INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 3 - V7I3-1707)
Available online at: https://www.ijariit.com

64-bit Vedic multiplier with modified architecture and improved performance using Verilog

Madhabhatthula Prem Kumar
premkumar.madhabhatthula@gmail.com
National Institute of Technology, Jamshedpur, Jharkhand

Shiva Nand Singh
snsingh.ece@nitjsr.ac.in
National Institute of Technology, Jamshedpur, Jharkhand

ABSTRACT

Vedic Maths is the ancient system of Indian mathematics. It has been extracted from 16sutras. Out of these 16 sutras, only one of these sutras named Urdhva Tiryagbhyam is used in this project. Urdhva is meant for Vertical and Tiryagbhyam is meant for the Crosswise multiplication process. This project is about designing of 8-, 16-, 32-, 64-bit Vedic multiplier with modified architecture and improved performance in comparison with a recent journal on 32-bit Vedic multiplier. And also 64-bit Vedic multiplier according to the reference journal is designed. The design of the proposed 64-bit Vedic multiplier consists of three stages, the first stage consists of four 32-bit Vedic multipliers, second stage has 96-bitcarry save adder and 95-bit modified carry skip adder in which internal stages consists of 8-bit and 16-bit Sklansky adders. and this design is implemented in XilinxISE14.7 and obtained a delay of 19.707ns, number of Slice LUT's of 6989, power consumption of 111mW. The selected device in Xilinx14.7ISE is xc7k70t-3fbg484 from the Kintex-7 family. And also the reference design(64-bit Vedic multiplier) is calibrated using XilinxISE14.7 with the same device and obtained a delay of 36.630ns, number of Slice LUT's of 8113, power consumption of 115mW.

Keywords— *MCSKA (Modified Carry-Skip Adder), CSA(Carry-Save Adder), CSLA(Carry-Select Adder), SKA(Sklansky adder)*

1. INTRODUCTION

Multipliers are very important components in any processor. A vital role in processors and in many of the computational systems is played by multipliers. The speed of multipliers affects greatly the speed of these systems. In order to further improve the quality of the speed of the systems, multipliers with more speed and compact area are to be used. The solution for this is to implement a Vedic Multiplier having ability of performing the faster multiplications by exclusion of the unwanted steps occurring in the multiplication process. There are other important constraints such as power dissipation and area. All these three factors cannot be reduced drastically. A high-speed Vedic multiplier is implemented here. The Vedic multiplier is built using Vedic sutras, here the used Vedic sutra is Urdhva Tiryagbhyam.

2. INTRODUCTION TO VEDIC MATHS

Vedic Maths is the ancient mathematics system and it was rediscovered from Vedas by Sri Bharati Krishna Tirthaji from one of the Vedas named as the Atharva Veda. The result of his experimentation is that sixteen Sutras is the basement for the mathematics. Total Sixteen Sutras are dealt in Vedic Mathematics. The study of all these sutras is vast and their discussion is out of scope in this paper. Only one of the Sutra named "Urdhva Tiryakbhyam" has been discussed in this paper.

Urdhva Tiryakbhyam sutra: Vertically and Crosswise – is the technique used in this sutra. By using this technique about all the computations in mathematics becomes faster and easier. Compared to other conventional multipliers, the advantage of this Vedic multiplier based on this sutra is that area and delay increase at a smaller rate in comparison to others, as the number of bit's increases. proceedings, and not as an independent document.

3. LITERATURE REVIEW

In 2020 [1], Y Sri Lakshmi has built a 32-bit Vedic multiplier in which adder is Carry- select adder which is designed using a hybrid full adder. She compared Vedic multipliers that are built using a conventional full adder, compressor, and hybrid full adder in terms of Delay, Area, and Power. In this paper, she had simulated and verified this design using the DE2-115 FPGA kit.

In 2019 [2], Rohith S, Chandrashekara M N has designed an 8-bit Vedic multiplier with modified architecture which we have used in our design. They have done a comparative analysis of their design with Booth multiplier, Array multiplier, Wallace multiplier, and also they have obtained a delay of 14.219ns for their design using Spartan-6 device.

In April 2016 [3], Shiksha Pandey presented 16x16 bit Vedic-multiplier architecture using Carry-Select adder. In this paper, a high speed of about 88ns is achieved. This is quite different from the Conventional method of multiplication like addition and shifting. She has shown simulations and design implementation and used software is XilinxISE9.2i.

In 2016 [4], M. Akila, C. Gowribala, S. Maflin Shaby have designed a 16-bit Vedic multiplier in which their focus is on modification of adder. In their design adder is designed using structures of Carry-Skip Adder and Carry Select Adder. They have obtained better speed which is about 10.730ns compared to that of other Vedic multiplier architectures.

