Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

Design and FPGA Implementation of 4x4 Vedic Multiplier using Different Architectures

Samiksha Dhole
UG Scholar, Dept. of ECE,
RCOEM, Nagpur, India

Sayali Shembalkar
UG Scholar, Dept. of ECE,
RCOEM, Nagpur, India

Tirupati Yadav
UG Scholar, Dept. of ECE,
RCOEM, Nagpur, India

Prof. Prasheel Thakre
Asst. Professor, Dept. of ECE,
RCOEM, Nagpur, India

Abstract: **The need of high speed multiplier is increasing as the need of high speed processors are increasing. A Multiplier is one of the key hardware blocks in most of the fast processing systems which is not only a high delay block but also a major source of power dissipation. A conventional processor requires substantially more hardware resources and processing time in the multiplication operation, rather than addition and subtraction. This paper describes about the design of 4-bit, 8-bit and 32-bit Vedic multiplier using ancient Vedic mathematics which helps in delay and power reduction. Simulation is done in Xilinx 14.7 software using VHDL. The results for vedic multiplier using various architecture and their delay comparision is done.**

Keywords: Carry save adder, ripple carry adder, carry select adder (CLA), Vedic Mathematics, and Urdhva Tiryagbhyam.

I. INTRODUCTION

The word "Vedic" is derived from the word "Veda" which means the store-house of all knowledge. Vedic Mathematics is an ancient system of mathematics existed in India. In this eminent approach, methods of basic arithmetic are simple, powerful and logical. Another advantage is its regularity. Because of these advantages, Vedic Mathematics has become an important topic for research. The technique use in Vedic Mathematics is mainly based on sixteen Sutras. Vedic mathematics was reconstructed from the ancient Indian scriptures (Vedas) by Swami Bharati Krishna Tirthaji Maharaja (1884-1960) after his eight years of research on Vedas [1]. Vedic mathematics is mainly based on sixteen principles or word-formulae which are termed as sutras [2]. This is a very interesting field and presents some effective algorithms which can be applied to various branches of engineering such as computing and digital signal processing. Integrating multiplication with Vedic Mathematics techniques would result in the saving of computational time.

Multipliers play an essential part in today's digital signal processing and various other applications. With advances in technology, many scholars have tried and are trying to design multipliers which compromise either of the following design targets - high speed, low power consumption, symmetry of layout and less area. In this paper we have made the use Vedic sutra in designing high speed multiplier we have proposed various architecture for designing Vedic multiplier so as to reduce delay as minimum as possible using urdhva-triyagbhyam.

II. URDHVA-TRIYAGBHYAM SUTRA

The word "Urdhva Tiryakbhyam" is vedi sutra which means vertical and crosswise multiplication [1]. This multiplication formula is equally applicable to all cases of algorithm for N bit numbers. Conventionally this sutra is used for the multiplication of two numbers in decimal number system. The same concept can be applicable to binary number system. Advantage of using this type of multiplier is that as the number of bits increases, delay and area increases very slowly as compared to other conventional multipliers [3].

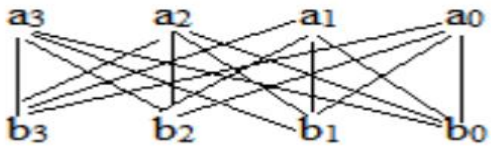

Fig -1 Example of 4x4 multiplication using Urdhva- triyakbhyam Sutra

In the above figure-1, 4-bit binary numbers *A0 A1 A2 A3* and *B0 B1 B2 B3* are considered. The result obtained is stored in *R0 R1 R2 R3 R4 R5 R6 R7*.In the first step *A0* and *B0* is multiplied and the result obtained is stored in *R*0. Similarly, in second step *[A0, B1]* and *[A1, B0]* are multiplied using a full adder and the sum is stored in *R1* and carry is transferred to next step. Likewise, the process continues till we get the result [3].

