

Area And Speed Wise Superior Multiply And Accumulate Unit Based On Vedic Multiplier

Mr. Virendra Babanrao Magar*

*(Lord Krishna College of Technology, Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Bhopal)

ABSTRACT

The speed of a multiplier is very important to any Digital Signal Processor (DSPs). This paper presents the efficiency of Urdhva Triyagbhyam Vedic method for multiplication, which strikes a difference in the actual process of multiplication itself. Vedic Mathematics is the earliest method of Indian mathematics which has a unique technique of calculations based on 16 Formulae. In this paper, a high performance, high throughput and area efficient architecture of a MAC architecture using Vedic multiplier for the Field Programmable Gate Array (FPGAs) is proposed. This paper presents 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 partial products and eliminates unwanted multiplication steps. Multiplier architecture is based on generating all partial products and their sums in one step. Chipscope VIO is used to give random inputs of desired values by user, on which proposed Vedic multiplication is performed. The proposed algorithm is modeled using VHDL i.e. Very High Speed integrated circuit hardware description language. The propagation time of the proposed architecture is found quiet less. The Xilinx Chipscope VIO generator allows us to give the runtime inputs. The Xilinx Chipscope tool will be used to test the FPGA inside results while the logic running on FPGA. The Xilinx Spartan 3 Family FPGA development board will be used for this circuit. The proposed multiplier implemented using Vedic multiplication is efficient and competent in terms of area and speed compared to its implementation using Array and Booth multiplier architectures. The results clearly indicate that Urdhava Tiryakbhyam can have a great impact on improving the speed of Digital Signal Processors.

Keywords – Chipscope, Low Power, Latency, Urdhva Tiryakbhyam, Vedic Multiplier, VHDL.

I. INTRODUCTION

The requirement for high speed processing has been increased because of newer computer applications and to achieve the desired performance in many real time signal and image processing applications, higher throughput arithmetic operations are important. Instead of having more consuming processing time system, we have proposed Urdhva Tiryagbhyam Vedic method for arithmetic operations which perform a large no of mathematical calculations in a very less time. It increases the overall speed of the different electronics devices.

Digital multipliers are the center components of all the digital signal processors (DSPs) and the speed of the DSP is mostly determined by the speed of its multipliers. They are essential in the implementation of computation systems like Fast Fourier transforms (FFTs) and multiply accumulate (MAC). Array multiplication algorithm and Booth multiplication algorithm are most commonly multiplication algorithms implemented in the digital hardware. The computation time in array multiplier is comparatively less because it calculate execution of partial products independently by using parallelism. The delay associated with the array multiplier is the time taken by the signals to propagate through the gates that form the multiplication array. Since it

consume low-power and have comparatively good performance.

Booth Multiplier is another standard approaches and multiplication algorithm to have hardware implementation of binary multiplier for VLSI implementation at low power. Large booth arrays are required for high speed multiplication and exponential operations which need huge partial sum and partial carry registers. It requires around $n = (2m)$ clock cycles to create the least significant half of the final product for multiplication of two n -bit operands using a radix-4 booth recording multiplier, where m is the number of Booth recorder adder stages. Hence, a large propagation delay is associated with Booth Multiplier. Due to the importance of digital multipliers in DSP, it has always been an active area of research and a number of interesting multiplication algorithms have been reported in the literature. In this paper, we have proposed one such new multiplication algorithm which avoids the need of large multipliers by reducing the large number to the smaller number multiplication's count which reduces the propagation delay linked with the conventional large multipliers significantly.

The structure of the proposed algorithm is based on the Urdhava Tiryakbhyam Sutra (formula) of Vedic mathematics which is simply means: "vertical and crosswise multiplication". The procedure of

multiplication using the Urdhva Tiryakbhyam involves minimum calculations, which in turn will lead to reduced number of steps in computation, reducing the space, saving more time for computation. Hence it optimizes to take full advantage of reduction in the number of bits in multiplication. Although Urdhva Tiryakbhyam is applicable to all cases of multiplication, it is more efficient when the numbers involved are large. The proposed multiply and accumulate unit, consist of Vedic multiplier, adder and buffer. The buffer stores result upon receiving enable clock signal instead of storing result based on each clock signal.