In 2016 [5], Ms. Ayushi Sharma, Mr. Ajit Singh have designed a 32-bit Vedic multi- plier using Brentkung adder and their design is synthesized and implemented on Spartan-6 XC6SLX4 FPGA board. They obtained a delay of 30.001ns.

In 2015 [6], V.Anil Kumar, S.Tamilselvan, CH.V.M.S.N.Pavan Kumar, V. Kamalkannan have designed a 16-bit Vedic multiplier and used this multiplier for the radix-2 FFT algo- rithm. They have done analysis on Carry-Skip Adder, Carry-Select Adder, Ripple-Carry Adder, and Carry Look-ahead Adder and found Carry-Select Adder gives less delay, so they have used this adder in their design of Vedic multiplier.

In 2013 [7], Shivaraj Kumar Patil, Poornima M, Shivakumar, Shridhar K P, Sanjay H presented a study on 8- bit Vedic multiplier architecture. Further, they have designed and implemented 2-,4-,8-bit Vedic multiplier. By using the Xilinx Synthesis tool, they have implemented the Verilog code for an 8-bit Vedic multiplier on the Spartan 3 kit and have obtained a delay of 28.27ns.

In 2012 [8], V. Charishma, G. Ganesh Kumar has implemented and designed a 32-bit Vedic multiplier and implemented on Spartan XC3S500-5-FG320 board and obtained a delay of about 31.526ns. They also used the Urdhva Tiryakbhyam sutra for their design.

4. ALGORITHM

Step1: First two input (each N-bit wide) binary numbers A,B are divided into half of parts i.e, A=AH-AL, B=BH-BL, Where AH-Higher word and AL-Lower word of A, BH-Higher wordand BL-Lower word of B

Step2:
- i) BL is multiplied with AL to get partial product PP0,
- ii) BL is multiplied with AH to get partial product PP1,
- iii) BH is multiplied with AL to get partial product PP2,
- iv) BH is multiplied with AH to get partial product PP3

Step3: Lower N/2 bits of PP0 is the lower N/2 bits of the final product.

Step4: Prefix N/2 0's higher word of PP1 and PP2 to get PP4 and PP5 respectively.

Step5: Concatenate PP3 and higher n/2 bits of PP0 to get PP6.

Step6: Add PP4, PP5 and PP6 to get partial sum(S) and carry(C) vectors.

Step7: Add the sum and carry vectors produced in above step to get remaining final product (3N/2) bits.

Example: 1234 x 1567 = 1933678

STAGE1: Generation of partial products

$$\begin{array}{cccc} 1\,2 & 3\,4 & 1\,2 & 3\,4 \\ \underline{\text{x }1\,5} & \underline{\text{x }1\,5} & \underline{\text{x }6\,7} & \underline{\text{x }6\,7} \\ \text{pp3=0180} & \text{pp2=0510} & \text{pp1=0804} & \text{pp0=2278} \end{array}$$

STAGE2: Generation of partial sum and partial carries

$$\begin{array}{r} 0\;1\;8\;0\;2\;2\;7\;8 \\ 0\;0\;0\;5\;1\;0\;\downarrow\;\downarrow \\ \underline{0\;0\;0\;8\;0\;4\;\downarrow\;\downarrow} \\ \text{Sum}=0\;1\;8\;3\;3\;6\;7\;8 \\ \text{Carry}=\;0\;0\;1\;0\;0\;0\;0\;0 \end{array}$$

