Proficient Architecture for Vedic Multiplier using Various VLSI Design Techniques of Optimized Adder

V. Thamizharasan
Assistant Professor, ECE
Erode Sengunthar Engineering College, Perundurai, Erode

Gokulapriya M.
U.G student, IV ECE
Erode Sengunthar Engineering College, Perundurai, Erode

Madhubala M.
U.G student IV ECE
Erode Sengunthar Engineering College, Perundurai, Erode

Divya T.
U.G student, IV ECE
Erode Sengunthar Engineering College, Perundurai, Erode

Ramya N.
U.G student, IV ECE
Erode Sengunthar Engineering College, Perundurai, Erode

ABSTRACT
Nowadays a mobile computing and multimedia applications are need for high-performance, reduced size and low-power. One of the most widely used operations in DSP is Multiplication. Different types of multipliers are available in digital. In these multipliers the Vedic multiplier is the most optimized multiplier used in signal processing module. In the existing method Vedic multiplier is designed with conventional adders, which having higher area utilization, lower speed, and power consumption.

The proposed design presents a proficient Vedic multiplier architecture for high-performance applications. Vedic multiplier architecture is consisting of adders. The proposed scheme is to changing the architecture of multiplier by memory-oriented adder cell which specifically doing add the partial product of the multiplier in memory-based approach. Which is a high speed and low area adder.

This Multiplier is functionally verified/simulated using Modelsim software and implemented to Spartan3E FPGA kit using Xilinx12.1 software. Also analyze the performance parameter like area, speed and power will be compared to conventional multiplier & adder architectures.

Keywords
DSP, Vedic multiplier Spartan 3E FPGA, Xilinx12.1

1. INTRODUCTION
In this research furnish a Multiplier and it involved in Vedic Mathematics which are known as Vedic Multiplier. In design of this multiplier, using Urdhva Triyakbhyam Sutra principle is feed into the binary number system, then build up the digital multiplier architecture [1],[2].

Apply the mathematical principle of Vedic concept to reduce number of calculations in traditional mathematics to very simple. It also contributes some powerful algorithms which can be applied to various mathematical computations in engineering.

Arithmetic adder is most important basic element for many digital applications especially for multipliers. In our design different types of adders are considered such as Conventional full adder, Carry Look ahead adder, Carry Select adder and our proposed adder [3],[4].

The rest of this paper is categorized as follows. A review of Full adders in sections II and proposed full adder is presented in sections III. The Vedic multiplier architecture is disclosed in section IV& V. Section VI presents Comparisons simulated results and comparisons of our proposed work with the existing works. Conclusion of this paper is presented in section VII.

2. CONVENTIONAL FULL ADDER
The conventional array multiplier is consisting of full adder as shown in figure 1. The congestion of full adder is consuming more power and high delay as presence of XOR gates [1]. In regular full adder critical path having 2 XOR Gates for sum output for XOR, AND & OR gate for carry output. Total delay for generating the sum is 2 XOR.

Sum = A XOR B XOR C (1)

Carry = AB + BC+CA (2)

The Figure 1 shows to gate schematic for the direct implementation.

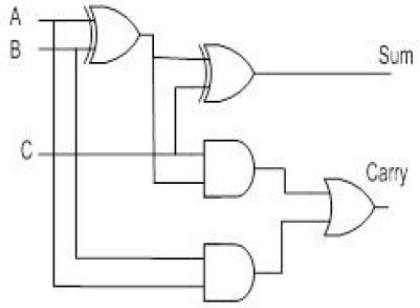

Fig. 1: Regular Full Adder structure

3. PROPOSED FULL ADDER
3.1 Full Adder using Delay Flip Flop
In Figure 2 presents that delay element orientated full adder. In that the adder apply the inputs serially with LSB is first. This adder consists of two components, in that adder and delay element. Delay element is used to store the present state of the carry, it mentioned as Ci. The Ci is fed to the input carry of the adder. This adder also gets the two inputs from Ai and Bi. The output carry is passed to input of the delay element and it stored if clock signal is applied. The Full adder is implemented in variety of structure based on our application to choose them.

