RESEARCH ARTICLE OPEN ACCESS

Fpga Implementation of 8-Bit Vedic Multiplier by Using Complex Numbers

Gundlapalle Nandakishore, K.V.Rajendra Prasad
P.G.Student scholar M.Tech (VLSI) ECE Department Sree vidyanikethan engineering college (Autonomous).
Assistant professor ECE Department Sree vidyanikethan engineering college (Autonomous)

Abstract
The paper describes the implementation of 8-bit vedic multiplier using complex numbers previous technique describes that 8-bit vedic multiplier using barrel shifter by FPGA implementation comparing the both technique in this paper propagation delay is reduced so that processing of speed will be high 8-bit vedic multiplier using barrel shifter propagation delay nearly 22nsec but present technique 8-bit vedic multiplier using complex numbers where propagation delay is 19nsec. The design is implemented and verified by FPGA and ISE simulator. The core was implemented on the Spartan 3E starts board the preferred language is used in verilog.
Keywords— Barrel shifter, base selection module, Propagation delay, power index determinant.

I. INTRODUCTION

Arithmetic operations such as addition, subtraction and multiplication are deployed in various digital circuits to speed up the process of computation. Arithmetic logic unit is also implemented in various processor architectures like RISC [2], CISC etc., In general, arithmetic operations are performed using the packed-decimal format. This means that the fields are first converted to packed-decimal format prior to performing the arithmetic operation, and then converted back to their specified format (if necessary) prior to placing the result in the result field.

Vedic mathematics has proved to be the most robust technique for arithmetic operations. In contrast, conventional techniques for multiplication provide significant amount of delay in hardware implementation of n-bit multiplier. Moreover, the combinational delay of the design degrades the performance of the multiplier. Hardware-based multiplication mainly depends upon architecture selection in FPGA or ASIC**.** In this work we have put into effect a high speed Vedic multiplier using barrel shifter.

II. VEDIC SUTRAS

Vedic Sutras apply to and cover almost every branch of Mathematics. They apply even to complex problems involving a large number of mathematical operations. Application of the Sutras saves a lot of time and effort in solving the problems, compared to the formal methods presently in vogue. Though the solutions appear like magic, the application of the Sutras is perfectly logical and rational. The computation made on the computers follows, in a way, the principles underlying the Sutras. The Sutras provide not only methods of calculation, but also ways of thinking for their application.

Application of the Sutras improves the computational skills of the learners in a wide area of problems, ensuring both speed and accuracy, strictly based on rational and logical reasoning. Application of the Sutras to specific problems involves rational thinking, which, in the process, helps improve intuition that is the bottom - line of the mastery of the mathematical geniuses of the past and the present such as Aryabhatta, Bhaskaracharya, Srinivasa Ramanujan, etc.,

Multiplier implementation using FPGA has already been reported using different multiplier architectures but the performance of multiplier was improved in proposed design.

What we call “Vedic mathematics” is comprised of sixteen simple mathematical formulae from the Vedas [5].

1. Ekadhikena Purvena
2. Nikhilam navatascaramam Dasatah
3. Urdhva - tiryagbhyam
4. Paravartya Yojayet
5. Sunyam Samya Samuccaye
6. Anurupye - Sunyamanyat
7. Sankalana - Vyavakalanabhyam
8. Puranapuranabhyam
9. Calana - Kalanabhyam
10. Ekanyunena Purvena
11. Anurupyena
12. Adyamadyenantya - mantyena
13. Yavadunam Tavadunikrtya Varganca Yojayet
14. Antyayor Dasakepi
15. Antyayoreva
16. Gunita Samuccayah.

