4 Bit Carry Selective Adder

The inputs for RCA_0 are b(1:0) a(1:0) and outputs are sum(1:0), $C_2(1)$. Generation of sum(1:0) needs two units of delay and carry $c_1(1)$ requires one unit of delay.

The inputs for RCA_1 are b(3:2), a(3:2) ,C_{in}=0 and outputs are $s_2(3)$, $C_2(1)$. Generation of $s_2(3)$ needs six units of delay and carry $c_2(1)$ requires one unit of delay.

The inputs for RCA_2 are b(3:2), a(3:2) C_{in}=1 and output is $s_3(3)$, $C_3(1)$. Generation of $s_3(3)$ needs three units of delay and carry $c_3(1)$ requires one unit of delay.

The inputs for MUX2are $C_2(1)$ $s_2(3)$, $s_3(3)$, $C_3(1)$ C1 are output is sum(1:0), carry .

It has three ripple carry adders and three multiplexers.

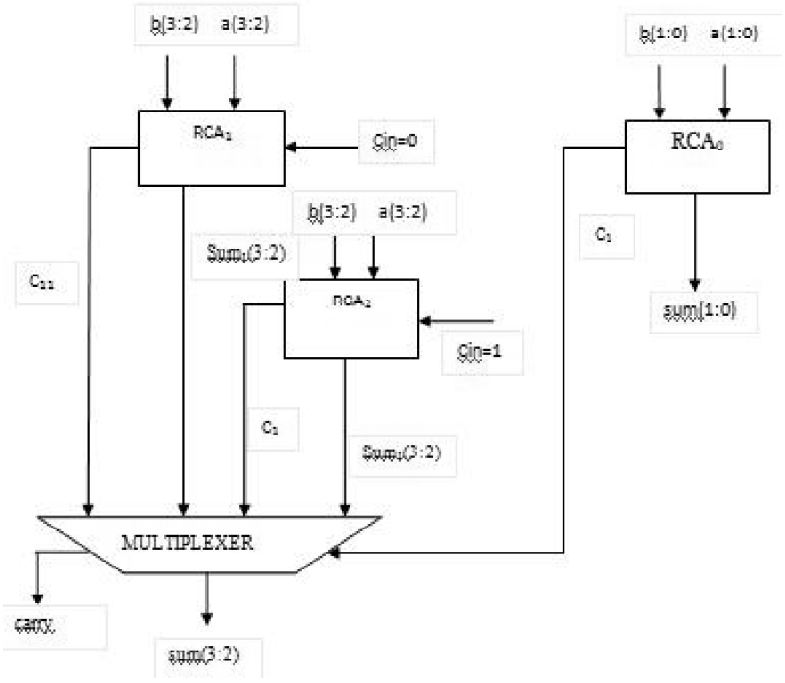

Figure 2: Carry Selective Adder(4 Bit)

RCA0 consists of two full adders.

For first full adder the inputs are a(0), b(0), c_{in} and outputs are sum(0), $C_o(0)$. Generation of sum(0) needs Six units of delay and carry $C_o(0)$ requires five units of delay.

For second full adder the inputs are a(1), b(1), $C_o(0)$. and outputs are sum(1),$C_0(1)$. Generation of sum(1) needs eight units of delay and carry $C_0(1)$ requires seven units of delay.

RCA1 consists of two full adders.

For first full adder the inputs are a(2), b(2),1'bo and outputs are sum(2),$C_1(2)$. Generation of $sum_1(2)$ needs Six units of delay and carry $C_1(2)$ requires five units of delay.

For second full adder the inputs are a(3), b(3), $C_1(2)$. and outputs are sum(3),$C_1(3)$. Generation of $sum_1(3)$ needs eight units of delay and carry $C_1(3)$ needs seven units of delay.

RCA2 consists of two full adders.

For first full adder the inputs are a(2), b(2),1'b1 and outputs are $sum_{11}(2)$,$C_{11}(2)$. Generation of $sum_{11}(2)$ needs Six units of delay and carry $C_{11}(2)$ needs five units of delay.