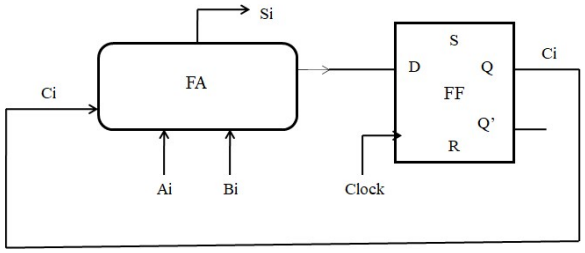

Fig.2: Memory Element based Full Adder

3.2 Multiplexer based full adder

The adder is a play an important role in most of the digital systems, to estimation of the critical path is justify the overall system sperformance. There is a concept called "Universal Logic", Universal logic can also be used to create any of the logic gates and digital circuits. MUX and Decoders are called "Universal Logic". Here we presented how a 2:1 MUX can be used to create different logic gates, like half adder and half subtractor and how a 4:1 MUX can be used to create a full adder and subtractor and all other circuits [5]. The multiplexer based proposed full adder is displayed in the Figure 3.

In full adder using multiplexer, we have to use the input as the select line of the MUX and the "0th" select line would be tied to "Logic 1" and "1st" select line would be tired to "Logic 0", Now when the select line (Input) goes to "1" the output will be "0" (inverted). Now, the output of the MUX would be "0" on high on the select line otherwise it would be "1" condition. The truth table is solved and it is simplified that the two inputs of the MUX are "0" and "1".

3.3 NAND based full adder

A NAND gate is mostly used in all the circuits because it can be implemented to any logic. Figure 4 shows that full adder using implemented only NAND gates [6]. If we need to perform number bit addition, this full adder is connection in cascade fashion.

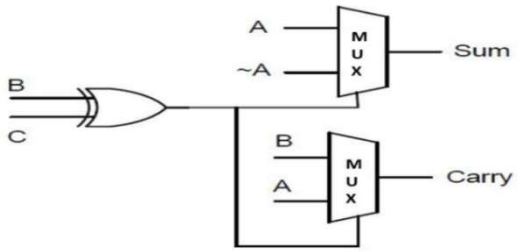

Fig. 3: Full Adder using Multiplexer

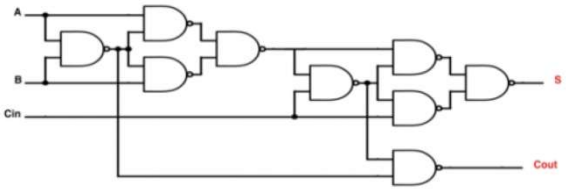

Fig. 4: NAND Based Full Adder

3.4 Memory Element and Multiplexer based Adder

Conventional full adders are replaced by multiplexer and memory element based full adder. The figure 5 shows that implementation of memory element and multiplexer based adder.

The adder is used to add the two-bit binary numbers produce the sum and carry outputs.

It can be constructed cascading of full adder with output carry of each full adder is fed into input of next stage full adder. The first and second input is represented as 'A' 'B' respectively with LSB is mentioned as 0 as subscript.

3.5 NAND based Adder

Figure 6 represents the implementation of CLA adder using the universal gate (NAND) structure. The Carry increment adder is used in the lower block of the proposed CSLA The carry increment adder receives the input from the CLA and produces the processed output in case of previous carry is one to the multiplexer. The Carry Increment adder is also implemented using the universal NAND gate only to reduce the gate count and delays.