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

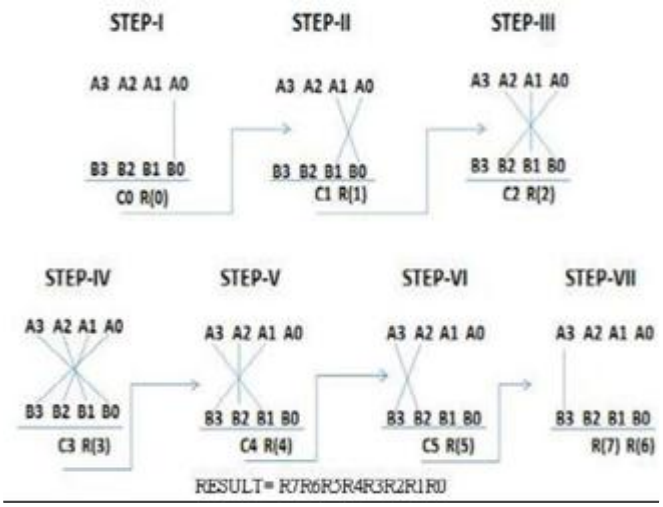

Fig-2: Multiplication method of Urdhva-Tiryakbhyam.

III. VEDIC MULTIPLIER FOR 2X2 BIT

The method is explained below for two, 2 bit numbers *A* and *B* where *A = a1a0* and *B = b1b0* as shown in Figure 3. Firstly, the Least Significant Bits are multiplied which gives the Least Significant Bit (LSB) of the final product (vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with, the product of LSB of multiplier and next higher bit of the multiplicand (crosswise) [4]. The sum gives second bit of the final product and the carry is added with the partial product obtained by multiplying the most significant bits to give the sum and carry. The sum is the third corresponding bit and carry becomes the fourth bit of the final product [4].

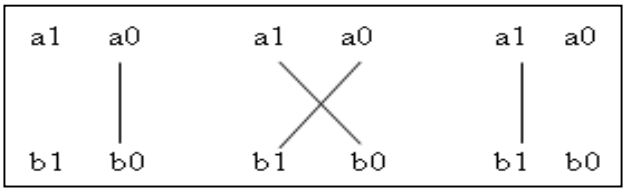

Fig-3: The Vedic Multiplication Method for two 2-bit binary numbers

s0 = a0b0;
c1s1 = a1b0 + a0b1;
c2s2 = c1 + a1b1;

Multiplier (VM) module is implemented using four input AND gates & two half-adders which is displayed in its block diagram in Figure 4 [4].

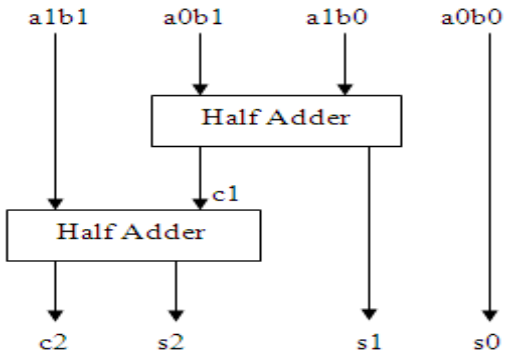

Fig-4: Block Diagram of 2x2 bit Vedic Multiplier (VM)

The same method can be extended for higher no. of input bits (say 4). But a little modification is required as discussed in section 3.2. This section illustrates the implementation of 4x4 bit VM which uses 2x2 bit VM as a basic module.

IV. VEDIC MULTIPLIER FOR 4X4 BIT

Divide the no. of bits in the inputs equally in two parts [5]. Let's analyze 4x4 bit multiplication, say multiplicand *A=A3A2A1A0* and multiplier *B= B3B2B1B0*. Following are the output line for the multiplication result, *S7S6S5S4S3S2S1S0*. Let's divide *A* and *B* into two parts, say "*A3 A2*" & "*A1 A0*" for *A* and "*B3 B2*" & "*B1B0*" for *B* [5]. Using the fundamental of Vedic multiplication, taking two bit at a time and using 2 bit multiplier block, we can have the following structure for 4x4 bit multiplication as shown in Figure 5 [5].