For second full adder the inputs are a(3), b(3), $C_{11}(2)$. and outputs are $sum_{11}(3)$,$C_{11}(3)$. Generation of $sum_{11}(3)$ needs eight units of delay and carry $C_{11}(3)$ needs seven units of delay.

Multiplexers:

For the first multiplexer the inputs are $sum_1(2)$, $sum_{11}(2)$, $C_0(1)$ and output is sum(2). Generation of sum (2) needs ten units of delay.

For the second multiplexer the inputs are $sum_1(3)$, $sum_{11}(3)$, $C_0(1)$ and output is sum(3). Generation of sum (3) needs ten units of delay.

For the third multiplexer the inputs are $C_1(3)$, $C_{11}(3)$, $C_0(1)$ and output is C_{-out} .Generation of C_{-out} needs seven units of delay [3].

GATE COUNT: Rca0 has two Full adders. Hence its gate count is 2x13=26

Rca1 has two Full adders. Hence its gate count is 2x13=26 Rca2 has two Full adders. Hence its gate count is 2x13=26 Three multiplexers having gate count is 3x4=12

Total gate count is 90.

3. INTRODUCTION TO PROPOSED METHOD

(A) Design Factors of Multiplication:

A suitable design requires the important factors are such as throughput, area, and design complexity. The measurement of number of multiplication steps can be conducted in a specific time period is known as throughput. The urdhvathiryakbhyam method in the Vedic sutras utilizes to perform most of the mathematical multiplications with fast and very convenient steps [4].

(B) Illustration steps of UrdhvaTiryakbhyamSutra

The following is the common procedural steps regarding to more multiplication problems. This sutra defined as Vertically and Crosswise multiplication method as shown in following fig. 3.

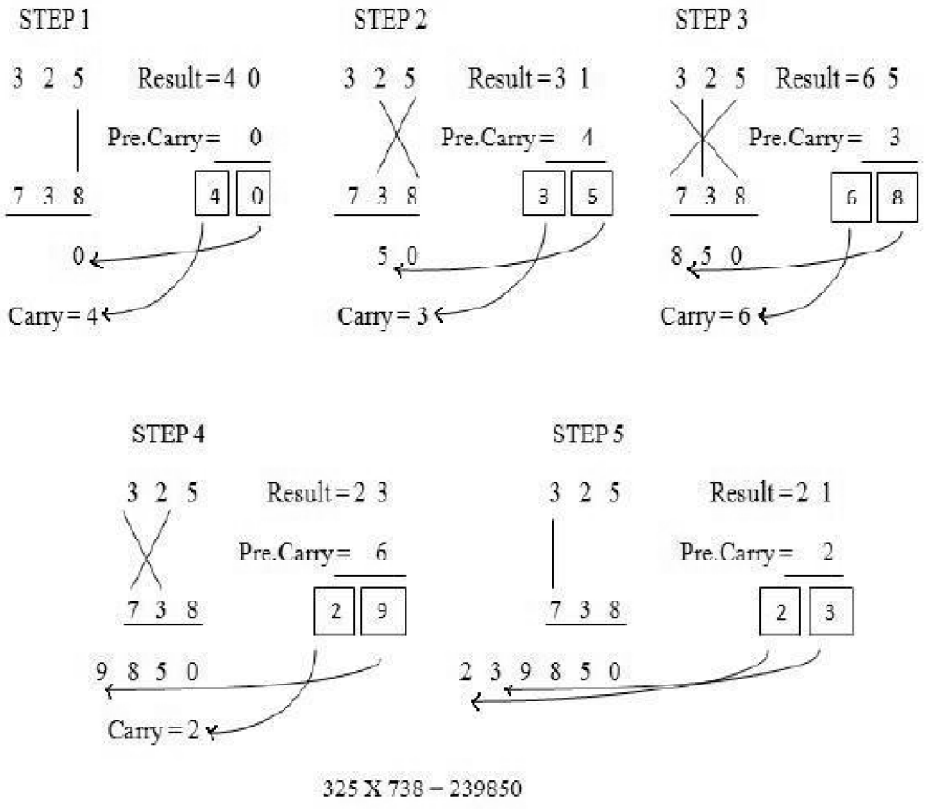

Figure 3: Multiplication steps of two decimal numbers using UrdhavaTiryakbhyam Sutra

