Impact Factor (SJIF): 4.542

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 4, Issue 7, July-2017

Design Approach for FPGA Implementation of 16-Bit Vedic Multiplier

Sankisha S. Moon[1], S. M. Sakhare[2]

[1]*M. Tech Scholar, Department of Department of Electronics & Communication Engineering, S. D. College of Engineering, Selukate, Wardha, Maharashtra, India*

[2]*Assistant Professor, Department of Department of Electronics & Communication Engineering, S. D. College of Engineering, Selukate, Wardha, Maharashtra, India*

Abstract — In this paper, a high speed and low power 16x16 Vedic Multiplier is designed by using low power and high speed modified carry select adder. Modified Carry Select Adder employs a newly incremented circuit in the intermediate stages of the Carry Select Adder (CSA) which is known to be the fastest adder among the conventional adder structures. A Novel technique for digit multiplication namely Vedic multiplication has been introduced which is quite different from normal multiplication by shift and addition operations. Normally a multiplier is a key block in almost all the processors and also introduces high delay block and also a major power dissipation source. This paper presents a new design methodology for less delay and less power efficient Vedic Multiplier based up on ancient Vedic Mathematic techniques. This paper presents a technique for N×N multiplication is implemented and gives very less delay for calculating multiplication results for 16×16 Vedic multiplier. In this paper, the main goal is to design the high speed and low power and area efficient Vedic multiplier based on the crosswise and vertical algorithm. Comparisons with existing conventional fast adder architectures have been made to prove its efficiency. The performance analysis shows that the proposed architecture achieves three fold advantages in terms of delay-area-power. The synthesis results of the Vedic multiplier has compared with the booth, array multiplier by different technologies. Booth multipliers are generally used for multiplication purposes. Booth Encoder, Wallace Tree, Binary Adders and Partial Product Generator are the main components used for Booth multiplier architecture. Booth multiplier is mainly used for 2 applications are to increase the speed by reduction of the partial products and also by the way that the partial products to be added. The Vedic mathematics mainly reduces the complex typical calculations in to simpler by applying sutras as stated above. These Vedic mathematic techniques are very efficient and take very less hardware to implement. These sutras are mainly used for multiplication of two decimal numbers and we extend these sutras for binary multiplications. Multiplexer is also called Universal element or Data Selector. A Multiplexer has of 2^n inputs have n select lines Basically MUX operation based on the select lines. Depending upon the select line the input is Send to the output. Multiplexers used to increase the amount of data that can be sent over the network. The values of 4 bit can be taken and remaining can be obtained from the next blocks. Like that we will obtain totally sixteen outputs and those are outputs of the sixteen bit addition.

Keywords- Vedic Mathematics, FPGA, Vedic Multiplier, MAC Unit

I. INTRODUCTION

As the FPGA architecture evolves and its complexity increases, CAD software has become more mature as well. Today, most FPGA vendors provide a fairly complete set of design tools that allows automatic synthesis and compilation from design specifications in hardware specification languages, such as Verilog or VHDL, all the way down to a bit stream to program FPGA chips. A typical FPGA design flow includes the steps and components shown in Fig. 1.

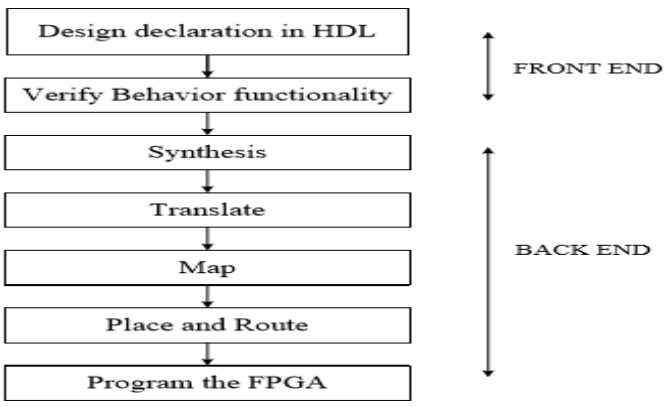

Fig 1: FPGA Design Flow

Field programmable Gate Arrays (FPGAs) are pre-fabricated silicon devices that can be electrically programmed in the field to become almost any kind of digital circuit or system. For low to medium volume productions, FPGAs provide cheaper solution and faster time to market as compared to Application Specific Integrated Circuits (ASIC) which normally require a lot of resources in terms of time and money to obtain first device. FPGAs on the other hand take less than a minute to configure and they cost anywhere around a few hundred dollars to a few thousand dollars. Also for varying requirements, a portion of FPGA can be partially reconfigured while the rest of an FPGA is still running. Any future updates in the final product can be easily upgraded by simply downloading a new application bit stream. However, the main advantage of FPGAs i.e. flexibility is also the major cause of its draw back. Flexible nature of FPGAs makes them significantly larger, slower, and more power consuming than their ASIC counterparts. These disadvantages arise largely because of the programmable routing interconnect of FPGAs which comprises of almost 90% of total area of FPGAs. But despite these disadvantages, FPGAs present a compelling alternative for digital system implementation due to their less time to market and low volume cost.
Normally FPGAs comprise of:
• Programmable logic blocks which implement logic functions.
• Programmable routing that connects these logic functions.
• I/O blocks that are connected to logic blocks through routing interconnect and that make off-chip connections.