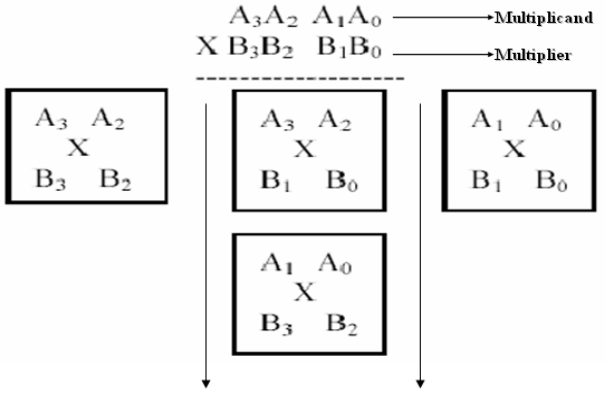

Fig-5 4x4 Vedic Multiplication Method

Each block as shown above is 2x2 bit multiplier. First 2x2 multiplier inputs are "*A1 A0*" and "*B1 B0*". The last block is 2x2 bit multiplier with inputs "*A3 A2*" and "*B3 B2*". The middle one shows two, 2x2 bit multiplier with inputs "*A3A2*" & "*B1B0*" and "*A1A0*" & "*B3B2*". So the final result of multiplication, which is of 8 bit, "*S7S6S5S4S3S2S1S0*".

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

Here we have shown various block diagram of architecture use to design 4x4 Vedic multiplier and their respective delays. The last block is 2x2 bit multiplier with inputs *"A3 A2"* and *"B3 B2"*. The middle one shows two, 2x2 bit multiplier with inputs "*A3A2*" & *"B1B0*" and *"A1A0"* & "B3B2". So the final result of multiplication, which is of 8 bit, "*S7S6S5S4S3S2S1S0"*.

Here we have shown various block diagram of architecture use to design 4x4 Vedic multiplier and their respective delays.

V. VARIOUS ARCHITECTURE OF VEDIC MULTIPLIER

In these architectures first partial products are obtained by 2x2 multipliers and then these partial products are added to obtain the result. In these architectures we have discussed various approaches to add this partial products and calculated delay for all architecture [6] [7].