III. PROPOSED MULTIPLIERS URDHWA TIRYAKBHYAM SUTRA

As mentioned earlier, Vedic Mathematics can be divided into 16 different sutras to perform mathematical calculations. Among these the UrdhwaTiryakbhyam Sutra is one of the most highly preferred algorithms for performing multiplication. The algorithm is competent enough to be employed for the multiplication of integers as well as binary numbers. The term "Urdhwa Tiryakbhyam" originated from 2Sanskrit words Urdhwa and Tiryakbhyam which mean "vertically" and "crosswise" respectively [7]. The main advantage of utilizing this algorithm in comparison with the existing multiplication techniques, is the fact that it utilizes only logical "AND" operations, half adders and full adders to complete the multiplication operation. Also, the partial products required for multiplication are generated in parallel and a priori to the actual addition thus saving a lot of processing time.

Let us consider two 8 bit numbers X7-X0 and Y7-Y0,where 0 to 7 represent bits from the Least Significant Bit(LSB) to the Most Significant Bit (MSB). P0 to P15 represent each bit of the final computed product. It can be seen from equation (1) to (15), that P0 to P15 are calculated by adding partial products, which are calculated previously using the logical AND operation. The individual bits obtained from equations (1) to (15), in turn when concatenated produce the final product of multiplication which is depicted in (16).The carry bits generated during the calculation of the individual bits of the final product are represented from C1 to C30. The carry bits generated in (14) and (15) are ignored since they are superfluous.

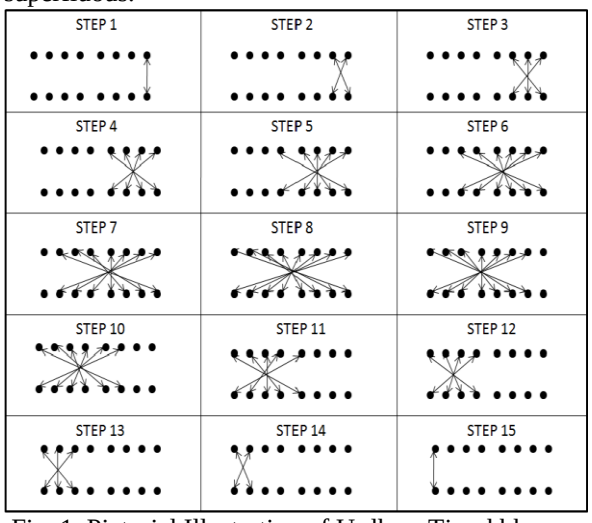

Fig. 1. Pictorial Illustration of Urdhwa Tiryakbhyam Sutra for multiplication of 2 eight bit numbers

The proposed Vedic multiplier is based on the "Urdhva Tiryagbhyam" sutra (algorithm). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, we apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware. It is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and Crosswise". It is based on a novel concept through which the generation of all partial products can be done with the concurrent addition of these partial products.

The algorithm can be generalized for n x n bit number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock frequency of the processor. Due to its regular structure, it can be easily layout in microprocessors and designers can easily circumvent these problems to avoid catastrophic device failures. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure. Due to its regular structure, it can be easily layout in a silicon chip.

The Multiplier based on this sutra has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other conventional multipliers.

Multiplication of two decimal numbers 252 x 846To illustrate this scheme, let us consider the multiplication of two decimal numbers 252 x 846 by Urdhva-Tiryakbhyam method as shown in Fig. 2. The digits on the both sides of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result and a carry. This carry is added in the next step and hence the process goes on. If more than one line are there in one step, all the results are added to the previous carry. In each step, least significant bit acts as the result bit and all other bits act as carry for the next step. Initially the carry is taken to be zero.