II. LITERATURE REVIEW

The day Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho [1]introduced vedic mathematics is the name given to the ancient Indian system of mathematics that was rediscovered in the early twentieth century from ancient Indian sculptures (Vedas).The algorithms based on conventional mathematics can be simplified and even optimized by the use of Vedic Sutras. These methods and ideas can be directly applied to trigonometry, plain and spherical geometry, conics, calculus (both differential and integral), this paper introduced new multiplier and square architecture is proposed based on algorithm of ancient Indian Vedic Mathematics, for low power and high speed applications. It is based on generating all partial products and their sums in one step. The design implementation on ALTERA Cyclone –II. A multiplier architecture based on this Sutra has been developed and is seen to be similar to the popular array multiplier where an array of adders is required to arrive at the final product. Due to its structure, it suffers from a high carry propagation delay in case of multiplication of large numbers. [1]

The M. Ramalatha, Deena Dayalan, P. Dharani, Deborah Priya, reduced the load by supplementing the main processor with Co-Processors, which are designed to work upon specific type of functions like numeric computation, Signal Processing, Graphics etc. The ever increasing demand in enhancing the ability of processors to handle the complex and challenging processes has resulted in the integration of a number of processor cores into one chip. Still the load on the processor is not less in generic system. The speed of ALU depends greatly on the multiplier. In algorithmic and structural levels, numerous multiplication techniques have been developed to enhance the efficiency of the multiplier which concentrates in reducing the partial products and the methods of their addition but the principle behind multiplication remains the same in all cases.

Though there are many sutras employed to handle different sets of numeric, exploring each one gives new results. Our work has proved the efficiency of Urdhva Triyagbhyam– Vedic method for multiplication which strikes a difference in the actual process of multiplication itself. It enables parallel generation of intermediate products, eliminates unwanted multiplication steps with zeros and scaled to higher bit levels using Karatsuba algorithm with the compatibility to different data types. This sutra is to used to build a high speed power efficient multiplier in the coprocessor. [2]

L. Sriraman, T. N. Prabakar proposed multiplier can multiply two variables. A novel multiplier architecture based on ROM approach using Vedic Mathematics is proposed. This multiplier's architecture is similar to that of a Constant Co efficient Multiplier (KCM). However, for KCM one input is to be fixed. The proposed multiplier is implemented on a Cyclone III FPGA, compared with Array Multiplier and Urdhava Multiplier for both 8 bit and 16 bit cases and the results are presented. The proposed multiplier is 1.5 times faster than the other multipliers for 16x16 case and consumes only 76% area for 8x8 multiplier and 42%area for 16x16 multiplier. In the proposed multiplier for higher order bit multiplication i.e. for 16x16 and more, the multiplier is realized by instantiating the lower order bit multipliers like 8x8.This is mainly due to memory constraints. Effective memory implementation and deployment of memory compression algorithms can yield even better results. [3]

III. PROPOSED WORK

The existing work mainly focuses on development of low power and high speed adder which can be used in the Vedic multiplier for improved operation. The selected adder has good performance, but the performance can be improved in terms of delay, and power consumption.

3.1 Vedic Mathematics (Urdhwa Tiryakbhyam Sutra for binary number system)

Vedic arithmetic is separated into 16 unique sutras to perform to perform scientific estimations. Among these sutras Urdhwa Tiryakbhyam is the most ideal and adequate calculation to perform augmentation of whole numbers and in addition paired numbers. The expression "Urdhwa Tiryakbhyam" from 2 Sanskrit words Urdhwa and Tiryakbhyam which implies that "vertically" and "crosswise" separately. Give us a chance to consider the two 8 bit numbers A7-A0and B7-B0,

P0=A0*B0 (1)
C1P1=(A1*B0)+(A0*B1) (2)
C3C2P2=(A2*B0)+(A0*B2)+(A1*B1)+C1 (3)
C5C4P3=(A3*B0)+(A2*B1)+(A1*B2)+(A0*B3)+C2 (4)
C7C6P4=(A4*B0)+(A3*B1)+(A2*B2)+(A1*B3) +(A0*B4)+C3+C4 (5)
C10C9C8P5=(A5*B0)+(A4*B1)+(A3*B2)+(A2*B3)+(A1*B4)+(A0*B5)+C5+C6 (6)
C13C12C11P6=(A6*B0)+(A5*B1)+(A4*B2)+(A3*B3)+(A2*B4)+(A1*B5)+(A0*B6)+C7+C8 (7)
C16C15C14P7=(A7*B0)+(A6*B1)+(A5*B2)+(A4*B3)+(A2*B5)+(A1*B6)+(A0*B7)+C9+C11 (8)
C19C18C17P8=(A7*B1)+(A6*B)+(A5*B3)+(A4*B4)+(A3*B5)+(A2*B6)+(A1*B7)+C10+C12+C14 (9)
C22C21C20P9=(A7*B2)+(A*B3)+(A5*B4)+(A4*B5)+(A3*B6)+(A2*B7)+C13+C15+C17 (10)
C25C24C23P10=(A7*B3)+(A6*B4)+(A5*B5)+(A4*B6)+(A3*B7)+C16+C18+C20 (11)
C27C26P11=(A7*B4)+(A6*B5)+(A5*B6)+(A4*B7)+C19+C21 +C23 (12)
C29C28P12=(A7*B5)+(A5*B6)+(A5*B7)+C22+ C24+C26 (13)
C30P13=(A7*B6)+(A6*B7)+C25+C27+C28 (14)
P14=(A7*B7)+C29+C30 (15)
P15=(A7*B7) (16)