1. *Vedic Multiplier for 4x4 Bit Using Ripple Carry Adder*

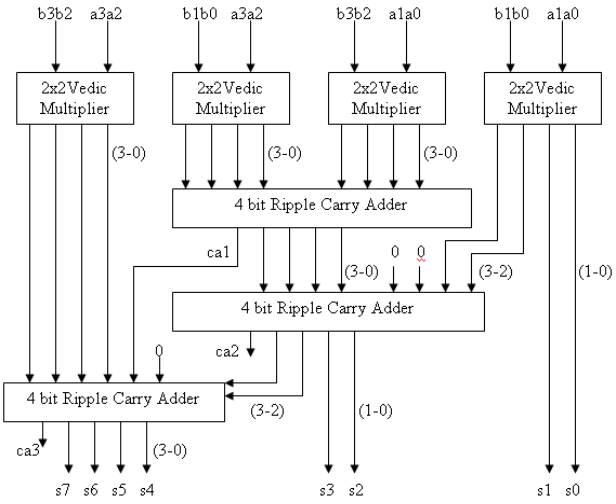

Fig-6 Vedic Multiplier for 4x4 Bit Using Ripple Carry Adder

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	26	2400	1%
Number of fully used LUT-FF pairs	0	26	0%
Number of bonded IOBs	17	102	16%

```
                                   Gate     Net
Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
----------------------------------------  ------------
 IBUF:I->O            14   1.222   1.205  a_2_IBUF (a_2_IBUF)
 LUT4:I0->O            2   0.203   0.961  r5/f1/x_y_AND_13_o1 (r5/b<0>)
 LUT6:I1->O            3   0.203   1.015  r5/f2/carry11 (r5/b<1>)
 LUT6:I0->O            4   0.203   1.048  r5/f3/Mxor_sum1_xo<0>1 (h1<2>)
 LUT6:I0->O            1   0.203   0.944  r6/f4/Mxor_sum1_xo<0>1_SW0 (N2)
 LUT6:I0->O            2   0.203   0.981  r6/f4/Mxor_sum1_xo<0>1 (a2<3>)
 LUT6:I0->O            3   0.203   0.995  r7/f2/carry11 (r7/b<1>)
 LUT6:I1->O            1   0.203   0.579  r7/f3/Mxor_sum1_xo<0>1 (p_6_OBUF)
 OBUF:I->O                 2.571          p_6_OBUF (p<6>)
----------------------------------------
Total                     12.942ns (5.214ns logic, 7.728ns route)
                                   (40.3% logic, 59.7% route)
```

Fig-7 Design Summary and Total Combinational Delay

2. *Vedic Multiplier for 4x4 Bit Using look ahead Carry Adder*

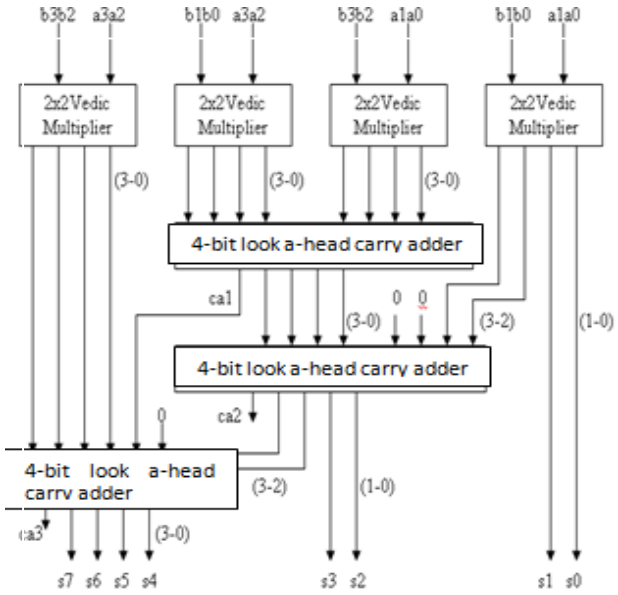

Fig-8 Vedic Multiplier for 4x4 Bit Using Look Ahead Carry Adder

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	25	2400	1%
Number of fully used LUT-FF pairs	0	25	0%
Number of bonded IOBs	17	102	16%

```
Data Path: a<2> to p<8>
                                   Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   IBUF:I->O            14   1.222   1.205  a_2_IBUF (a_2_IBUF)
   LUT4:I0->O            2   0.203   0.961  r5/c2<0>1 (r5/c2<0>)
   LUT6:I1->O            2   0.203   0.981  r5/c2<1>1 (r5/c2<1>)
   LUT6:I0->O            4   0.203   0.788  r5/Mxor_l1<2>_xo<0>1 (h1<2>)
   LUT5:I3->O            2   0.203   0.981  r6/Mxor_l1<3>_xo<0>1 (a2<3>)
   LUT6:I0->O            3   0.203   0.898  r7/c2<1>1 (r7/c2<1>)
   LUT5:I1->O            1   0.203   0.579  r7/Mxor_l1<2>_xo<0>1 (p_6_OBUF)
   OBUF:I->O                 2.571          p_6_OBUF (p<6>)
  ----------------------------------------
  Total                     11.405ns (5.011ns logic, 6.394ns route)
                                     (43.9% logic, 56.1% route)
```

Fig-9 Design Summary and Total Combinational Delay

3. *Vedic Multiplier For 4x4 Bit Using Carry save adder.*

3.1 *Carry save adder.*

Carry save adder used to perform 3 bit addition at once. Here 3 bit input *(A, B, C)* is processed and converted to 2 bit output *(S, C)* at first stage. At first stage result carry is not propagated through addition operation. In order to generate carry, implemented ripple carry adder on stage 2 for carry propagation. Carry Save adder VHDL Code can be constructed by port mapping full adder VHDL Code to 2 stage adder circuit [6].