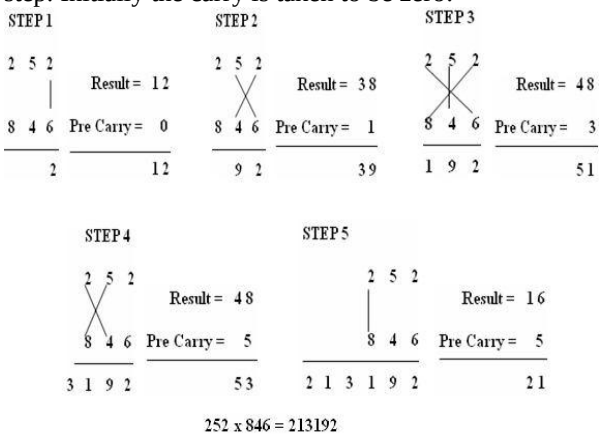

Fig. 2. Multiplication of two decimal numbers – 252 x 846

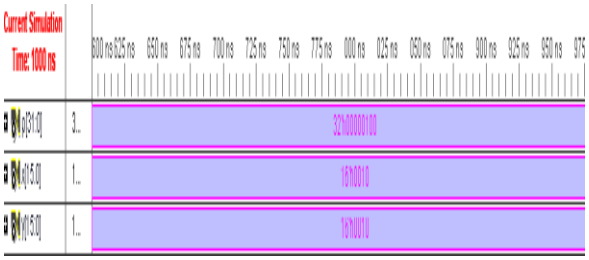

Fig. 3. Simulation results for timing wave forms.

IV. PROPOSDMULTIPLIERS

NIKHILAM SUTRA

Assume that the multiplier is 'X' and multiplicand is 'Y'. Though the designation of the numbers is different but the architecture implemented is same to some extent for evaluating both the numbers.

The mathematical expression for modified nikhilam sutra is given below.

$P=X*Y= (2^{K2})*(X+Z2*2^{(K1-K2)})+Z1*Z2.$

Where k1, k2 are the maximum power index of input numbers X and Y respectively.

Z1 AND Z2 ARE THE RESIDUES IN THE NUMBERS X AND Y RESPECTIVELY.

The hardware deployment of the above expression is partitioned into three blocks.

i. Base Selection Module
ii. Power index Determinant Module
iii. Multiplier.

The base selection module (BSM) is used to select the maximum base with respect to the input numbers. The second sub module power index determinant(PID) is used to extract the power index of k1 and k2. The multiplier comprises of base selection module (BSM), power index determinant (PID), sub tractor, barrel shifter, adder/sub tractor as sub-modules in the architecture.

A. Base selection module.

The base selection module has power index determinant (PID) as the sub-module along with barrel shifter, adder, average determinant, comparator and multiplexer.

An input 8-bit number is fed to power index determinant (PID) to interpret maximum power of number which is fed to barrel shifter and adder. The output of the barrel shifter is 'n' number of shifts with respect to the adder output and the input based to the shifter. Now, the outputs of the barrel shifter are given to the multiplexer with comparator input as a selection line. The outputs of the average determinant and the barrel shifter are fed to the comparator.

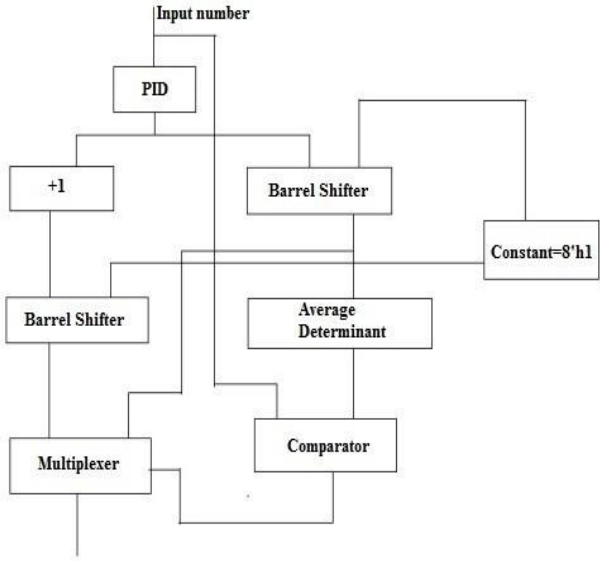

Fig.4. Base Selection Module, BSM

B. Power index determinant.

The input number is fed to the shifter which will shift the input bits by one clock cycle. The shifter pin is assigned to shifter to check whether the number is to be shifted or not. In this power index determinant (PID) the sequential searching has been employed to search for first '1' in the input number starting from MSB. If the search bit is '0' then the counter value will decrement up to the detection of input search bit is '1'. Now the output of the decrementer is the required power index of the input number.