II. RELATED WORKS

There have been several efforts for Urdhva Triyagbhyam Vedic method for multiplication. Ramesh Pushpangadan, Vineet Sukumaran has proposed methodology by using Urdhva Triyagbhyam Vedic method which having main advantages is delay increases slowly as input bit increases. V Jayaprakasan, S Vijayakumar, V S Kanchana Bhaskaran [1] has proposed methodology of A 4x4 multiplier based on the Vedic and Conventional methods have been designed using SPICE simulator. Simulation results depict the Vedic design incurring 29% of reduced average power.

Sandesh S. Saokar, R. M. Banakar, Saroja Siddamal [2] proposed fast multiplier architecture for signed Q-format multiplications using Urdhva Tiryakbhyam method of Vedic mathematics. Since Q-format representation is widely used in Digital Signal Processors the proposed multiplier can substantially speed up the multiplication operation which is the basic hardware block. They occupy less area and are faster than the booth multipliers. But it has not introduced pipeline stages in the multiplier architecture for maximizing throughput. We have gone through different existing research on multiplier for less power consumption and time efficiency. But many lacunas present in this methodology. Here we have proposed our methodology which consumes less power; less area as well as less time since it optimizes the overall performance of the system.

III. WALLACETREE MULTIPLIER

In 1964, Australian Computer Scientist Chris Wallace has developed Wallace tree which is an efficient hardware implementation of a digital circuit that multiplies two integers. A fast process for multiplication of two numbers was developed by Wallace [3]. Using this method, a three step process is used to multiply two numbers; the bit products are formed, the bit product matrix is reduced to a two row matrix where sum of the row equals the sum of bit products, and the two resulting rows are summed with a fast adder to produce a final product. Three bit signals are passed to a one bit full adder ("3W") which is called a three input Wallace tree circuit and the output of sum signal is supplied to the next stage full

adder of the same bit. The carry output signal is passed to the next stage full adder of the same no of bit, and the carry output signal thereof is supplied to the next stage of the full adder located at a one bit higher position. Wallace tree is a tree of carry-save adders (CSA) arranged as shown in Fig. 1. A carry save adder consists of full adders like the more familiar ripple adders, but the carry output from each bit is brought out to form second result vector rather being than wired to the next most significant bit.

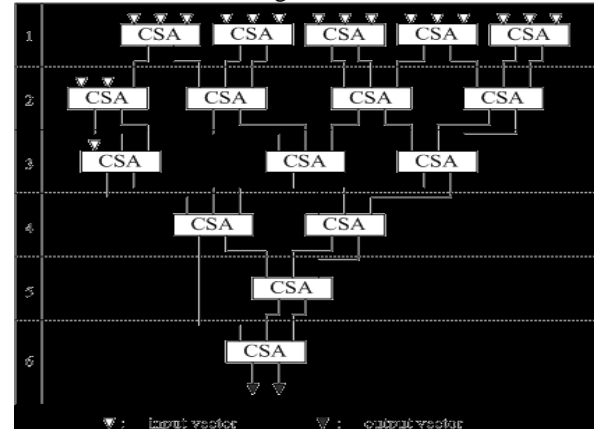


Fig.1. Wallace tree of carry-save adders

The carry vector is 'saved' to be combined with the sum later. In the Wallace tree method, the circuit layout is not easy although the speed of the operation is high since the circuit is quite irregular. Wallace tree is known for their optimal computation time, when adding multiple operands to two outputs using carry-save adders. The Wallace tree guarantees the lowest overall delay but requires the largest number of wiring tracks. The number of wiring tracks is a measure of wiring complexity.