4. DESIGN STEPS FOR 16X16 BIT VEDIC MULTIPLIER

(A) The Basic Building Module (2×2 Bit Vedic Module)

The costruction of the Basic2×2 Bit Moduleis shown in fig.4. This module is used in 4 x 4 bit Vedic Multiplier module. Generally for multiplication of two single bit binary bits we ANDed them in 2-input AND gate. For multiplication of two 2 Bit binary words the following steps are considered. LSB bits in each word are ANDed this gives LSB of the final result. Then we ANDed MSB of first word with LSB of second word and result forwarded to first half adder. Similarly we ANDed LSB of first word with MSB of second word and result is forwarded to first half adder. The half adder sum output is the adjacent bit to the LSB in final result. The half adder carry output is fed to the second half adder as one input. MSB bits in each word are ANDed and its result is forwarded to second half adder as second input. The second half adder sum output is the second adjacent bit to the LSB in final result. The second half adder carry output is considered as the MSB in final result [5].

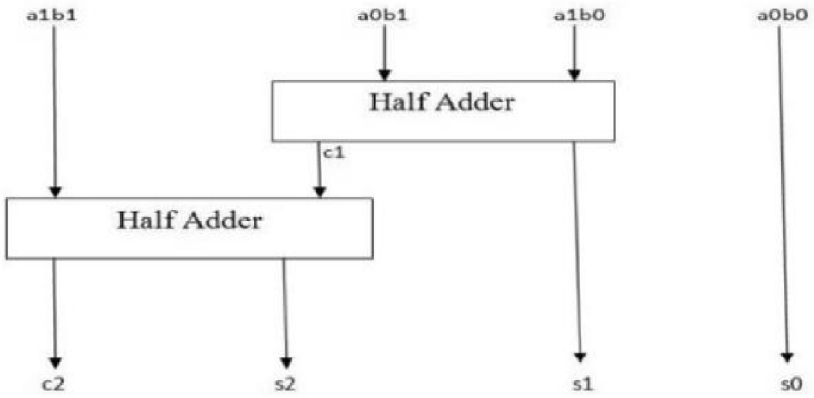

Figure 4: Vedic Multiplier 2X2 Bit using Two Half Adders

(B) Structure of 4×4 Bit Vedic Module

The structure of 4×4 Bit Vedic Multiplier module as shown in fig.5.It consists of an arrangement of four 2×2 Bit Vedic Modules are in a successive manner. The first module inputs are first two LSB bits of each 4 bit input. The