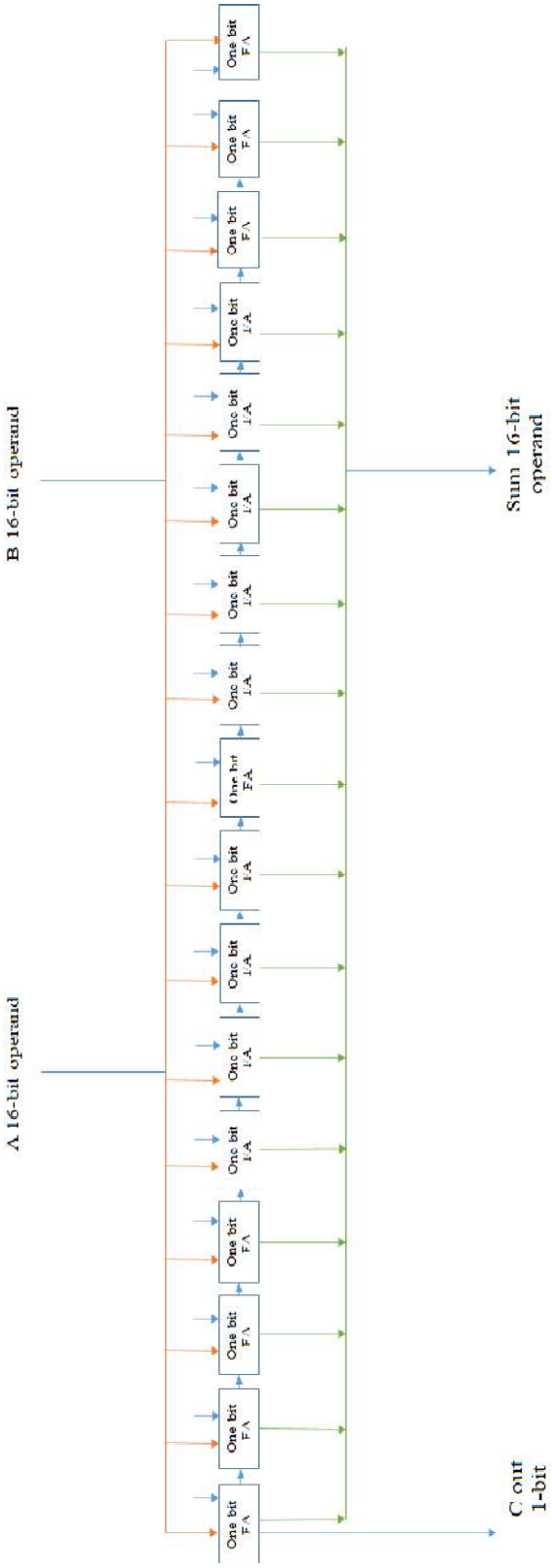

Fig. 5. Memory Element and Multiplexer Based Adder

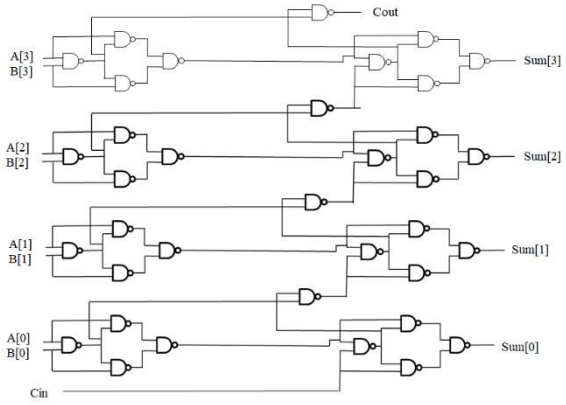

Fig.6: NAND based 4-bit CLA

Figure 7 represents the implementation of Carry increment adder using NAND gate. The 4 bit carry increment adder is shown in figure 8 is as same as the Carry Select adder in which the present sum and carry results are based on the previous stage carry.

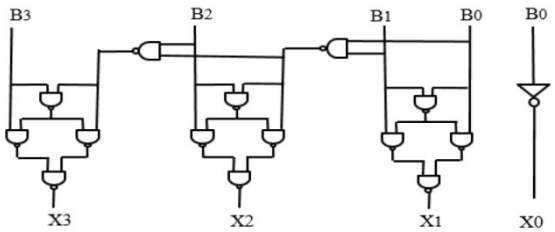

Fig.7: 4-bit Carry increment adder using NAND

If the previous carry is 0 then the output from the upper block is selected and if the previous carry is one then the output from the lower block is selected as the final output. The output from both the upper and lower block is first fed into the Quadruple multiplexer then the output for the multiplexer is selected from previous stage carry as the selector line of the Quadruple multiplexer [7].

4. VEDIC MULTIPLIER

In fundamental important arithmetic operation is multiplication. Multiplications are basically done by multiplication of input binary bits and accumulate the partial products. Efficient Multiplication is need for many digital signal processing and communication applications like convolution, Fast Fourier Transform (FFT), filtering and microprocessors. Therefore, multiplication is determining the speed of particular system. Achieving the high performance of system by achieving high through put.