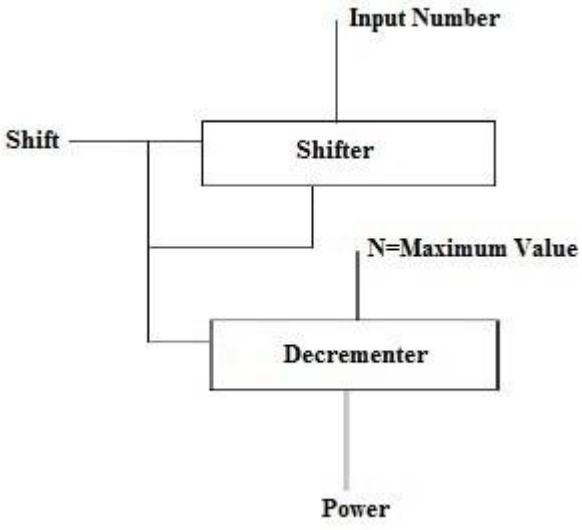

Fig.5. Power Index Determinant

C. Multiplier Architecture

The base selection module and the power index determinant form integral part of multiplier architecture. The architecture computes the

mathematical expression in equation1.Barrel shifter used in this architecture.

The two input numbers are fed to the base selection module from which the base is obtained. The outputs of base selection module (BSM) and the input numbers 'X' and 'Y' are fed to the sub tractors. The sub tractor blocks are required to extract the residual parts z1 and z2. The inputs to the power index determinant are from base selection module of respective input numbers.

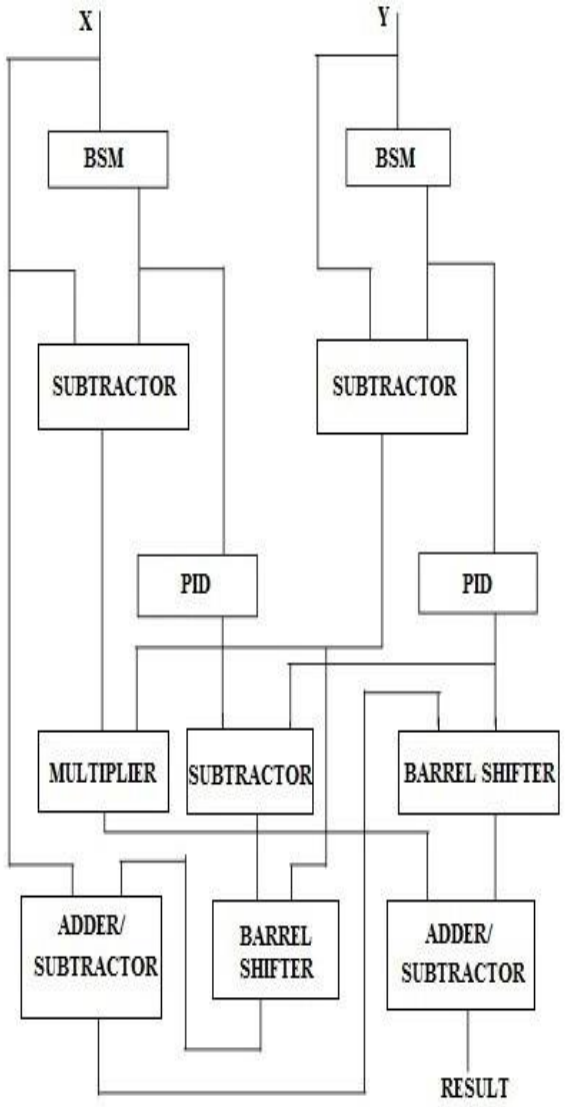

Fig.6. Multiplier Architecture

The Nikhilam sutra can also be applicable to binary number system. The compliment of multiplicand or multiplier is represented by taking 2"s compliment of that number. The right hand side part of the product is implemented using 16X16 bit multiplication. The left hand side part is implemented using 16-bit carry save adder. Hence the multiplication of two 16-bitnumbers is reduced to the multiplication of their compliments and addition.