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

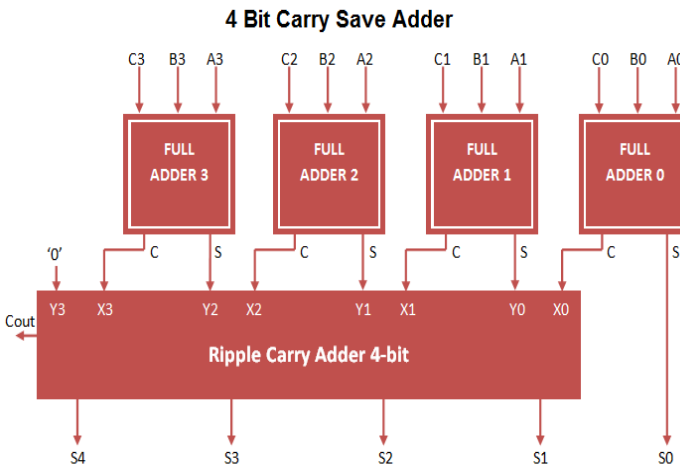

Fig-10: 4 bit carry save adder.

Using carry save adder we have used two architecture as follow:

a) Vedic Multiplier For 4x4 Bit Using two Carry Save Adder by using concede operation

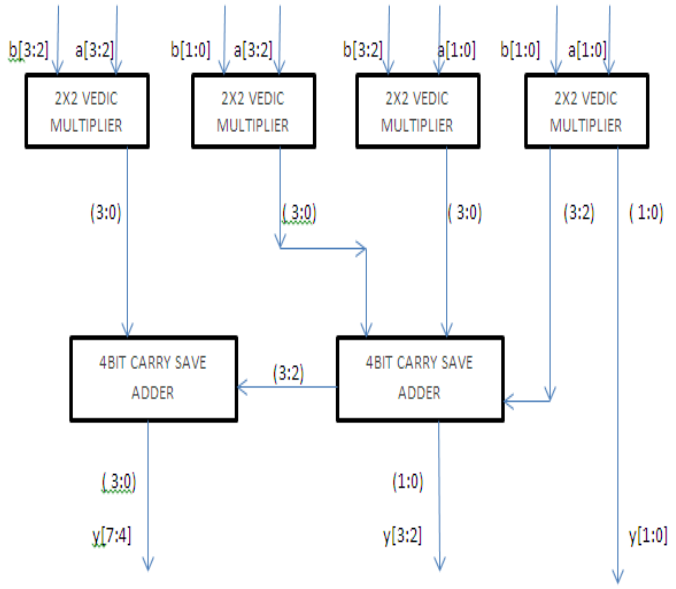

Fig-11: Vedic Multiplier For 4x4 Bit Using two Carry Save Adder by using concede operation

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	24	2400	1%
Number of fully used LUT-FF pairs	0	24	0%
Number of bonded IOBs	16	102	15%

```
  Maximum combinational path delay: 10.332ns

Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 526 / 8
-------------------------------------------------------------------------
Delay:               10.332ns (Levels of Logic = 7)
  Source:            b<1> (PAD)
  Destination:       p<6> (PAD)

  Data Path: b<1> to p<6>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             9   1.222   1.058  b_1_IBUF (b_1_IBUF)
     LUT4:I1->O            2   0.205   0.981  r2/Mxor_z<1>_xo<0>1 (q1<1>)
     LUT6:I0->O            5   0.203   0.962  r5/FA2/Cout1 (r5/X<1>)
     LUT6:I2->O            2   0.203   0.961  r6/FA2/Mxor_S_xo<0>1 (r6/Y<0>)
     LUT6:I1->O            2   0.203   0.981  r6/FA7/A_B_AND_13_o1 (r6/C1)
     LUT6:I0->O            1   0.203   0.579  r6/FA8/Mxor_S_xo<0>1 (p_6_OBUF)
     OBUF:I->O                 2.571          p_6_OBUF (p<6>)
    ----------------------------------------
    Total                     10.332ns (4.810ns logic, 5.522ns route)
```