STAGE3: Addition of partial sum and partial carries

Product = Sum + Carry = 1933678

5. REFERENCE DESIGN

5.1 Block diagram

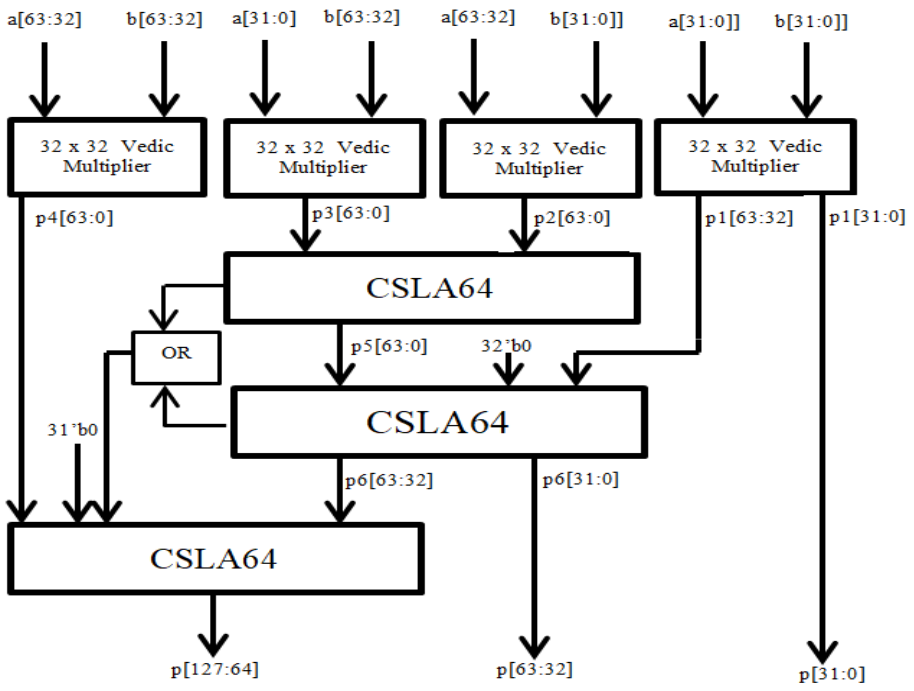

Figure 1: Block diagram of Reference 64-bit Vedic multiplier

In this block diagram, CSLA64 is 64-bit Carry Select Adder. This block diagram architecture is taken from a recent journal[1]. In journal[1], she has used Hybrid adder and Carry Select Adder in this reference Vedic multiplier and she provided only a 32-bit Vedic multiplier according to this architecture. But we have designed a 64-bit Vedic multiplier according to this reference architecture and calibrated using device xc7k70t-3fbg484 from the Kintex-7 family in XilinxISE14.7.

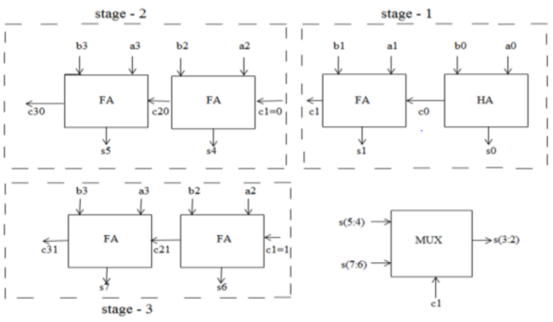

Figure 2: Block diagram of Carry-Select Adder

The above figure 2[1] shows the architecture of Adder which she had used in her paper. This adder is called a Carry-Select Adder. Here, the operation of Carry-Select Adder is in such a way that stage-1, stage-2, and stage-3 gets the inputs at the same time and produce the Sum bits at the same time but the carry out of stage-1 decides which stage has to be selected. This carryout from stage-1 is given to a multiplexer. If the carry out from stage-1 is '0', then multiplexer selects the stage-2 sum outputs. If the carry out from stage-1 is '1', then the multiplexer selects the stage-2 sum outputs. In this way, as the stages increase, carry propagation delay also increases but comparatively less with other adders. Reduction of usage of hardware and output can be obtained easily with less propagating delay [1]. Another thing she has used is a Hybrid adder. The Hybrid adder is similar in functionality to that of a normal full adder. It is formed by the combination of the multiplexer and various logic gates and this is functionally the same as a full adder. This hybrid adder contains two muxes and one XOR gate in which the XOR gate output acts as the selection line for both the MUX's[1]. The hybrid adder structure is as follows:

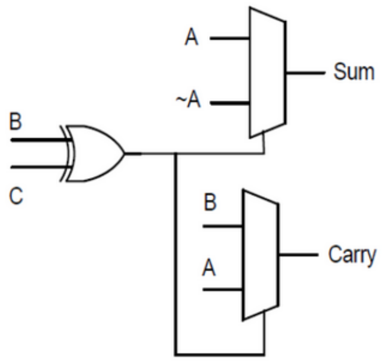

Figure 3: Hybrid full adder

This hybrid adder is further replaced with the full adder used in the CSA of Vedic multiplier and the various parameters of performance have been calculated in this paper[1].