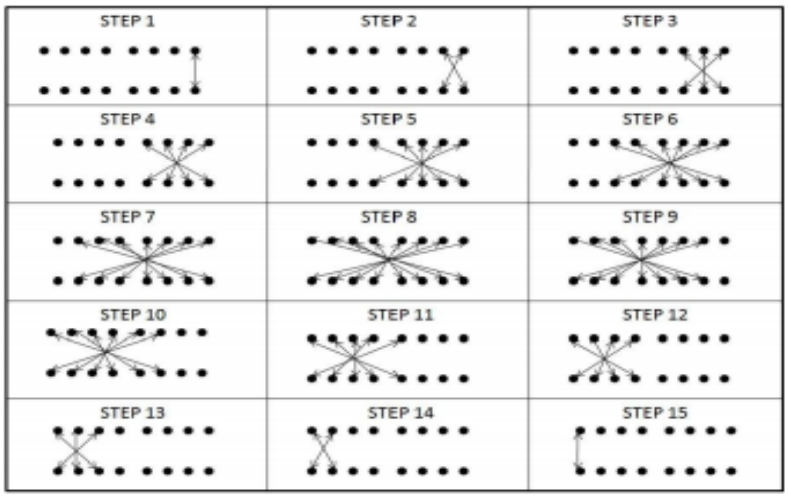

Fig 2. Line diagram for 8 bit Urdhwa multiplier

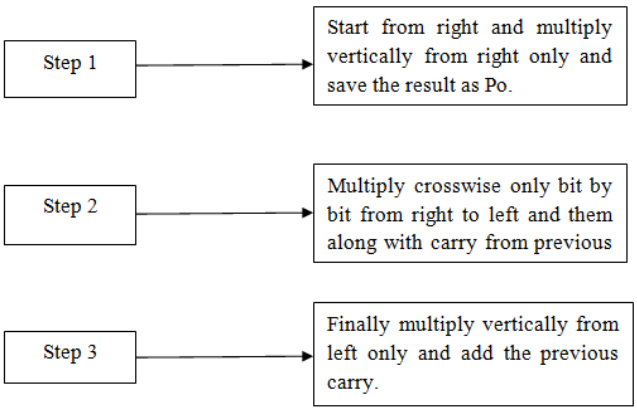

Fig 3: Steps in Urdhwa Tiryakbhyam sutra

3.2 Vedic Multiplier

The design starts first with Multiplier design that is 2x2 bit multiplier. Here, "Urdhva Tiryakbhyam Sutra" or "Vertically and Crosswise Algorithm" for multiplication has been effectively used to develop digital multiplier architecture. This algorithm is quite different from the traditional method of multiplication that is to add and shift the partial products. This Sutra shows how to handle multiplication of a larger number (N x N, of N bits each) by breaking it into smaller numbers of size (N/2 = n, say) and these smaller numbers can again be broken into smaller numbers (n/2 each) till we reach multiplicand size of (2 x 2). Thus, simplifying the whole multiplication process.

3.3 FPGA Architecture

Field-Programmable Gate Arrays (FPGAs) have become one of the key digital circuit implementation media over the last decade. A crucial part of their creation lies in their architecture, which governs the nature of their programmable logic functionality and their programmable inter-connect. FPGA architecture has a dramatic effect on the quality of the final device's speed performance, area efficiency, and power consumption. Field-Programmable Gate Arrays (FPGAs) are prefabricated silicon devices that can be electrically programmed to become almost any kind of digital circuit or system.

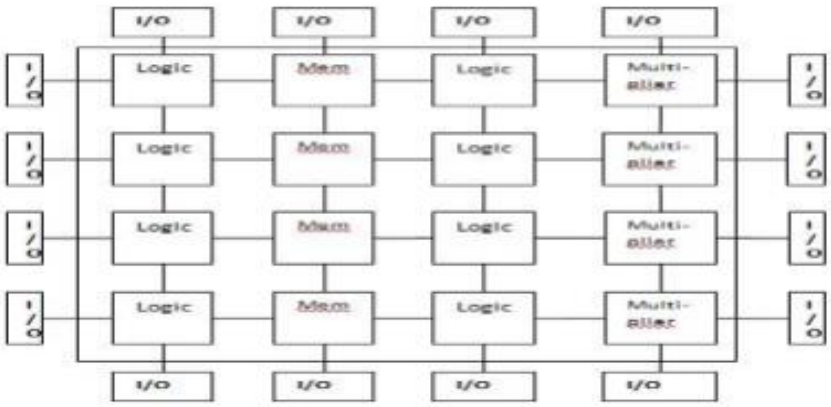

Fig 4: Basic FPGA structure.