IV. BOOTH MULTIPLIER

Another improvement in the multiplier is by reducing the number of partial products generated. Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation. The algorithm was invented by Andrew Donald Booth in 1950[4]. The Booth recording multiplier is such multiplier which scans the three bits at a time to reduce the number of partial products. These three bits are: the two bit from the present pair; and a third bit from the high order bit of an adjacent lower order pair. After examining each triplet of bits, the triplets are converted by Booth logic into a set of five control signals used by the adder cells in the array to control the operations performed by the adder cells. To speed up the multiplication Booth encoding performs several steps of multiplication at once. Booth's algorithm takes advantage of the fact that an adder, subtractor is nearly as fast and small as a simple adder. If 3 consecutive bits are same then addition/subtraction operation can be skipped. Thus in most of the cases the delay associated with Booth Multiplication are smaller than that with Array

Multiplier. However the performance of Booth Multiplier for delay is input data dependant. In the worst case the delay with booth multiplier is on par with Array Multiplier. The method of Booth recording reduces the numbers of adders and hence the delay required to produce the partial sums by examining three bits at a time. The high performance of booth multiplier comes with the drawback of power consumption. The reason is large number of adder cells required that consumes large power.

V. VEDIC MATHEMATICS

Vedic Mathematics hails from the ancient Indian scriptures called “Vedas” or the source of knowledge. This system of computation covers all forms of mathematics, be it geometry, trigonometry or algebra. The prominent feature of Vedic Mathematics is the rationality in its algorithms which are designed to work naturally. This makes it the easiest and fastest way to perform any mathematical calculation mentally. Vedic Mathematics is believed to be created around 1500 BC and was rediscovered between 1911 to 1918 by Sri Bharti Krishna Tirthaji (1884-1960) who was a Sanskrit scholar, mathematician and a philosopher [5]. He organized and classified the whole of Vedic Mathematics into 16 formulae or also called as *sutras*. These formulae form the backbone of Vedic mathematics. Great amount of research has been done all these years to implement algorithms of Vedic mathematics on digital processors. It has been observed that due to coherence and symmetry in these algorithms it can have a regular silicon layout and consume less area along with lower power consumption.

VI. URDHAVA TIRYAKBHYAM METHOD

Urdhava Tiryakbhyam [6] is a Sanskrit word which means vertically and crosswire in English. The method is a general multiplication formula applicable to all cases of multiplication. It is based on a novel concept through which all partial products are generated concurrently. Fig. 2 demonstrates a 4x4 binary multiplication using this method. The method can be generalized for any N x N bit multiplication. This type of multiplier is independent of the clock frequency of the processor because the partial products and their sums are calculated in parallel. The net advantage is that it reduces the need of microprocessors to operate at increasingly higher clock frequencies. As the operating frequency of a processor increases the number of switching instances also increases. This results in more power consumption and also dissipation in the form of heat which results in higher device operating temperatures. Another advantage of Urdhva Tiryakbhyam multiplier is its scalability. The processing power can easily be increased by increasing the input and output data bus widths since it has a regular structure. Due to its regular structure, it can be easily layout in a silicon

chip and also consumes optimum area. As the number of input bits increase, gate delay and area increase very slowly as compared to other multipliers. Therefore Urdhava Tiryakbhyam multiplier is time, space and power efficient. The line diagram in fig. 2 illustrates the algorithm for multiplying two 4-bit binary numbers a3a2a1a0 and b3b2b1b0. The procedure is divided into 7 steps and each step generates partial products. Initially as shown in step 1 of fig. 2, the least significant bit (LSB) of the multiplier is multiplied with least significant bit of the multiplicand (vertical multiplication). This result forms the LSB of the product. In step 2 next higher bit of the multiplier is multiplied with the LSB of the multiplicand and the LSB of the multiplier is multiplied with the next higher bit of the multiplicand (crosswire multiplication). These two partial products are added and the LSB of the sum is the next higher bit of the final product and the remaining bits are carried to the next step. For example, if in some intermediate step, we get the result

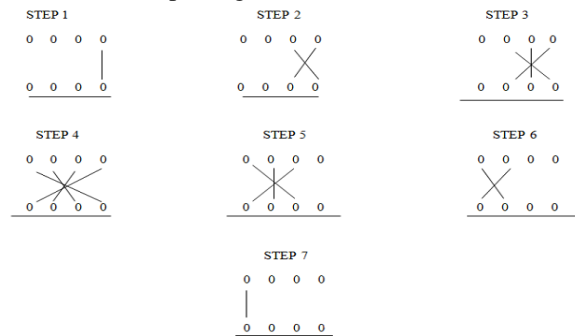


Fig.2. Line diagram of the multiplication of two 4 bit numbers using Urdhava Tiryakbhyam method.