6. PROPOSED DESIG

6.1 Block Diagram

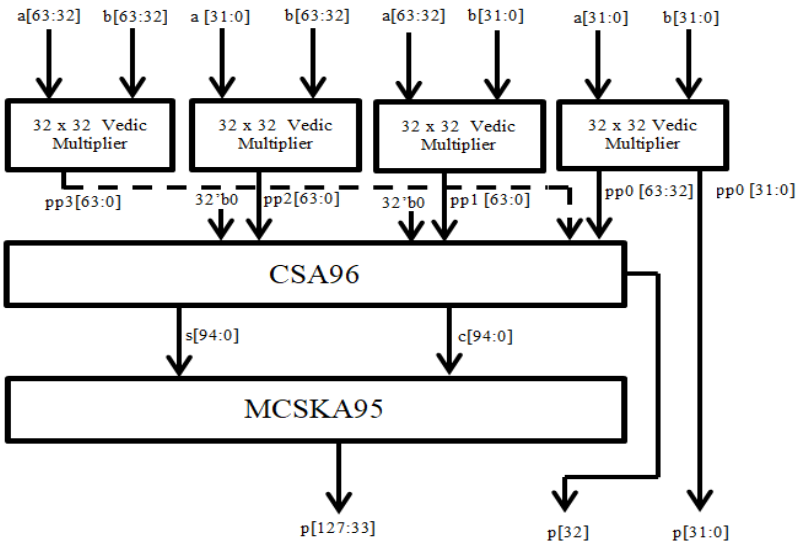

Figure 4: Block diagram of the Proposed 64 bit Vedic multiplier

The above figure4 represents the architecture of the proposed 64-bit Vedic multiplier. In this architecture, there are three stages. Stage1 consists of four 32-bit Vedic multipliers, the second stage consists of 96-bit Carry Save Adder, the third stage consists of 96-bit Modified Carry Skip Adder. This architecture of three stages is similar in the remaining Vedic multipliers. Stage1 produces partial products namely pp0(64-bit output of rightmost 32-bit Vedic multiplier), pp1, pp2, pp3(64-bit output of left most 32-bit Vedic multiplier). Here pp0[31:0] is the product bits p[31:0]. The 96-bit inputs {pp3,pp0[63:32]}, {32'b0,pp1}, {32'b0,pp2} are given to (6 bit Carry Save Adder. Now in Stage2 CSA96 will produce partial sum (s) and partial carry (c) vectors, here the least significant bit of CSA96 adder's output bit is the product bit p[32]. Finally, s and c are added in Stage3 that will produce the remaining product bits (p[127:33]). This proposed 64-bit Vedic multiplier is coded in Verilog and this design is simulated and synthesized in XilinxISE14.7 and the selected device is xc7k70t-3fbg484 from the Kintex-7 family. The proposed 64-bit Vedic multiplier produced a critical delay of 22.462ns, Slice LUT's of 6636, power consumption of 111mW.

6.2 Carry Save Adder

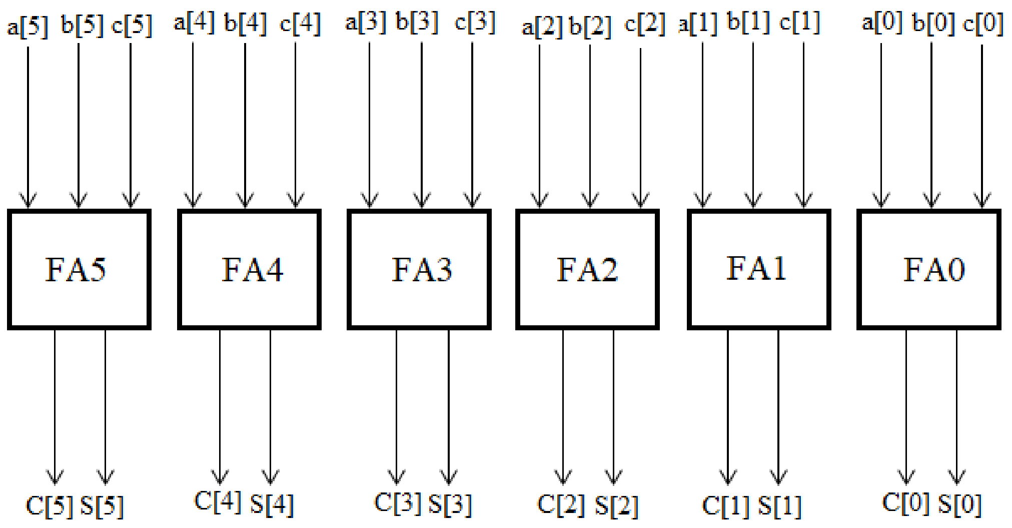

Figure 5: 6-bit Carry-save adder

The figure5 which is the block diagram representation of 6-bit Carry-save adder. This block diagram comes in the second stage of the 4-bit Vedic multiplier. It consists of six fill adders(FA0-FA5) as shown in figure6. Let there are three 6-bit binary inputs namely, a[5:0], b[5:0], and c[5:0]. The inputs to the first full Adder (FA0) are a[0], b[0], c[0] and outputs are S[0]and C[0], the inputs to the second full Adder (FA1) are a[1], b[1], c[1] and the outputs are S[1], C[1] and soon. Here all the sum and carry bits are produced within 1 full adder delay.