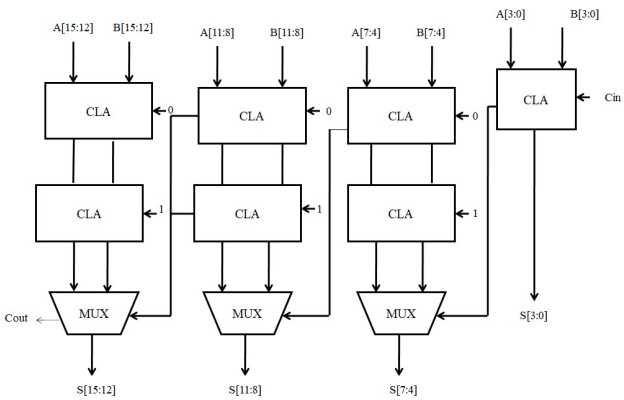

Fig.8: NAND based proposed adder

Also, one more essential requirement is power consumption for many signal processing applications. We have presented various representations of multiplier architectures [8].

In presently various multiplier architectures are available. In digital system, minimizing the power consumption by based on usage of technology, circuit style and topology of the circuit, implementation algorithm and architecture of the system. Choosing the particular architecture of multiplier based on the application and requirement. Based on the Arrangement of the components the multiplier architecture is generated.

There are two types of algorithms used in architecture of multiplier like array multiplication and booth multiplication. The computation time is very essential think. The time delay of array multiplier is very high compare other multiplier architecture because input carry of each is wait until the previous stage output carry generation. In booth multiplier is partial product accumulation stage is minimized so that time delay is reduced. Since the speed of operation is improved. Based on the requirement in DSP, the digital multipliers, is active area of research and a number of remarkable multiplication algorithms have been reported in the literature. In that literature Vedic Multiplier is one of the fast and low power multipliers.

4.1 Vedic Algorithim

The Vedic multiplier is designed using sutra algorithm called as "Urdhva Tiryakbhyam". This algorithm is normally used for multiplying two binary numbers in decimal. This proposed system is hardware implementation of binary multiplication using above principle.

This algorithm is exactly working on the principle of "Vertically and crosswise". The Partial products can be generated based on the principle of vertically and crosswise. This principle can consist of N*N bit multiplication. Since the partial products and their sums are computed for simultaneously. In this multiplier, if the no. of bits increased, the gate delay and area utilization of multiplier increase slowly as compared to the existing multiplier.

For example, the multiplication of two decimal numbers shown in figure 8 using the above algorithm. The digits on the left most bit (2 and 6) are multiplied (6 * 2 = 12) and then added with initial carry as '0'. This result (12) is the one of the digit in final multiplication result bit of LSB (2) and the carry is '1'. This carry (1) is added to the next step and this process goes on. If more than one bit is there (Step 2: 5*6,4*2) in a single step process of multiplication, all results are accumulated ((5*6) +(4*2) =38), and add with previous carry (38+previous carry (1) =39). In each step, LSB acts as the final result digit and all other bits are act as carry to the next step. Initially the carry is considered as to zero.

4.2 Design of Vedic Multiplier

The hardware implementation procedure is explained with the help of figure 9. Consider two 2-bit numbers A & B, where A=a1a0 and B=b1b0.The first step is LSB bits of two numbers are Multiplied and produced the LSB of final result. Then the next step is multiplication of LSB of multiplicand with next higher digit of multiplier. Then these results are added with multiplication of LSB of the multiplier and next higher digit of multiplicand. This result gives a final product of second digit. The same time carry is summed with partial product computed by the multiplication of MSB to give the sum and carry. This result is corresponding digit of final product [9].

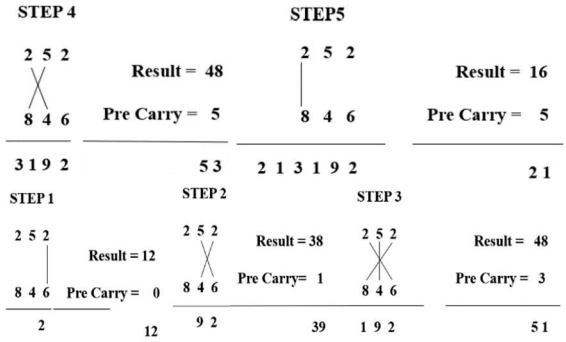

Fig. 9: Sample procedure for Vedic Algorithm

The architecture of 2*2 Vedic multiplier is shown in figure 9. It consists of 4 input AND gates & 2 half adders. Its shows that 2*2 Vedic and conventional multiplier architecture is same [10].

Hence, does not provide a significant improvement for 2*2 Vedic architecture as compare to conventional architecture. The total delay is only 2 half adders delay for generating the final product in both the cases. But 4*4-bit Vedic multiplier which uses the 2*2-bit multiplier as a basic building block. The same architecture can be expanded for higher bits of input 4, 8 & 16. As we go for higher no. of bits in input, only slight modification is required in architecture. The implementation of 16-bit proposed Vedic Multiplier is shown in Figure 10.