Bit (referred as r_n) and 110 as the carry (referred as c_n). Therefore c_n may be a multi-bit number. Similarly other steps are carried out as indicated by the line diagram. The important feature is that all the partial products and their sums for every step can be calculated in parallel.

VII. THE MULTIPLIER ARCHITECTURE

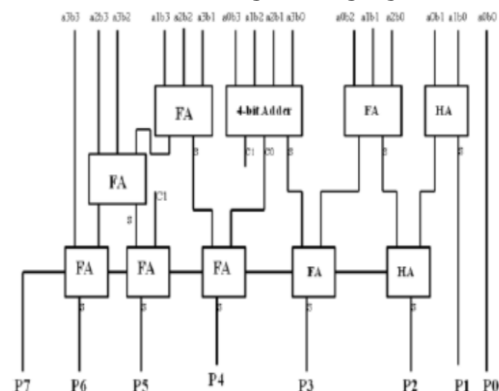


Fig.3. 4X4 Vedic multiplier structure For inputs a3 a2 a1 a0 and b3 b2 b1 b0

$$\begin{aligned}
 P0 &= a0b0 \\
 COP1 &= a0b1 + a1b0 \\
 C1P2 &= C0 + a0b2 + a2b0 + a1b1 \\
 C2P3 &= C1 + a3b0 + a0b3 + a1b2 + a2b1 + C00 \\
 C3P4 &= C2 + a3b1 + a1b3 + a2b2 + C01 + C10 \\
 C4P5 &= C3 + a3b2 + a2b3 + C11 + C20 \\
 C5P6 &= C4 + a3b3 + C21
 \end{aligned}$$

The 8x8 Vedic multiplier module is implemented using four 4x4 Vedic multiplier modules as shown in Fig.3. Here partial product generation and addition is done concurrently. $b7b6b5b4b3b2b1b0$ and $a7a6a5a4a3a2a1a0$ are taken as two binary numbers. 4x4 Vedic multiplier modules, three 8 bit ripple carry adder are used to generate the desired 16 bit product $s15$ down to $s0$. The least significant 4 bits of the result of rightmost 4x4 Vedic multiplier produce the result $s3s2s1s0$. The 8 bit ripple carry adder (located in middle in Fig.4.) adds two 8 bits operands i.e. concatenated 8 bits ("0000" and most The upper 8 bit ripple carry adder adds the results of two 4x4 Vedic multiplier modules (second and third from right) and generates one carry and 8 bit result. The bottom 8 bit ripple carry adder adds 4x4 Vedic multiplier module result and concatenated 8 bits ("000", carry from upper 8 bit ripple carry adder and most significant bits of the result from middle 8 bit ripple carry adder) to generate the most significant bits of the final product i.e. $s15s14s13s12s11s10s9s8$. Significant four bits of rightmost 4x4 Vedic. It generates the resultant bits $s7s6s5s4$ at its output. multiplier module) and the result of second from right 4x4 Vedic multiplier module).