6.3 CSA96

CSA96 is 96-bit Carry Save Adder. It is driven by three 96-bit binary input vectors and then it produces two 96-bit outputs(s[95:0] and c[95:0]) as shown in figure4. In the next stage, these carry and sum bits are added using a 96-bit Modified Carry Skip adder.

6.4 MCSKA95

MCSKA95 is a 95-bit adder. It is derived from fixed stage Carry Increment-Carry Skip Adder[9]. It consists of variable stage-sized RCA(Ripple Carry Adder) and Increment carry blocks. It allows us a better way of using carry skip logic.

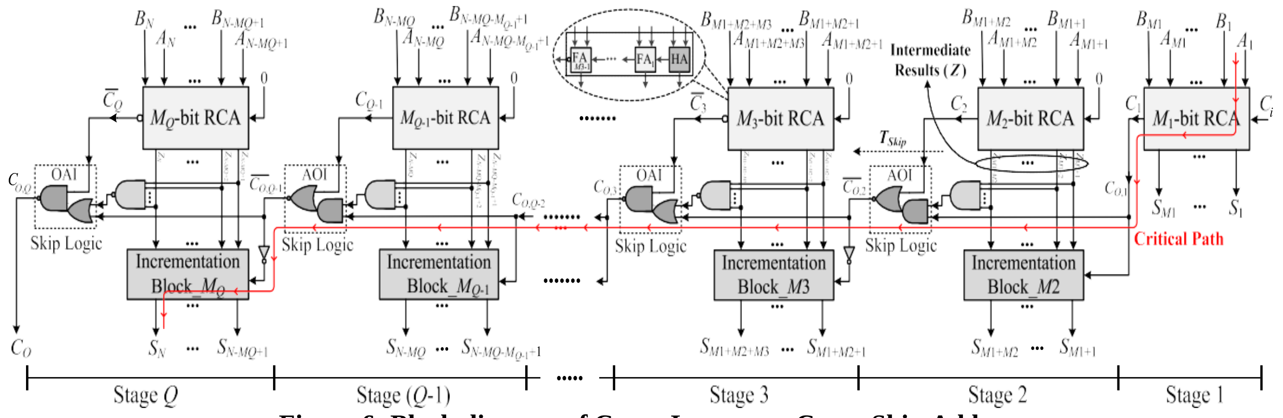

Figure 6: Block diagram of Carry Increment Carry Skip Adder

The 2:1 multiplexer position in conventional Carry Skip Adder have been occupied by compound gates (OAI/AOI as shown in figure 6). Here gates used will have a lesser number of transistors, so delay, area, and power consumptions are reduced compared to that of 2:1 multiplexer [11]. According to this structure basically the carry propagated via the skip logic will become complemented, so from the even stages, the skip logic produces the carry which is complemented[9]. Compared to conventional Carry Skip Adder, this design has significantly lower propagation delay and also slightly less area. The AOI (or OAI) compound gates consume less power compared to the power consumed by the multiplexer, however the power consumption of the Carry Increment-Carry Skip Adder is a little bit higher than the power consumption of the conventional Carry-skip adder. Since, in non-critical paths, the number of gates are increased which will impose higher wiring capacitance.