The Block Diagram of Multiplier using Nikhilam Sutra is shown in Fig.7.The two inputs „a" and „b" represents 16-bit multiplier operand and 16-bit multiplicand operand respectively. The compliment of multiplicand or multiplier is represented by taking 2"s compliment of that number .So the 2"s complement block produces the complemented output of 16-bit multiplier operand (- a) and 16-bit multiplicand operand ( ). The complemented output of 16-bit multiplier operand and 16-bitmultiplicand operand are two inputs to multiplier. The number of bits required in the right hand side of the product should have16- bits irrespective of the number of bits in the product of compliments. So the surplus 16-bits of the right hand side part of the product is fed to one of the input of carry save adder for left hand part of the result. The left hand side part is implemented using 16-bit carry save adder. The negative of complemented multiplicand is implemented by taking output of the 2"scomplement block in the left hand side. The three 16-bits input operands of carry saver are 16-bit multiplier, the negative of complemented multiplicand and the surplus 16-bits of the right hand side part of the product. The two output of carry save adder i.e. sum vector and carry vector are inputs to binary adder block. The output of the adder represents left hand side of the answer.

The Nikhilam Sutra literally means "all from 9 and last from10". It is more efficient when the numbers involved are large. The Nikhilam Sutra algorithm is efficient for multiplication only when the magnitudes of both operands are more than half their maximum values. For n-bit numbers, therefore both operands must be larger than 2n-1. Nikhilam Sutra is explained by considering the multiplication of two single digit decimal numbers 8 and 7 where the chosen base is 10 which is nearest to and greater than both these two numbers.

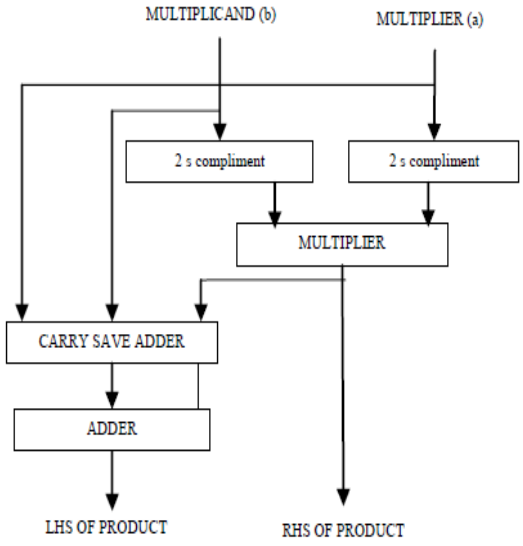

Fig.7. The Architecture of proposed Vedic Multiplier using Nikhilam Sutra.

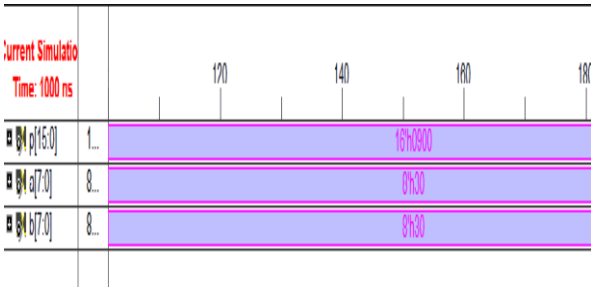

Fig. 8. Simulation results for timing wave forms.

V. EXTENSION METHOD COMPLEX NUMBERS

Multiplication is the process of adding a number of partial products. Multiplication algorithms differ in terms of partial product generation and partial product addition to produce the final result[8].Higher throughput arithmetic operations are important to achieve the desired performance in many real time signal and image processing applications. With time applications, many researchers have tried to design multipliers which offer either of the following- low power consumption, high speed, regularity of layout and hence less area or even grouping of them in multiplier.