FPGA consist of an array of programmable logic blocks of potentially different types, including general logic, memory and multiplier blocks, surrounded by a programmable routing fabric that allows blocks to be programmable interconnected. The array is surrounded by programmable input/output blocks, labelled I/O in the figure that connect the chip to the outside world. The "programmable" term in FPGA indicates an ability to program a function into the chip after silicon fabrication is complete. This customization is made possible by the programming technology, which is a method that can cause a change in the behaviour of the pre-fabricated chip after fabrication, in the "field," where system users create designs.

3.4 Digital signal processing (DSP)

Digital signal processing (DSP) is firmly being established as an extremely vibrant and vital field in the Electronics industry. The past few decades have seen an exponential growth in the number of products and applications that involve DSP, with a wide reach into diverse domains such as audio signal processing, digital image processing, video compression, speech processing, speech recognition, digital communications, RADAR, SONAR, financial signal processing, seismology and even biomedicine. Especially since computers have evolved into powerful machines capable of high computational complexity, almost all the signal processing takes place in the Digital Domain.
Frequently used algorithms include the Convolution operation, Finite Impulse Response (FIR) Filter, Infinite Impulse Response (IIR) Filter, and Fast Fourier Transform (FFT), all of which require intensive computation. DSP algorithms generally require a large number of mathematical operations to be performed quickly and repeatedly on a series of incoming data. The signals are constantly converted from analog to digital, digitally manipulated, and then converted back to analog.

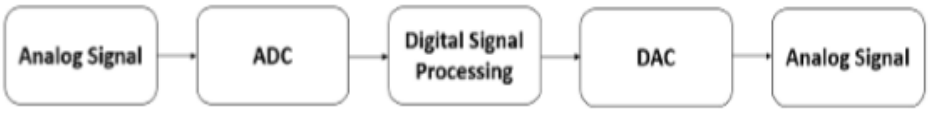

Fig 5: A typical Digital Processing System

Most general–purpose microprocessors and operating systems can execute DSP algorithms successfully, but consume more power and occupy a larger area which is not suitable for most portable applications like those on mobile phones,

biomedical devices, etc. A specialized digital signal processor, the Digital Signal Processor (DSP processor), having different architectures and features optimized specifically for digital signal processing, is hence preferred. This will tend to provide a lower-cost solution, with better performance, lower latency and lesser power consumption. Thus, the efficiency in the design of the underlying hardware in the DSP processors will reflect in the performance of the applications.

3.5 Multiply Accumulate (MAC) unit

One of the most important hardware structures in a DSP processor is the Multiply Accumulate (MAC) unit. A conventional MAC unit consists of an n−bit multiplier, the output of which is added to/subtracted from the contents of an Accumulator that stores the result. Thus, the MAC unit implements functions of the type A + BC. The ability to compute with a fast MAC unit is essential to achieve high performance in many DSP algorithms, and which is why there is at least one dedicated MAC unit in all of the modern commercial DSP processors.

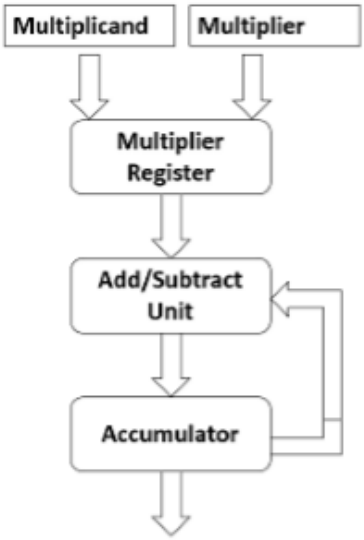

Fig 6: A MAC unit

Hence as it can be observed, Digital Multipliers are the core components of all MAC units and hence all DSP processors. The multiplier lies in the Critical Delay Path and ultimately determines the performance of any algorithm in the processor. Currently, multiplication time is still the major factor in determining the instruction cycle time of a DSP chip apart from contributing to the bulk of its power expenditure. Since multiplication drains power quickly and dominates the execution time of most DSP algorithms, there is a need for Low–Power, High–Speed Multipliers. In this concern, design of efficient multipliers has long been a topic of interest to digital design engineers. The other function that a MAC unit inherently performs is the addition operation. It is one of the most essential operations in the instruction set of any processor. Other instructions such as subtraction and multiplication employ addition in their operations, and their underlying hardware is primarily dependent on the addition hardware. Hence the performance of a design will be often be limited by the performance of its adders. It is therefore as important to choose the correct adder to implement in a design as it is to choose a multiplier because of the many factors it affects in the overall chip.The main expected features of any DSP block, be it an adder or a multiplier, are speed, accuracy and easy integrability. Power consumption, forming an active area of research.