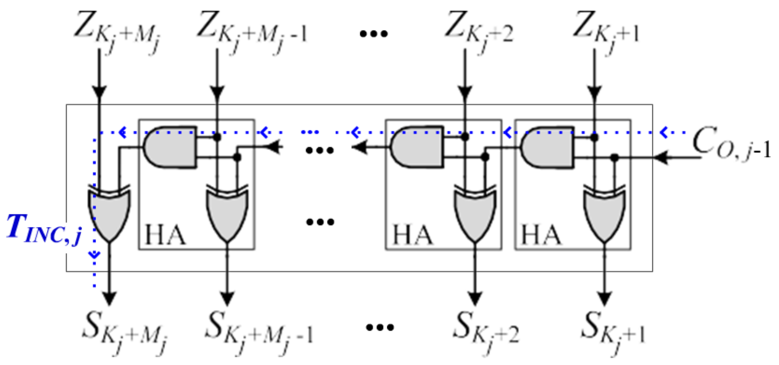

Figure 7: Block diagram of Incrementation block

Now, going deep into structure of the Fixed and Variable stage-size CI-CSKA as shown in Fig. 6 has been explained in detail. The adder has two *N*- bit wide inputs, *A* and *B*, and has *Q* stages. In each of the stage, there is an RCA block which contains *Ml (l* = 1, . . . , *Q)* bits, where *l* represents stage number. In this structure, Stage1 has only one RCA block with carry input *Ci* and the remaining stages have RCA blocks with carry input zero and also incrementation blocks(as shown in figure7). RCA block in every stage execute parallelly. This means when the first block executes the process of addition of its corresponding input bits (i.e., *AM*1,*BM*1 . . . , *A*1,*B*1*)*, and *C*i, and the remaining RCA blocks in every stage compute parallelly the intermediate results [i.e., {*ZK l*+*Ml* , . . . , *ZK l*+2, *ZK l*+1} for *Kl* =_*l*−1 *r*=1 *Mr (l* = 2, . . . , *Q)*], and also *Cl* signals. These intermediate results generated by the RCA blocks (from stages M2-MQ) and the previous stage's carry output are used in the increment block to compute the final summation of that corresponding stage. Internal structure of the incrementation block consists of a series of Half-Adders (HAs), which is shown in figure7. The carry output of the incrementation block is not used which adds to the reduction of delay considerably. As shown in Fig. 6, the *l* th stage, carry output (*CO, l)* is determined by the skip logic based on the *l* th stage's intermediate results and the previous stage's carry output (*CO, l*−1) and also on the carry (*Cl*) output of the corresponding RCA block. While calculating the *CO, l*, the cases encountered can be as follows. If *Cl* is equal to 1, *CO, l* will also be 1. And when *Cl* is equal to 0, if the intermediate results product is 1 (0), then the value of *CO, l* will be the same as *CO, j*−1 (otherwise 0). Both the compound gates such as AOI as well as OAI are used in the skip logic circuit because both of these gates inverting functions are present in the libraries of standard cells. So, by this way, the usage of an inverter gate is eliminated, otherwise the delay and power consumption will increase. As shown in Figure 6, the AOI gate is utilized as the skip logic in a stage, then in the following stage OAI gate is used as skip logic. In this adder, the carry output of the previous stage and the output carries of each block are calculated in parallel. This design is taken from paper[9]. In this proposed design, we have used the Variable Stage Size Carry Increment Carry Skip Adder of 95 bit. This is named in this paper as MCSKA95. This used adder in the proposed design has 20 stages. This adder has the block sizes as 1,2,8,8,16,16,16,16,8,2,1,1 (from Stage1-Stage12) according to figure 6, here 8-bit stage has 8-bit Sklansky adder and 16-bit stage has 16-bit Sklansky adder. This type of approach is given in paper[10]. This adder takes two 95-bit inputs and produces a sum vector which is a part of the final product vector.

7. RESULTS AND DISCUSSIONS

In the proposed 64-bit Vedic multiplier, we have coded this design in Verilog HDL. This design is simulated and implemented in XilinxISE14.7. The selected device in Xilinx14.7ISE is xc7k70t-3fbg484 from the Kintex-7 family. For the proposed design we have obtained a delay of 22.483ns, number of Slice LUT's of 6927, power consumption of 111mW. The following figures show RTL views and Simulation results of the Proposed 64-bit Vedic multiplier.