Fig-12 Design Summary and Total combinational delay

b) Vedic Multiplier For 4x4 Bit Using single Carry Save Adder

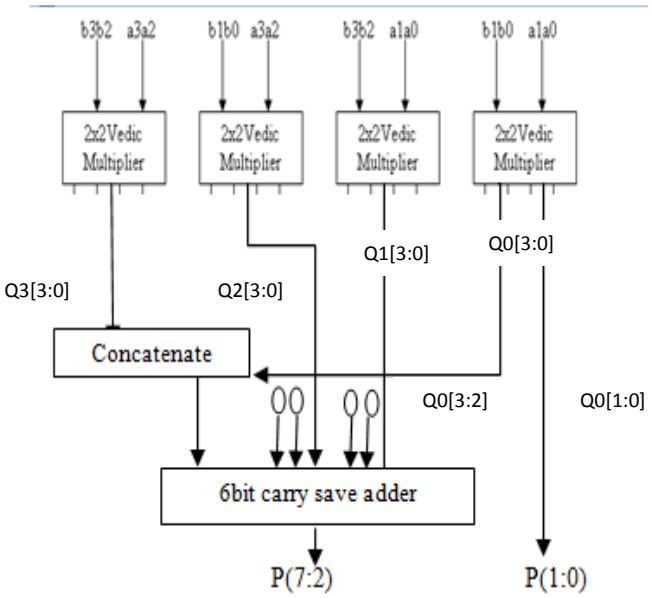

Fig-13: Vedic Multiplier For 4x4 Bit Using single Carry Save Adder by using concede operation

In this architecture instead of using two 4 bit carry save adder we have used only one 6bit carry save adder. First partial products are obtained using 2x2 Vedic multiplier, the partial product obtained from LSB 2x2 multiplier *Q0(3:0),Q0[1:0]=p[1:0]*, the remaining bits *Q[3:2]* are concatenated to bits from MSB 2x2 multiplier that is *Q3[3],Q3[2],Q3[1],Q3[0],Q0[3],Q0[2]*. Now partial products *Q1*, *Q2* are concatenated with "00" in MSB side so as to take it 6 bit. Thus three 6-bit numbers are added using single 6-bit carry look ahead adder. Hence this architecture requires only one 6-bit carry save adder instead of three 4 bit adder used in Vedic Multiplier For 4x4 Bit Using carry look ahead adder and ripple carry adder.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	27	2400	1%
Number of fully used LUT-FF pairs	0	27	0%
Number of bonded IOBs	17	102	16%

```
  Maximum combinational path delay: 9.173ns

Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 398 / 9
-------------------------------------------------------------------------
Delay:               9.173ns (Levels of Logic = 6)
  Source:            b<1> (PAD)
  Destination:       p<5> (PAD)

  Data Path: b<1> to p<5>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             11   1.222   1.111  b_1_IBUF (b_1_IBUF)
     LUT4:I1->O             2   0.205   0.981  z1/z<3>1 (q0<3>)
     LUT6:I0->O             2   0.203   0.981  aw/FA2/Cout1 (aw/X<1>)
     LUT6:I0->O             4   0.203   0.912  aw/FA8/Cout1 (aw/C2)
     LUT3:I0->O             1   0.205   0.579  aw/FA9/Mxor_S_xo<0>1 (p_5_OBUF)
     OBUF:I->O                  2.571          p_5_OBUF (p<5>)
    ----------------------------------------
    Total                       9.173ns (4.609ns logic, 4.564ns route)
                                        (50.2% logic, 49.8% route)
```

Fig-14 Design Summary and Total Combinational Delay

VI. CONCLUSION

Thus form stimulations of all above architectures we have observe following results

1. 4x4 bit Vedic multiplier.