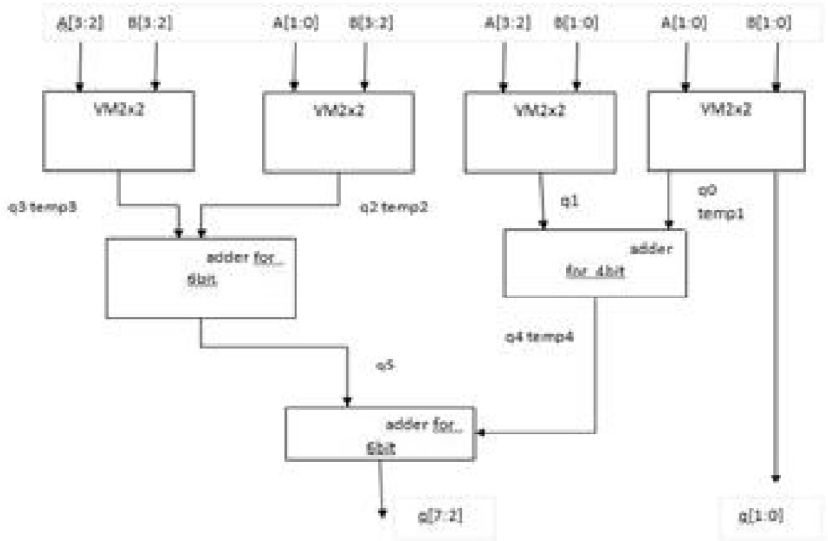

Figure 5: 4X4 Bit Vedic Multiplier using CSLA adders

second module inputs are first two LSB bits of second 4 bit input and last two MSB bits of first four bit input. The third module inputs are first two LSB bits of first 4 bit input and last two MSB bits of second 4 bit input. The fourth module inputs are last two MSB bits of each 4 bit input. These LSB and MSB bit pairs are used for the multiplication of vertical and crosswise terms [6]. Finally the outputs of these modules fed to the one 4bit adder and two 6 bit adders. The final result is a eight bit word.

(C) Structure of 8×8 Bit Vedic Module

The structure of 8×8 Bit Vedic Multiplier module as shown in fig.6.It consists of an arrangement of four 4×4 Bit Vedic Modules are in a successive manner. The first module inputs are first four LSB bits of each 8 bit input. The second module inputs are first four LSB bits of second 8 bit input and last four MSB bits of first 8 bit input. The third module inputs are first four LSB bits of first 8 bit input and last four MSB bits of second 8 bit input. The fourth module inputs are last four MSB bits of each 8 bit input. These LSB and MSB nibble(4 Bit) pairs are used for the multiplication of vertical and crosswise terms[7]. Finally the outputs of these modules fed to the one 8 bit adder and two 12 bit adders. The final result is a 16 bit word.

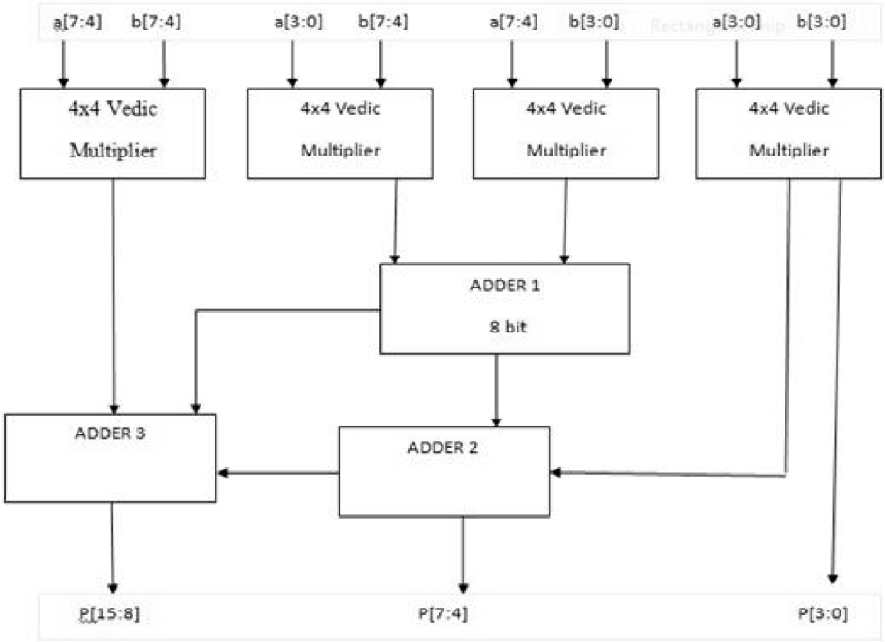

Figure 6: 8X8 Bit Vedic Multiplier using RCA adders

(D) Structure of a 16×16 Bit Vedic Module

The structure of 16×16 Bit Vedic Multiplier module as shown in fig.7. It consists of an arrangement of four 8×8 Bit Vedic Modules are in a successive manner. The first module inputs are first eight LSB bits of each 16 bit input. The second module inputs are first eight LSB bits of second 16 bit input and last eight MSB bits of first sixteen bit input. The third module inputs are first eight LSB bits of first 16 bit input and last eight MSB bits of second 16 bit input. The fourth module inputs are last eight MSB bits of each 16 bit input. These LSB and MSB byte pairs are used for the multiplication of vertical and crosswise terms. [8][9]. Finally the outputs of these modules fed to the one 16 bit adder and two 24 bit adders. The result is a 32 bit word.