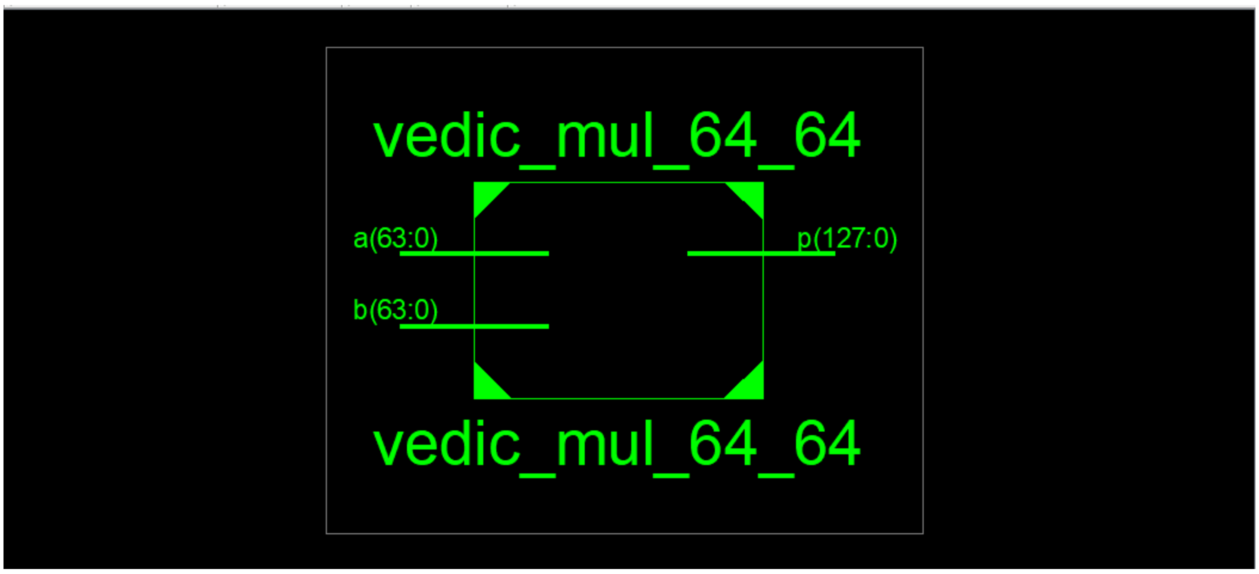

Figure 8: RTL view of 64-bit Vedic multiplier

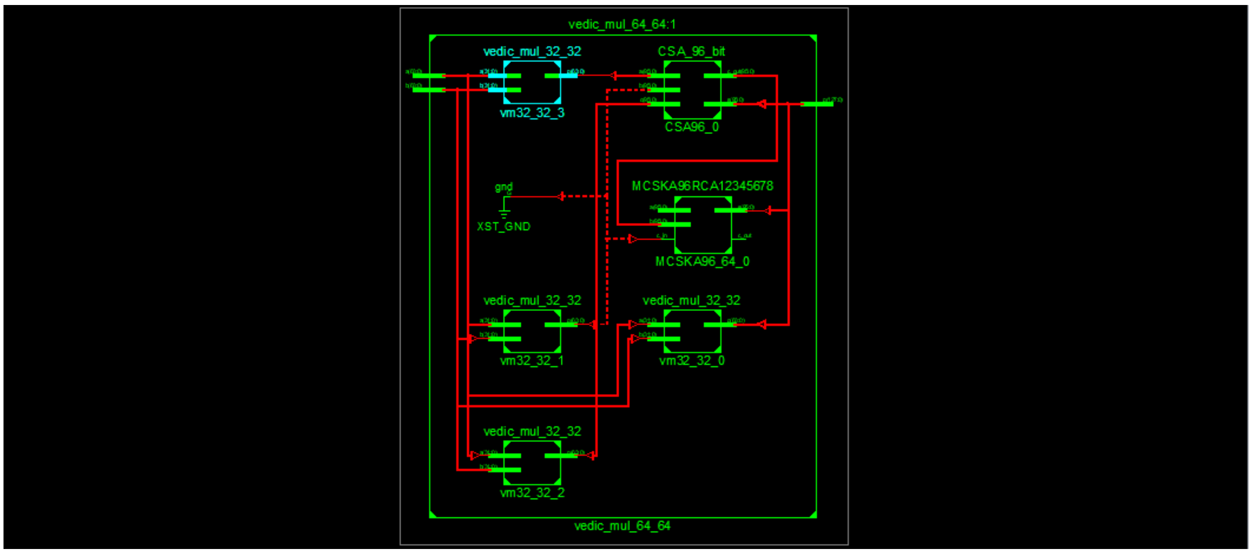

Figure 9: Insight view of 64-bit Vedic multiplier

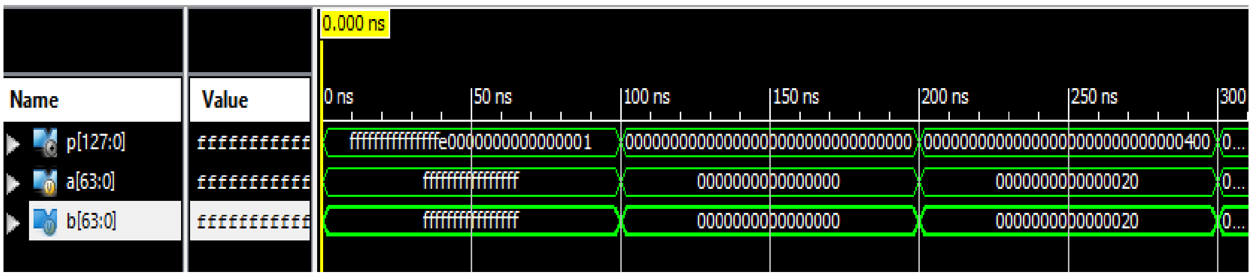

Figure 10: Simulation result of 64-bit Vedic multiplier

The below table1 provides information about the area occupied by the Proposed design and Reference designs 8-,16-,32-,64-bit Vedic multiplier.