Complex number arithmetic computation is a key arithmetic feature in modem digital communication and optical systems. Many algorithms based on convolutions, correlations, and complex filters require complex number multiplication, complex number division, and high-speed inner-products. Among these computations, complex number multipliers and complex number inner-products are becoming more and more demanded in modem digital communication, modem optical systems, and radar systems.

Multiplication is an essential operation for high speed hardware implementation of complex number computation. To compute the product of two complex numbers, the conventional method is to use four binary multiplications, one addition, and one subtraction.

$$A = A,+ JA, \text{ AND } B=B,+ JB,$$

Multiplication of A and B is given by

$$AXE = ARB, - A,B, + j(A,B, +A\$,)$$

The paper is organized as follows. In section 2, Vedic multiplication method based on Urdhava Tiryakbhyam sutra is discussed. Section 3 deals with the design of the above said multiplier. Section 4summarizes the experimental results obtained, while section 5 presents the conclusions of the work.

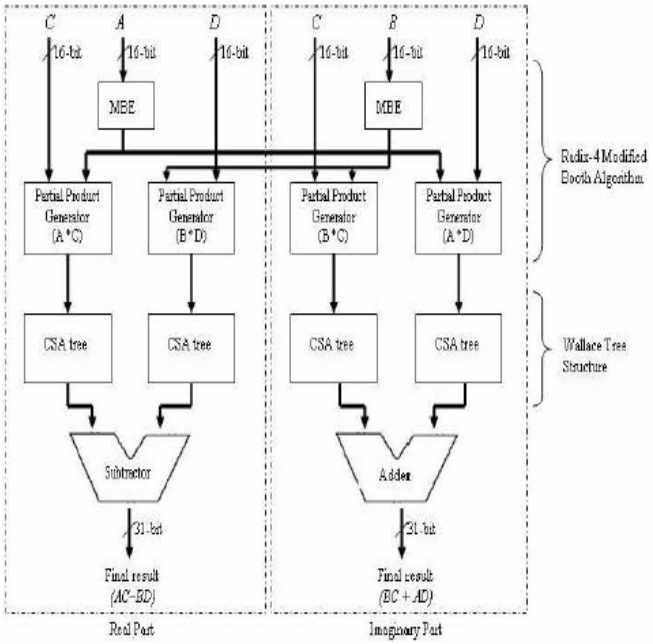

Figure 9. Block diagram of complex number multiplier.

While implementing complex number multiplication, the multiplication system can be divided into two main components giving the two separate results known as real part (R) and imaginary part (I).

$$R + j\,I = (A + j\,B)\,(C + j\,D)$$

Gauss's algorithm for complex number multiplication gives two separate equations to calculate real and imaginary part of the final result. From equation (1) the real part of the output can be given by (AC - BD), and the imaginary part of the result can be computed using (BC + AD). Thus four separate multiplications and are required to produce the real as well as imaginary part numbers.

Multiplication process is the critical part for any complex number multiplier design. There are three major steps involved for multiplication. Partial products are generated in first step. In second step partial product reduction to one row of final sums and carries is done .Third and final stage adds the final sums and carries to give the result. Since the Radix 4 modified Booth algorithm miscapable to reduce the number of partial products by nearly equal to half, this algorithm is used in the first step of implementation of complex number multiplication using Booth's method. In the second step Wallace tree structure is used to rapidly reduce the partial product rows to the final two rows giving sums and carries. The multiplication design based onRadix-4

modified Booth algorithm consists of two main blocks known as MBE (Modified Booth Encoding) and partial product generator as shown in Fig. 2. Wallace Tree CSA structures have been used to sum the partial products in reduced time. In this regard, combining both algorithms in one multiplier, we can expect a significant reduction in computing multiplications.