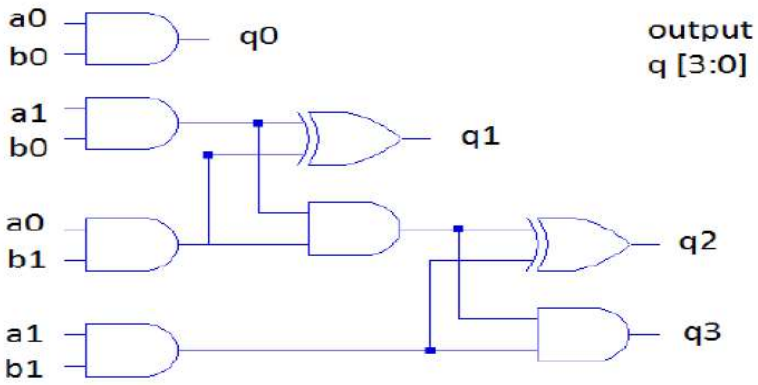

Fig. 10(a): 2*2 Vedic Multiplier

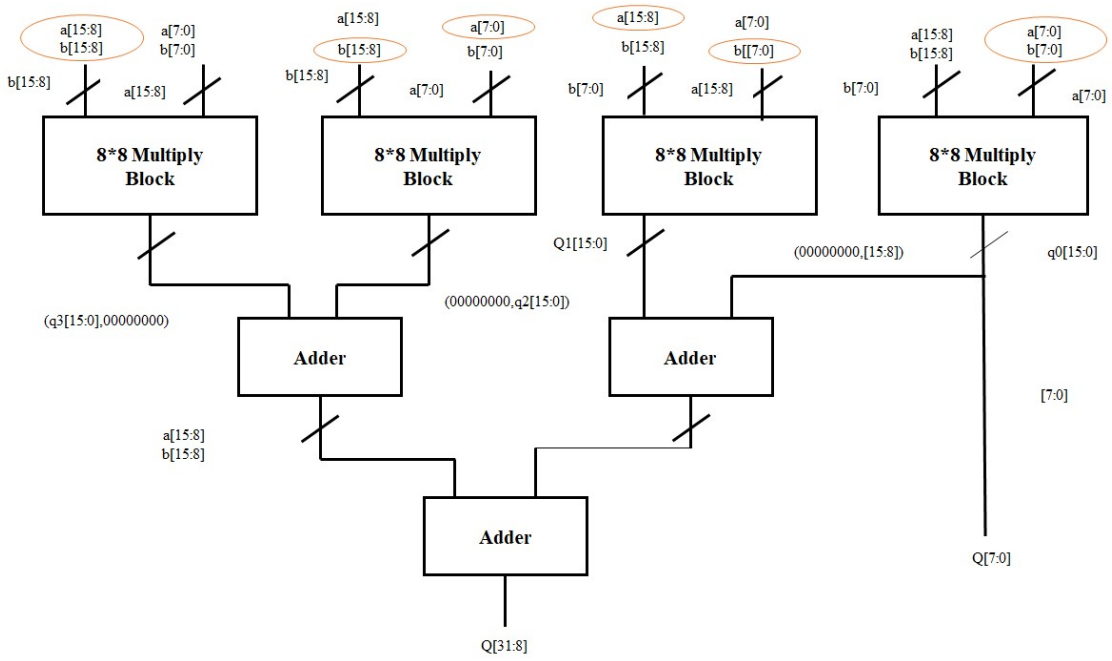

Fig. 10(b): 16-bit Proposed Vedic Multiplier

5. COMPARISON

Table –I : Comparison of Adders

S. No.	Adder techniques	No. of LUTs	Delay (ns)	ADP
1.	Conventional adder	44	28.96	1274.24
2.	Memory based adder	32	05.77	184.64
3.	MUX based adder	32	21.69	694.08
4.	NAND based adder	55	14.90	819.50
5.	BEC oriented adder	40	15.44	617.60

The table No.1 shows the performance analysis of adders. From the table we conclude that the Memory based adder is the fastest adder among all our conventional and proposed adders with the delay of 5.77ns. So, the Memory based adder is picked for the implementation of proposed Vedic Multiplier architecture. This result shows about simulation result for Existing Multiplier using Vedic concept. in which 'a','b' are inputs and 'c' is the output. For inputs a=49982 and b=30397 the output would be c=1519302854.