Table1

Vedic multiplier	Reference (Slice LUT's)	Proposed (Slice LUT's)
8-bit	107	89
16-bit	474	395
32-bit	1983	1691
64-bit	8113	6989

The below table2, comparison of reference and Proposed design of 8-,16-,32-,64-bit Vedic multiplier in terms of Speed(Delay).

Table 2

Vedic multiplier	Reference (Delay in ns)	Proposed (Delay in ns)
8-bit	5.963	4.889
16-bit	9.912	8.094
32-bit	17.950	13.777
64-bit	36.630	19.707

The below table3 provides information on power consumptions of 8-,16-,32-,64-bit Vedic multipliers according to Reference and Proposed designs.

Table 3

Vedic multiplier	Reference (Power in mW)	Proposed (Power in mW)
8-bit	80	80
16-bit	80	80
32-bit	92	91
64-bit	115	111

8. CONCLUSION

Finally, 8-bit,16-bit,32-bit,64-bit Vedic multipliers have been designed in Verilog HDL and simulated in XilinxISE14.7. Urdhva Tiryagbyam sutra is very much helpful in reducing area and delay. The device used in XilinxISE14.7 is xc7k70t-3fbg484 from the Kintex-7 family. Hence, finally, 64-bit Vedic multipliers of the Proposed design are simulated and implemented using XilinxISE14.7 and obtained an 13.85% of decrease in area, 46.19% decrease in delay, 3.47% decrease in power consumption compared to the reference design. This proposed 64-bit vedic multiplier can be used with 50MHz clock operation.

.

9. FUTURE SCOPE

This architecture is further be used to implement 128-bit, 256-bit Vedic multipliers.

10. REFERENCES

[1] Y. Sri Lakshmi , Dr. T. Vigneswaran," Design and Implementation of High- Speed Vedic Multiplier Using Hybrid Full Adder", International Journal of Advanced Science and Technology Vol. 29, No. 3,Tamilnadu ,India,2020, pp. 10663 – 10669.

[2] Chandrashekara M N , Rohith S ," Design of 8 Bit Vedic Multiplier Using Urdhva Tiryagbhyam Sutra with Modified Carry Save Adder", International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Karnataka, India, MAY 2019, pp. 116-120.

[3] Shiksha Pandey, Deepak Kumar, " A Fast 16x16 Vedic Multiplier Using Carry Select Adder on FPGA", International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 4,Bhopal,India, April 2016, pp. 989-994.

[4] M. Akila, C. Gowribala, S. Maflin Shaby," Implementation of High-Speed Vedic Multiplier using Modified Adder", International Conference on Communication and Signal Processing, Jamshedpur, India, April 6-8, 2016, pp. 2244-2248.

[5] Ms. Ayushi Sharma, Mr. Ajit Singh, "Implementation of 16x16bit and 32x32bit Vedic Multiplier using FPGA board ", International Journal of Engineering Trends and Technology (IJETT), Volume-42, Number-1, Uttarpradesh, India, December 2016, pp. 1-5.

[6] S. Tamilselvan, V. Anil Kumar, V. Kamalkannan, CH.V.M.S.N.Pavan Kumar, "Design, Analysis and FPGA Implementation of N Bit Vedic Multiplier Based on Different Adder 1 Architectures", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 8, Pondicherry, India, August 2015, pp. 453-459.

[7] Poornima M, Shivaraj Kumar Patil, Shivukumar , Shridhar K P , Sanjay H, "Implementation of Multiplier using Vedic Algorithm", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-6, Bangalore, India, May 2013, pp. 219-223.

[8] G. Ganesh Kumar, V. Charishma, "Design of High Speed Vedic Multiplier using Vedic 1 Mathematics Techniques" International Journal of Scientific and Research Publications, 1 Volume 2, Issue 3, Tirupathi, India, March 2012, pp. 1-5.

[9] Milad Bahadori, Mehdi Kamal, Ali Afzali-Kusha and Massoud Pedram, "High-Speed and Energy-Efficient Carry Skip Adder Operating Under a Wide Range of Supply Voltage Levels", IEEE Transactions on Very Large Scale Integration Systems, Volume 24, No. 2, Tehran, Iran, February 2016, pp. 421-433.

[10] M. Alioto and G. Palumbo, "A Simple Strategy for Optimized Design of One-Level Carry-Skip Adders", IEEE Transactions on Circuits and Systems, Volume 50, No. 1, January 2003, pp. 141-148.

[11] J. M. Rabaey, A. Chandrakasa, and B. Nikolic, "Digital Integrated Circuits: A Design Perspective", 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003.