Theoretical delay assessment method for Vedic multipliers by RCA, CSLA and KSA adders is been conducted. Delay values are signified in table 1.

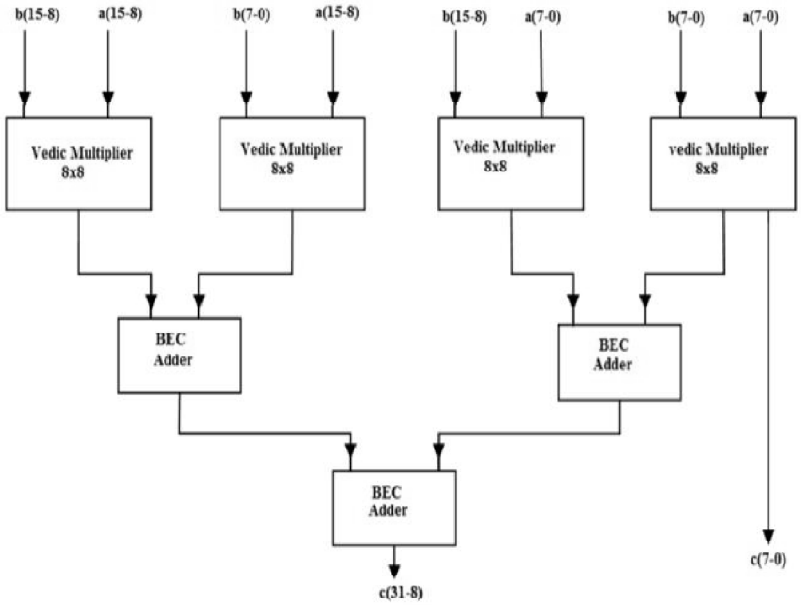

Figure 7: 16X16 Bit Vedic Multiplier using BEC adders

Table 1
Theoretical area evolution methodology for basic multiplier blocks

Vedic Multiplier Module	*Vedic Multiplier With RCA Adders*	*Vedic Multiplier With BEC Adders*	*Vedic Multiplier With CSLA Adders*
2x2 bit	16	16	16
4x4 bit	257	390	393
8x8 bit	1029	2226	2557
16x16 bit	4933	10189	11708

5. RESULTS

The intended architecture is executed by Xilinx ISIM tool for simulation on an INTEL core 2 (TM) Duo processor, 32 bit operating System, RAM 2 GB with 2.93GHZ clock frequency. At first two 4 bit inputs are considered & the outcomes are provided. Carry Select Adder, Binary ripple carry adder, koggstone adder for four bit, 8 bit, 16 bit Vedic multipliers are Simulated on Xilinx ISE 12.2. The simulation outcomes are produced via XILINX software is exposed in figure 8.

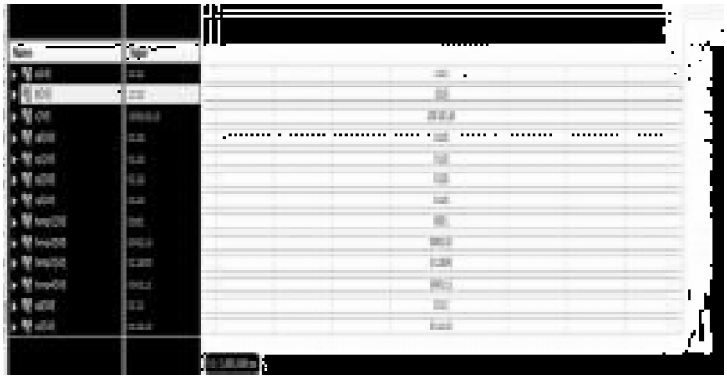

Figure 8: Simulation Results for 4 bit vedic multiplier using CSLA adders

The inputs for CSLA a(3:0) , b(3:0) ,Cin for 4 bit are considered as a "1111", "1010", '1' and the achieved output is "10010110". Following Simulation HDL Synthesis is carried out.