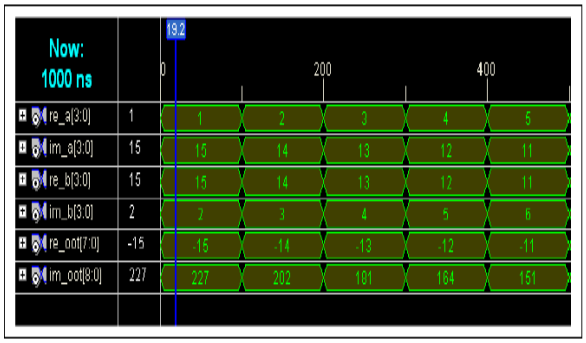

FIG. 10. SIMULATION RESULTS FOR TIMING WAVE FORMS.

Comparison Table.

Types of TDC /parameter	**Vedic multiplier**	**Complex numbers**
Number of LUTs	1920	1900
Delay(ns)	21.679	19.536

VI. CONCLUSION

The design have been made to reduced the propagation delay With 45% compared to existing multiplier. The propagation delay for proposed 8-bit vedic multiplier is found to be 21.679ns. There complex numbers is much more efficient than compared with vedic multiplier and execution time will be speed.

REFERENCES

[1] Prabir Saha, Arindam Banerjee, Partha Bhattacharyya, Anup Dandapat, "*High speed ASIC design of complex multiplier using vedic mathematics*" , Proceeding of the 2011 IEEE Students' Technology Symposium 14-16 January, 2011, lIT Kharagpur, pp. 237-241.

[2] Shamsiah Suhaili and Othman Sidek, "*Design and implementation of reconfigurable alu on FPGA*", **3**rd International Conference on Electrical & Computer Engineering ICECE 2004, 28-30 December 2004, Dhaka, Bangladesh, pp.56-59.

[3] Sumit Vaidya and Deepak Dandekar, "*Delay-power performance comparison of multipliers in vlsi circuit design*", International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010, pp.47-56.

[4] P. Mehta, and D. Gawali, "*Conventional versus Vedic mathematical method for Hardware implementation of a multiplier*," in Proceedings IEEE International Conference on Advances in Computing, Control, and Telecommunication Technologies, Trivandrum, Kerala, Dec. 28-29, 2009, pp. 640-642.

[5] J. S. S. B. K. T. Maharaja, Vedic mathematics, Delhi: Motilal Banarsidass Publishers Pvt Ltd,(2010).

[6] Prabir Saha, Arindam Banerjee, Partha Bhattacharyya, Anup Dandapat, "*High speed ASIC design of complex multiplier using vedic mathematics*" , Proceeding of the 2011 IEEE Students' Technology Symposium 14-16 January, 2011, lIT Kharagpur, pp. 237-241.

[7] Shamsiah Suhaili and Othman Sidek, "*Speed Comparison of 16x16 Vedic Multipliers* ICECE 2004, 28-30 December 2004, Dhaka, Bangladesh, pp.56-59.

[8] Sumit Vaidya and Deepak Dandekar, "*Delay-power performance comparison of multipliers in vlsi circuit design*", International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010, pp.47-56.

[9] P. Mehta, and D. Gawali, "*Conventional versus Vedic mathematical method for Hardware implementation of a multiplier*," in Proceedings IEEE International Conference on Advances in Computing, Control, and Telecommunication Technologies, Trivandrum, Kerala, Dec. 28-29, 2009, pp. 640-642.

AUTHORS BIOGRAPHY

1.K.V.Rajendra Prasad working as Assistant Professor in sree Vidyanikethan Engineering College, Tirupati. He Completed M.Tech from sree Vidyanikethan Engineering College and completed B. Tech From K.S.R.M. College of Engineering, Kadapa. His interested areas are Low Power VLSI and FPGA.

2.G.Nandakishore P.G.Student scholar M.Tech (VLSI) ECE Department Sree vidyanikethan engineering college (Autonomous)