Architecture used	Vedic Multiplier For 4x4 Bit Ripple carry adder	Vedic Multiplier For 4x4 Bit Carry look ahead adder	Vedic Multiplier For 4x4 Bit Using two Carry save adder	Vedic Multiplier For 4x4 Bit Using single Carry save adder
[Stimulation on Spartan 6]	12.942ns	11.405ns	10.932ns	9.173ns

Fig-15: 4x4 bit Vedic multiplier synthesis analysis.

2. 8x8 bit Vedic multiplier.

Architecture used	Vedic Multiplier For 8x8 Bit Carry look ahead adder	Vedic Multiplier For 8x8 Bit Using single Carry save adder
[Stimulation on Spartan 6]	18.248ns	16.629ns

Fig-16: 8x8 bit Vedic multiplier synthesis analysis.

3. 32x32 bit Vedic multiplier.

Architecture used	Vedic Multiplier For 32x32 Bit Carry look ahead adder	Vedic Multiplier For 32x32 Bit Using single Carry save adder
[Stimulation on Spartan 6]	65.442ns	49.400ns

Fig-17: 32x32 bit Vedic multiplier synthesis analysis.

Thus from these observations we can conclude that among architecture of Vedic multiplier for 4x4 using ripple carry adder, carry look ahead adder, carry save adder; carry save adder gives minimum combinational delay. And same is observed for 8x8 bit and 32x32 bit multiplication.

VII. REFERENCES

[1] Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja,"Vedic Mathematics or Sixteen Simple Mathematical formula form the veda ,delhi (1965)",motilal Banarsidas, Varanasi, India, 1986.

[2] Vitthal B. Jadhav, Charan Lal "Demystifying Speed Mathematics" First Edition May 2016.

[3] Sayali Shembalkar, Samiksha Dhole, Tirupati Yadav, Prasheel Thakre, " Vedic Mathematics Sutras -A Review", International Conference on Recent Trends in Engineering Science and Technology (ICRTEST 2017), ISSN: 2321-8169, Volume: 5 Issue: 1(Special Issue 21-22 January 2017),pg-148 -155.

[4] Anuva Das, Mrs. J. K. Kasthuri Bha, "Design Optimization of Vedic Multiplier using Reversible Logic" International Journal of Engineering Research & Technology (IJERT), Vol. 3 Issue 3, March – 2014, ISSN: 2278-0181.

[5] B.Ratna Raju, D.V.Satish, "A High Speed 16*16 Multiplier Based On Urdhva Tiryakbhyam Sutra", International Journal of Science Engineering and Advance Technology, IJSEAT, Vol 1, Issue 5, October – 2013.

[6] Vengadapathiraj, M Rajendhiran.V Gururaj, M Vinoth Kannan, A Gomathi.R, "Design Of High Speed 128x 128 Bit Vedic Multiplier Using High Speed Adder" International Journal Of Science, Engineering And Technology Research (Ijsetr) Volume 4, Issue 3, March 2015.------carry save adder

[7] Pushpalata Verma, "Design of 4x4 bit Vedic Multiplier using EDA Tool", International Journal of Computer Applications (0975 – 888) Volume 48– No.20, June 2012.-----4x4 delay comparision

[8] Ila Chaudhary, Deepika Kularia, "Design of 64 bit High Speed Vedic Multiplier", Vol. 5, Issue 5, May 2016, ISSN (Print): 2320 – 3765 ISSN (Online): 2278 – 8875.------delay comparision.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)