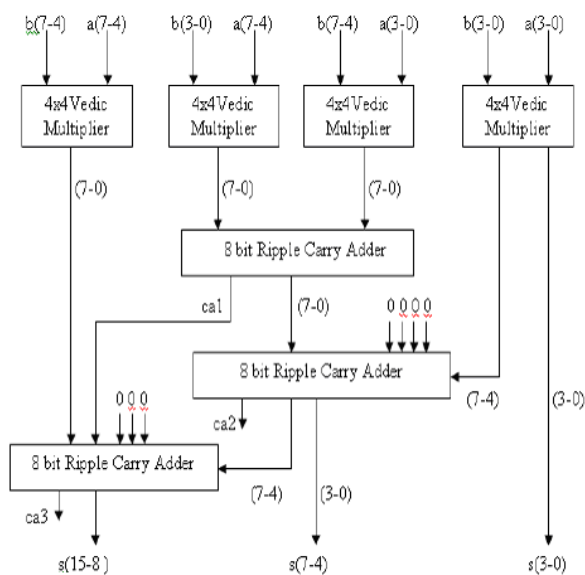


Fig.4. Block Diagram of 8 X 8 bit Vedic Multiplier

VIII. FPGA ARCHITECTURE

Field-Programmable Gate Arrays (FPGAs) have become one of the key digital circuit implementation media over the last decade. FPGA architecture has a dramatic effect on the quality of the final that can be electrically programmed to become

FPGA's, 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(as shown in Fig.5.). The array is surrounded by programmable input/output blocks. 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 behavior of the pre-fabricated chip after fabrication, in the "field," where system users create designs.

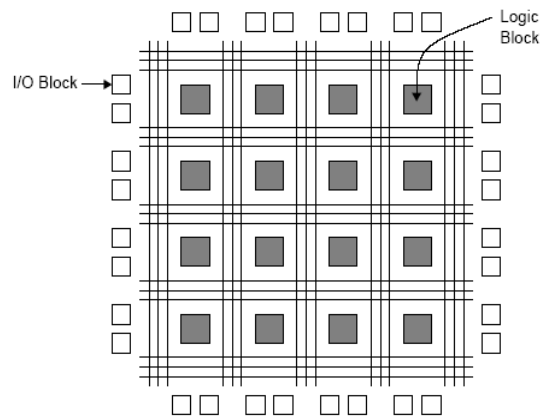


Fig.5. Structure of an FPGA.

IX. CONSTRUCTION OF MAC UNIT

A basic MAC architecture consist of a multiplier and an accumulate adder organized as in Fig. 6. The MAC unit computes the product of two numbers and adds the product to an accumulator register. The output of the register is feedback to one input of the adder as shown in Fig. 6.

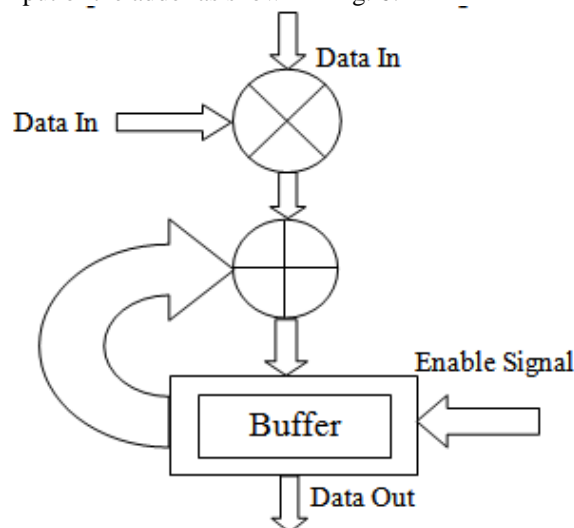


Fig.6. MAC Architecture.

On each clock, the output of the multiplier is added to the register[7]. The proposed Multiply & Accumulate Unit architecture has a enable signal which enables buffer register only on rising edge of clock. The combinational multipliers require a large

amount of logic, but can compute a product much more quickly than the conventional method of shifting and adding.

X. VERIFICATION AND IMPLEMENTATION

The algorithm is implemented in VHDL and logic synthesis is done in Modelsim simulator, whereas the synthesis is done in Xilinx Project navigator[8]. The chipscope tool gives the hardware verification of results as compared to simulation results. Xilinx Spartan 3 family FPGA development board is used for this work. Simulation result and Technology view of 8 bit Vedic multiplier using Vedic Mathematics are shown in Fig.7 and Fig.8. Chipscope VIO result are shown in Fig.8 and Fig.10. The simulation result and technology view of MAC architecture using 8 X 8 vedic multiplier is shown in Fig.11 & Fig.12.