The Table 2 shows the investigation of Vedic Multiplier. In that our proposed Vedic Multiplier architecture has the minimum delay of 12.164ns than the existing Vedic Multiplier architecture which is the faster multiplier architecture than the existing one [11],[12],[13].

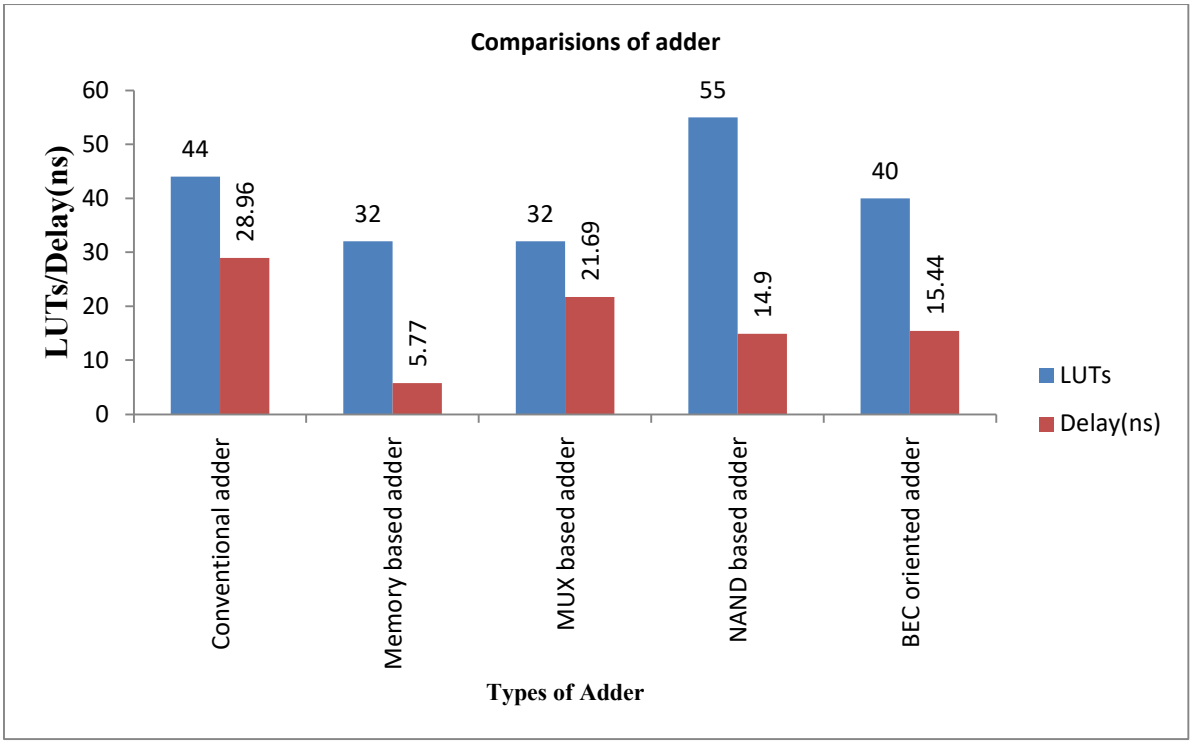

Fig. 11: Comparisons of different adder

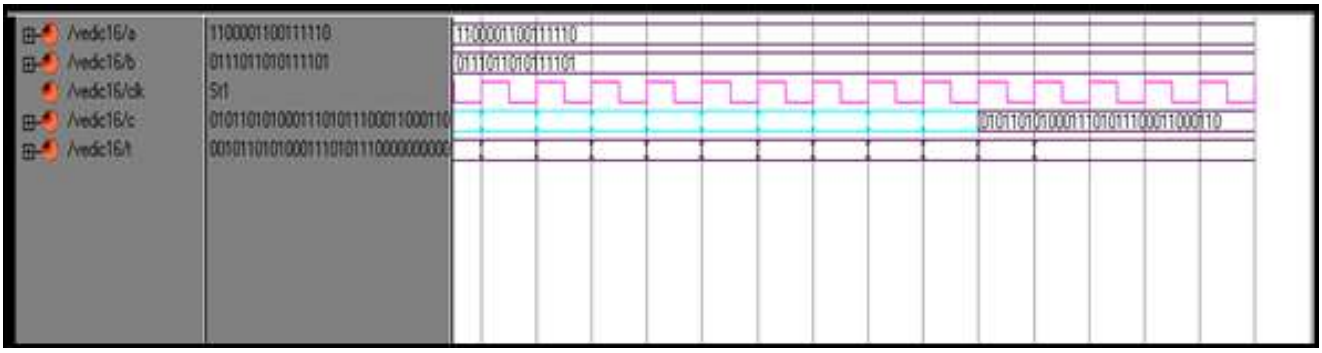

Fig. 12: Simulation results for proposed Multiplier

Table - II: Investigation of Vedic Multiplier

S. No.	Multipliers Techniques	Delay (ns)	No. of LUT	Power (uW)	ADP	PDP
	conventional Vedic Multiplier	41.88	847	80.25	35472.36	3360.87
	Proposed Vedic Multiplier	12.16	912	83.83	11089.6	1019.37

6. CONCLUSION

In this paper presents about a Vedic Multiplier using memory-based adder and is functionally verified and simulated using Modelsim software and implemented on Spartan3E FPGA kit using Xilinx software. Also, the parameters like area, delay and power will be compared to their existing system. Vedic Multiplier using memory-based adder architecture improves the speed of operation 29.723ns over existing method. In future this architecture was implementing the adaptive filter to improve the speed and area.

7. REFERENCES

[1] Chyn Wey-I, Cheng-Chen Ho, Yi-Sheng Lin, and Chien-Chang Peng., "An Area-Efficient Carry Select Adder Design by Sharing the Common Boolean Logic Term," *IMECS*, Volume II ,pp. 1091-1094.Mar.2012.

[2] Poornima M, Shivaraj Kumar Patil, Shivukumar, Shridhar K P,Sanjay H "Implementation of Multiplier using Vedic Algorithm," *Int. J. of Innovative Technology and Exploring Engineering (IJITEE)* ISSN: 2278-3075, Volume-2, Issue-6. May 2013.

[3] Premananda B.S., Samarth S. Pai, Shashank B., Shashank S. Bhat "Design and Implementation of 8-Bit Vedic Multiplier," *Int. J. of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* Volume. 2, Issue 12, Dec.2013.