Table 2
Performance analysis for the device utilization summary of 16X16 Bit Vedic Multiplier using different adders

Logic utilization	*16x16 multiplier Using RCA adders*	*16x16 multiplier Using CSLA adders*	*16x16 multiplier Using BEC adders*
Power	86.73	87.14	86.92
No.of Slice Registers	493	493	493
No.of IOB's	66	66	66
No. of LUT's	1195	1221	1243
Memory in KB	213388	232072	253452
Delay in ns	38.15	32.344	36.82

From the Table 2 observed that Performance analysis for the device utilization summery of different Vedic multipliers. It is examined that CSLA architecture has lower delay once contrasted with KSA architecture.

6. CONCLUSION

In this paper "Urdhva Tiryakbhyam multiplication method" related to Vedic maths is presented. A 16x16 bit vedic multiplier using different adders such as carry save adders, carry save look ahead adder and BEC adders are implemented. Memory is also calculated for 2x2, 4x4, 8x8, 16x16 bit Vedic multipliers. Thus Vedic multiplier reduces the memory of the system. Comparison for different parameters of multipliers are also performed.been simulated using Verilog Synthesis is carried out using Xilinx ISE 12. The CSLA architecture is an efficient architecture for VLSI hardware implementation in the aspect of low area.

REFERENCES

[1] V. Priyanka Brahmaiah., L. Dhrma Teja, Dr Y. Padmasai "Study on comparison of various multipliers"-IJECET, ISSN 0976-6464 volume 4 Issue 5 September – October, 2013 pp-132-142.

[2] Y. RamaLakshmanna, Prof. G.V.S.Padma Rao, Prof. N.Udaya Kumar, K.BalaSinduri *et al*. "A Survey on Different MultiplierTechniques [SSRG-IJECE]– Volume 3 Issue 3–March 2016 ISSN: 2348 – 8352.

[3] G.Challaram, Y.RamaLakshmanna, D.Sudha Rani, K.BalaSinduri, *et al.* "Area Efficient Modified Vedic Multiplier" 2016 [ICCPCT] 978-1-5090-1276-3/16 ©2016 IEEE.

[4] S. Dhivya, Mr. T. Nallusamy "radix-8 modified booth recoder for high speed add-multiply operator". International conference on engineering trends and science & humanities [icetsh-2015] issn: 2348 8549 www.internationaljournalssrg.org page 1.

[5] Y. Rama Lakshmanna, G.V.S. Padma Rao, K. BalaSindhuri, N. Udaya Kumar "Modified Vedic Multiplier using Koggstone Adders" Vol. 5, Issue 10, October 2016 IJARCCE ISSN (Online) 2278-1021 ISSN (Print) 2319 5940 DOI10.17148/ IJARCCE.2016.51074 page no 361-371.

[6] Sulakshna Thakur#1, PardeepKumar "Area-Efficient & High Speed Ripple Carry based Vedic Multiplier" SSRG International Journal of Electronics and Communication Engineering [SSRG-IJECE] – EFES April 2015 ISSN: 2348 - 8549 http:// www.internationaljournalssrg.org Page 6.

[7] Premananda B.S. Samarth S. PaiSashank B Sashank S Bhat "Design and Implementation of 8 –bit Vedic multiplier" [IJAREEIE] Vol. 2 Issue 12, December 2013 ISSN [online] 2278-8875.

[8] Sajesh Kumar, U,MohamedSalih K.K and SajithK."Design and Implementation of Carry Select Adder without using Multiplexers" 978-4673-1627- 9112/$31.00 2012 IEEE.

[9] B.Tapasvi, Lakshmi, K.BalaSinduri, N. Udaya Kumar. Implementation of 64-Bit Kogge Stone Carry Select Adder with ZFC for Efficient Area. 978-1-4799-6085-9/15/$31.00 ©2015 IEEE.