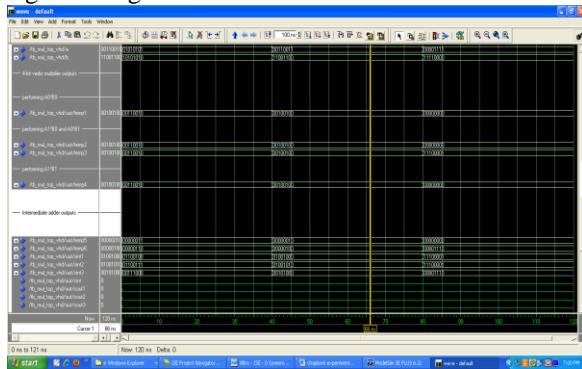


Fig.7. Simulation result for 8 bit Vedic multiplier using Vedic Mathematics

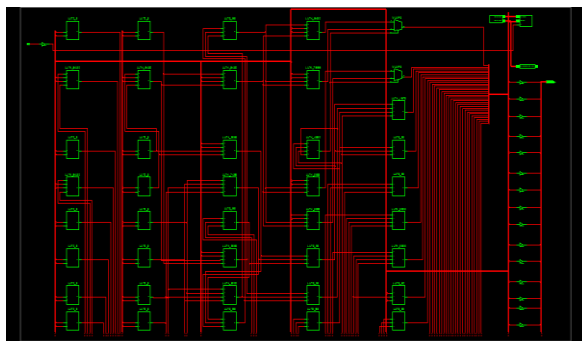


Fig.8. Technology view of 8 bit multiplier using Vedic Mathematics

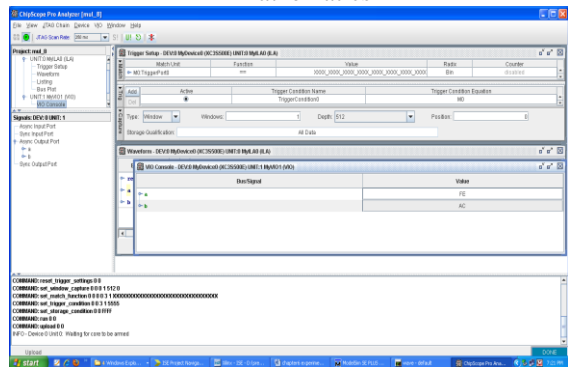


Fig.9. Data inputs given to Vedic multiplier using Chipscope

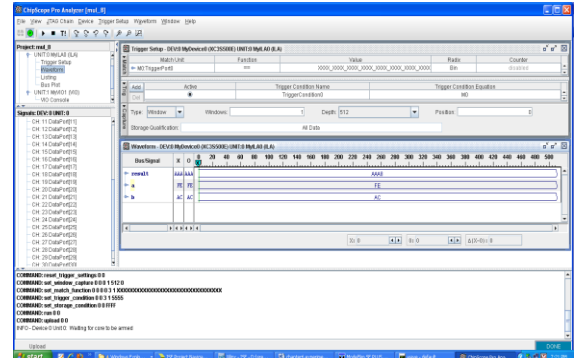


Fig.10. Data output of Vedic multiplier with inputs

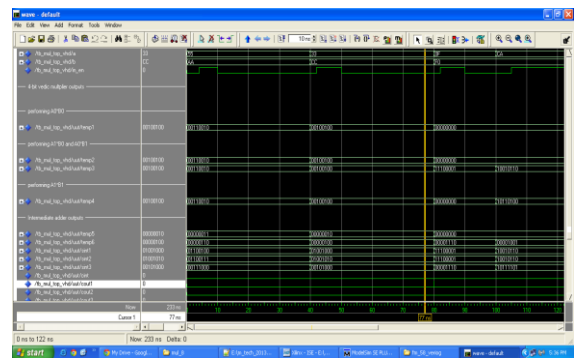


Fig.11. MAC output using Vedic multiplier

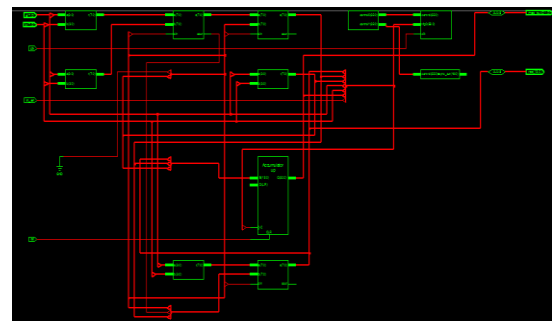


Fig.12. Technology view of 8 bit multiplier using MAC Architecture using Vedic Mathematics.

Table.1.Comparison of Combinational Delay (ns)

Device:	Modifie d Booth Wallace Multiplier[9]	Ramesh Pushpan gadam[9]	My
SPARTAN3:XC3S50:-4			
8X8	25.756	25.175	20.499

XI. RESULT AND DISCUSSION

The combination path delay found is 20.499ns with speed grade -4. The comparison of combinational delay is given in Table.1. The MAC delay obtained with this architecture on SPARTAN3:XC3S50:-4 is 22.240ns. Vedic multiplier attains high speeds because 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. (Since the partial products and their sums are calculated in parallel, the multiplier is

independent of the clock frequency of the processor. Thus the multiplier will require the same amount of time to calculate the product and hence is independent of the clock frequency.).

XII. CONCLUSION

It can be concluded that Vedic multiplier and square is faster than array multiplier and Booth multiplier. The MAC architecture using Vedic Multiplier with enable signal stores data in buffer upon receiving enable clock. The speed improvements are gained by parallelizing the generation of partial products with their concurrent summations. The Vedic Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Regular structure makes the implementation of any higher order multiplier (NXN) with basic multiplier structure. It is seen that this design is quite efficient in terms of silicon area/speed. Such a design should enable substantial savings of resources in the FPGA when used for image/video processing applications.