[4] Jasbirkaur, lalitsood "Comparison between Various Types of Adder Topologies," *IJCST* Volume. 6, Issue 1. Mar.2015

[5] J.Eric Clapten, E. Konguvel, M. Thangamani, "VLSI

Implementation of Low Power Area Efficient Fat Carry Select Adder," *Int. J. of Communications,* Volume 115-No.6 Mar.2015

[6] H. Dorosti, A. Teymouri, S. M. Fakhraie and M. E. Salehi, "Ultralow-Energy Variation-Aware Design: Adder Architecture Study," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 3, pp. 1165-1168, March 2016.

[7] D. K. Patel, R. Chouksey and M. Saxena, "Design of fast FIR filter using compressor and Carry Select Adder," *3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 460-465, 2016.

[8] S. Srikanth, I. T. Banu, G. V. Priya and G. Usha, "Low power array multiplier using modified full adder," IEEE *International Conference on Engineering and Technology (ICETECH)*, 2016, pp. 1041-1044,2016.

[9] Rajeswar Reddy B, Lakshmi Prasad E and A. R. Reddy, "Multi precision arithmetic adders," International *Conference on Computer Communication and Informatics (ICCCI)*, 2016, pp. 1-6, 2016.

[10] V .Thamizharasan and V .Parthipan, "An Efficient VLSI Architecture for FIR Filter using Computation Sharing Multiplier," *Int. J. of Computer applications,* Vol. 54, no.14, pp.1-6, Sep.2012.

[11] V.Thamizharasan and N.Kasthuri, "High-Speed Hybrid Multiplier Design Using a Hybrid Adder with FPGA Implementation," in *IETE Journal of Research,* pp. 1-9, Apr.2021.

[12] V.Thamizharasan and N.Kasthuri, "Design of Proficient Two operand adder using Hybrid Carry Select adder with FPGA implementation*,* pp. 1-14, May.2022.

[13] V.Thamizharasan and N.Kasthuri, "FPGA implementation of high performance digital FIR filter design using a hybrid adder and multiplier*,* pp. 1-21, July.2022.