IV. SIMULATION RESULTS

To perform the comparison in between various multipliers such as Urdhwa Tiryakbhyam, compressor based Urdhwa Tiryakbhyam, modified compressor based Urdhwa Tiryakbhyam multiplier and compressor based Urdhwa Tiryakbhyam using pipelining technique were designed implemented on Xilinx ISE Design Suite 13.2_1 Spartan 6 family for reducing the delay and improve power.

Table 1: Comparison of normal Vedic multiplier and compressor based Vedic multiplier architectures

For Delay

Type	Normal Vedic Multiplier	Vedic Multiplier	% Improvement
8*8	13.75 ns	5.60 ns	59.36 %

For Power

Type	Normal Vedic Multiplier	Vedic Multiplier	% Improvement
8*8	13.55 mw	10 uw	99.99%

It can be clearly noted from Table I., that in terms of speed, the compressor based Vedic maths multiplier performs exceptionally well and faster than the existing Vedic maths based multiplier.

Table 2 : Analysis result of LUT's, power and delay

Algorithm Used	LUT's Used	Power (W)	Delay(Nsec)
Compressor based Urdhwa Tiryakbhyam	170	13.55m	20.16
Modified Compressor based Urdhwa Tiryakbhyam Multiplier	126	13.55m	13.78
Compressor based Urdhwa Tiryakbhyam Multiplier using Pipelining	86	10u	5.60

From the synthesis report it is clear that the area utilization and total delay of the Vedic multiplier with pipelining stage is improved.

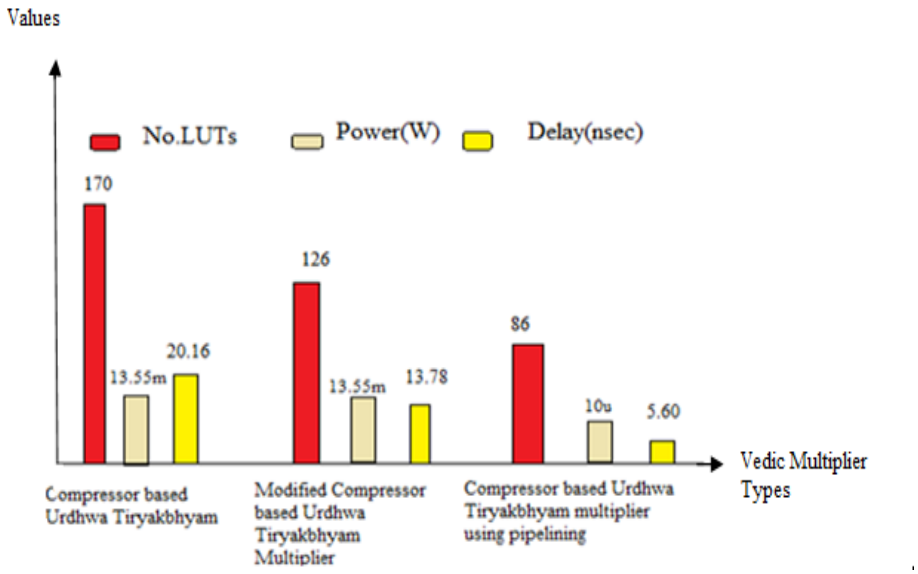

Figure 7: Graph of different 8*8 bit multipliers

The graph shows the comparison between the various based multiplier. It is observe that the compressor based Urdhwa Tiryakbhyam multiplier using pipelining structure the delay is reduced and power get improve as compared to the other multiplier used in earlier.

The multiplication of two 8 bit binary number i.e. a=01110111 and b=01110011 the output result is 0011010101110101.following figure shows the output waveform using Xilinx ISE Design Suite 13.2_1.