REFERENCES

- [1] V. Jayaprakasan, S. Vijayakumar, V S KanchanaBhaaskaran, "Evaluation of the Conventional vs. Ancient Computation methodology for Energy Efficient Arithmetic Architecture".*International Conference on Process Automation, Control and Computing (PACC)*, 2011978-1-61284-764-1/11/\$26.00 ©2011 IEEE
- [2] Sandesh S. Saokar, R. M. Banakar, Saroja Siddamal, "High Speed Signed Multiplier for Digital Signal Processing Applications" 2012 IEEE. *IEEE International Conference on Signal Processing, Computing and Control (ISPCC)*, 2012
- [3] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans.Electronic Comput.*, vol. EC-\3, pp. 14-17, Dec. 1964A.
- [4] D. Booth, "A signed binary multiplication technique," *Q. J. Mech. Appl. Math.*, vol. 4, pp. 236-240, 1951.
- [5] Jagadguru Swami Sri Bharati Krisna Tirthaji Maharaja, "Vedic Mathematics: Sixteen Simple Mathematical Formulae from the Veda," Motilal Banarasidas Publishers, Delhi, 2009, pp. 5-45.
- [6] H. Thapliyal and M. B. Shrinivas and H. Arbania, "Design and Analysis of a VLSI Based High Performance Low Power Parallel Square Architecture," *Int. Conf. Algo. Math. Comp. Sc.*, Las Vegas, June 2005, pp. 72-76
- [7] Devika Jaina, Kabiraj Sethi, and Rutuparna Panda, "Vedic Mathematics based Multiply Accumulate Unit", 2011 *International conference on Computational Intelligence and Communication Systems*, Pages 754-757.
- [8] 'Xilinx ISE User manual', Xilinx Inc, USA, 2007.
- [9] R. Pushpangadan, V. Sukumaran, R.Innocent, D.Sasikumar, and V.Sunder, "High Speed Vedic Multiplier for Digital Signal Processors", *IETE Journal of Research*, vol.55, pp.282-286, 2009.