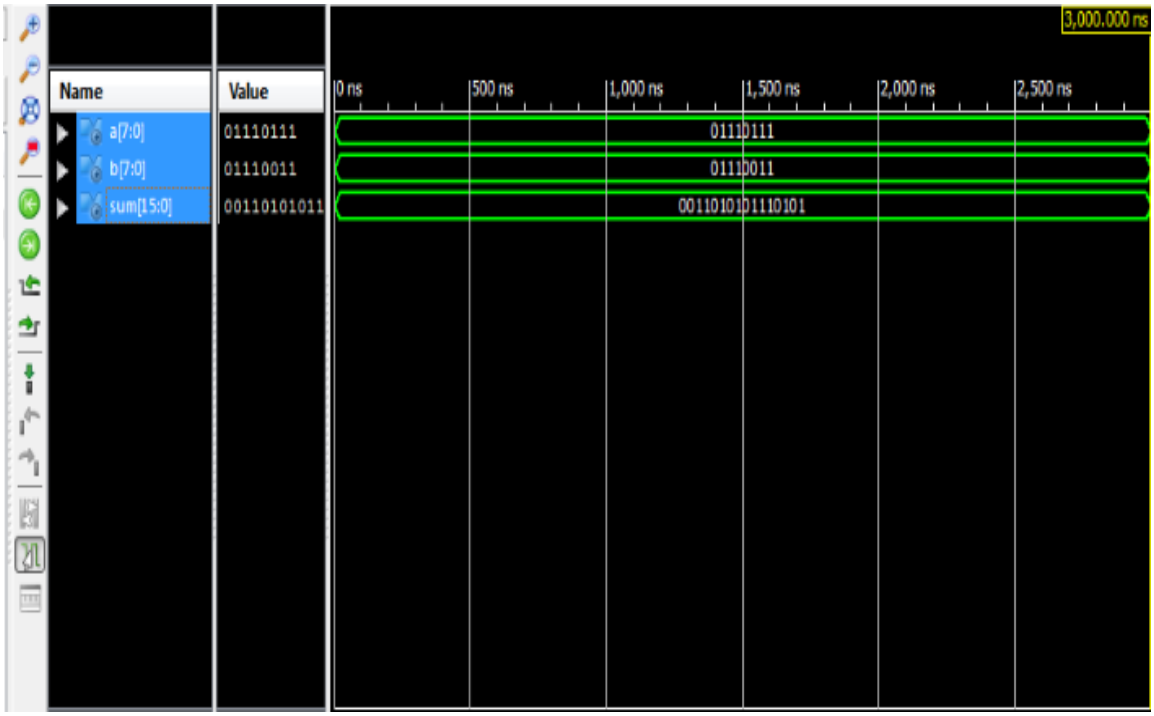

Figure 8: 16×16 bit multiplication using Vedic multiplier

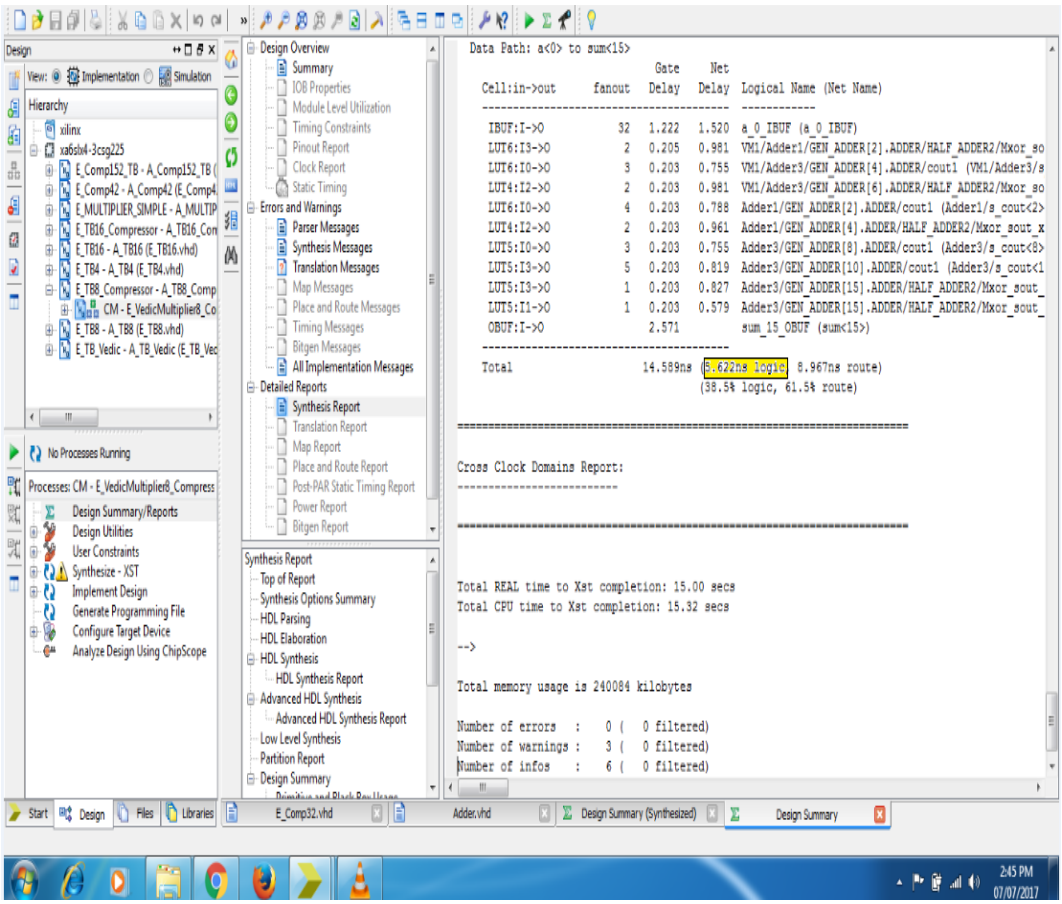

Figure 9 : Analysis of delay in 16 bit multiplier

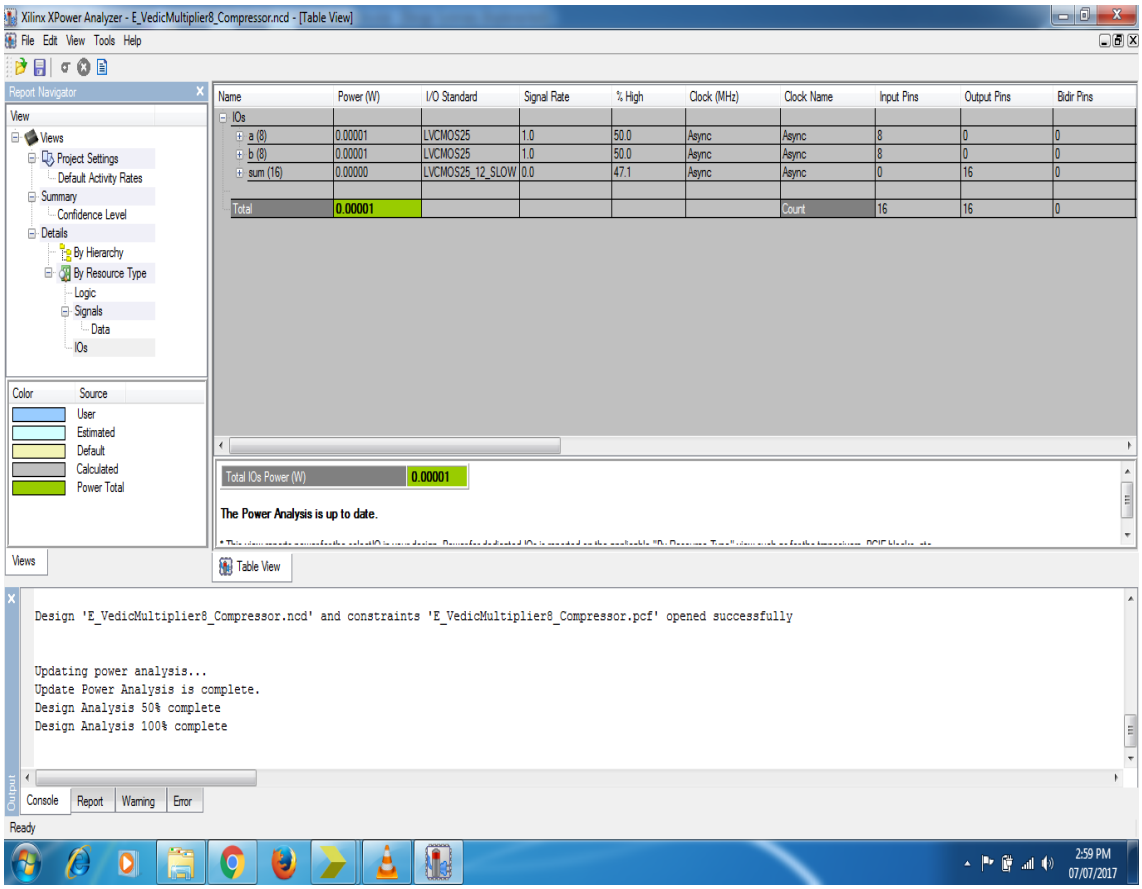

Figure 10 : Analysis of output power in 16 bit Vedic multiplier

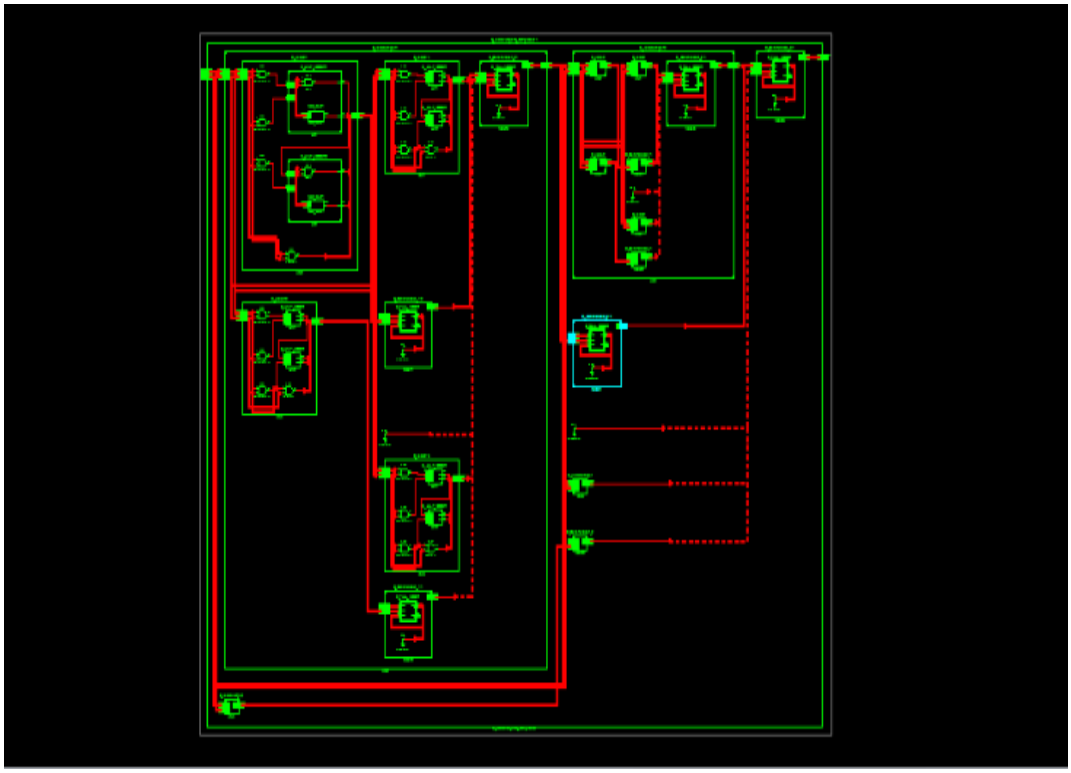

Figure 11 : RTL view of 16 bit Vedic multiplier

V. CONCLUSION & FUTURE SCOPE

This is a not worthy change as for rapid multiplier design. Also, it can be seen that, a considerable lot of the stages have now been lessened to a negligible legitimate XOR operation, with an activity to decrease area. Hence from the vedic arithmetic, augmentation of two 8 bit numbers was created. At long last it is inferred that, the altered structure gives the better execution as far as speed and range. All these multiplier plans are planned utilizing VHDL. This process is done utilizing Xilinx 13.2-1 instrument for Spartan 6E. High speed data rates are achieved in pipeline based Vedic multiplier. Vedic multiplier using compressor based urdhwa tiryakbham with pipelining gives the fast operation and it reduces the delay and almost power constant.

Vedic Mathematics, developed about 2500 years ago, gives us a clue of symmetric computation. Vedic mathematics deals with various topics of mathematics such as basic arithmetic, geometry, trigonometry, calculus etc. All these methods are very efficient as far as manual calculations are concerned. If all those methods effectively implement hardware, it will reduce the computational speed drastically. Therefore, it could be possible to implement a complete ALU using all these methods using Vedic mathematics methods. Vedic mathematics is long been known but has not been implemented in the DSP and ADSP processors employing large number of multiplications in calculating the various transforms like FFTs and the IFFTs. By using these ancient Indian Vedic mathematics methods world can achieve new heights of performance and quality for the cutting edge technology devices.

REFERENCES

[1] Jagadguru Swami Sri Bharati Krishna Tirthji Maharaja,"Vedic Mathematics", Motilal Banarsidas, Varanasi, India, 1986.

[2] Harpreet Singh Dhillon and Abhijit Mitra, "A Reduced- Bit Multiplication Algorithm for Digital Arithmetics", International Journal of Computational and Mathematical Sciences 2;2 © www.waset.org Spring 2008.

[3] Shripad Kulkarni, "Discrete Fourier Transform (DFT) by using Vedic Mathematics", report, vedicmathsindia.blogspot.com, 2007.

[4] Himanshu Thapliyal, Saurabh Kotiyal and M. B Srinivas, "Design and Analysis of A Novel Parallel Square and Cube Architecture Based On Ancient Indian Vedic Mathematics", Centre for VLSI and Embedded System Technologies, International Institute of Information Technology, Hyderabad, 500019, India, 2005 IEEE.

[5] Shamim Akhter, "VHDL Implementation of Fast NXN Multiplier Based on Vedic Mathematics", Jaypee Institute of Information Technology University, Noida, 201307 UP, INDIA, 2007 IEEE.

[6] Himanshu Thapliyal and M.B Srinivas, "An Efficient Method of Elliptic Curve Encryption Using Ancient Indian Vedic Mathematics", IEEE, 2005.

[7] El hadj youssef wajih, Zeghid Medien, Machhout Mohsen, Bouallegue Belgacem,Tourki Rached,"Efficient Hardware Architecture of Recursive Karatsuba-Ofman Multiplier", 2008 International Conference on Design & Technology of Integrated Systems in Nanoscale Era

[8] www.mathworld.wolfram.com/KaratsubaMultiplication.html

[9] Abhijeet Kumar, Dilip Kumar, Siddhi, "Hardware Implementation of 16*16 bit Multiplier and Square using Vedic Mathematics", Design Engineer, CDAC, Mohali.

[10] M. Ramalatha, K. Deena Dayalan, P. Dharani, S. Deborah Priya, High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques, July 15-17, 2009 Zouk Mosbeh, Lebanon

[11] Neil H.E Weste, David Harris, Ayan Banerjee,"CMOS VLSI Design, A Circuits and Systems Perspective",Third Edition, Published by Person Education, PP-327-328]

[12] Himanshu Thapliyal and Hamid R. Arabnia, "A Time-Area- Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics", Department of Computer Science, The University of Georgia, 415 Graduate Studies Research Center Athens, Georgia 30602-7404, U.S.A.

[13] E. Abu-Shama, M. B. Maaz, M. A. Bayoumi, "A Fast and Low Power Multiplier Architecture", The Center for Advanced Computer Studies, The University of Southwestern Louisiana Lafayette, LA 70504.

[14] www.fpga-central.com

[15] "Spartan-3E FPGA Starter Kit Board User Guide", UG230 (v1.1) June 20, 2008

[16] www.en.wikipedia.com