



A Time-Area-Power Efficient High Speed Vedic Mathematics Multiplier using Compressors

Kishan.P
M.Tech Scholar (VLSI)
Dept. of ECE

Ashoka Institute of Engineering & Technology

G. Sai Kumar
Assitant. Professor
Dept. of ECE

Ashoka Institute of Engineering & Technology

Abstract: With the advent of new technology in the fields of VLSI and communication, there is also an ever growing demand for high speed processing and low area design. It is also a well known fact that the multiplier unit forms an integral part of processor design. Due to this regard, high speed multiplier architectures become the need of the day. In this paper, we introduce a novel architecture to perform high speed multiplication using ancient Vedic maths techniques. A new high speed approach utilizing 4:2 compressors and novel 7:2 compressors for addition has also been incorporated in the same and has been explored. Upon comparison, the compressor based multiplier introduced in this paper, is almost two times faster than the popular methods of multiplication. With regards to area, a 1% reduction is seen. The design and experiments were carried out on a Xilinx Spartan 3e series of FPGA and the timing and area of the design, on the same have been calculated.

Keywords—4:2 Compressor, 7:2 Compressor, Booth's multiplier, high speed multiplier, modified Booth's multiplier, Urdhwa Tiryakbhyam Sutra, Vedic Mathematics.

1. INTRODUCTION

The challenge of the verifying a large design is growing exponentially. There is a need to define new methods that makes functional verification easy. Several strategies in the recent years have been proposed to achieve good functional verification with less effort. Recent advancement towards this goal is methodologies. The methodology defines a skeleton over which one can add flesh and skin to their requirements to achieve functional verification.

Complex multiplication is of immense importance in Digital Signal Processing (DSP) and Image Processing (IP). To implement the hardware module of Discrete Fourier Transformation (DFT), Discrete Cosine Transformation (DCT), Discrete Sine Transformation (DST) and modem broadband communications; large numbers of complex multipliers are required. Complex number multiplication is performed using four real number multiplications and two additions/ subtractions. In real number processing, carry needs to be propagated from the least significant bit (LSB) to the most significant bit (MSB) when binary partial products are added. Therefore, the addition and subtraction after binary multiplications limit the overall speed. Many alternative method had so far been proposed for complex number multiplication like algebraic transformation based implementation, bit-serial multiplication using offset binary and distributed arithmetic, the CORDIC (coordinate rotation

digital computer) algorithm, the quadratic residue number system (QRNS), and recently, the redundant complex number system (RCNS). Blahut et. al proposed a technique for complex number multiplication, where the algebraic transformation was used. This algebraic transformation saves one real multiplication, at the expense of three additions as compared to the direct method implementation. A left to right array for the fast multiplication has been reported in 2005, and the method is not further extended for complex multiplication. But, all the above techniques require either large overhead for pre/post processing or long latency. Further many design issues like as speed, accuracy, design overhead, power consumption etc., should not be addressed for fast multiplication. In algorithmic and structural levels, a lot of multiplication techniques had been developed to enhance the efficiency of the multiplier; which encounters the reduction of the partial products and/or the methods for their partial products addition, but the principle behind multiplication was same in all cases. Vedic Mathematics is the ancient system of Indian mathematics which has a unique technique of calculations based on 16 Sutras (Formulae). "Urdhva-tiryakbyham" is a Sanskrit word means vertically and crosswise formula is used for smaller number multiplication. "Nikhilam Navatascaramam Dasatah" also a Sanskrit term indicating "all from 9 and last from 10", formula is used for large number multiplication and subtraction. All these formulas are adopted from ancient Indian Vedic Mathematics. In this work we formulate this mathematics for designing the complex multiplier architecture in transistor level with two clear goals in mind such as: i) Simplicity and modularity multiplications for VLSI implementations and ii) The elimination of carry propagation for rapid additions and subtractions. Mehta et al. have been proposed a multiplier design using "Urdhva-tiryakbyham" sutras, which was adopted from the Vedas. The formulation using this sutra is similar to the modem array multiplication, which also indicating the carry propagation issues. A multiplier design using "Nikhilam Navatascaramam Dasatah" sutras has been reported by Tiwari et. al in 2009, but he has not implemented the hardware module for multiplication.

Multiplier implementation in the gate level (FPGA) using Vedic Mathematics has already been reported but to the best of our knowledge till date there is no report on transistor level (ASIC) implementation of such complex multiplier. By employing the Vedic mathematics, an N bit complex number



multiplication was transformed into four multiplications for real and imaginary terms of the final product. "Nikhilam Navatascaramam Dasatah" sutra is used for the multiplication purpose, with less number of partial products generation, in comparison with array based multiplication. When compared with existing methods such as the direct method or the strength reduction technique, our approach resulted not only in simplified arithmetic operations, but also in a regular arraylike structure. The multiplier is fully parameterized, so any configuration of input and output word-lengths could be elaborated. Transistor level implementation for performance parameters such as propagation delay, dynamic leakage power and dynamic switching power consumption calculation of the proposed method was calculated by spice spectre using 90 nm standard CMOS technology and compared with the other design like distributed arithmetic, parallel adder based implementation and algebraic transformation based implementation. The calculated results revealed (16,16)x(16,16) complex multiplier have propagation delay only 4 ns with 6.5 mW dynamic switching power.

2. ALGORITHMS OF VEDIC MATHEMATICS

2.1. VEDIC MULTIPLICATION: The proposed Vedic multiplier is based on the Vedic multiplication formulae (Sutras). 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. Vedic multiplication based on some algorithms, some are discussed below:

2.1.1. Urdhva Tiryakbhyam sutra: The multiplier is based on an algorithm Urdhva Tiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. Urdhva Tiryakbhyam Sutra 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 parallelism in generation of partial products and their summation is obtained using Urdhva Tiryakbhyam explained in fig 2.1. 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. Thus the multiplier will require the same amount of time to calculate the product and hence is independent of the clock frequency. The net advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. While a higher clock frequency generally results in increased processing power, its disadvantage is that it also increases power dissipation which results in higher device operating temperatures. By adopting the Vedic multiplier, microprocessors 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 has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore it is time, space and power efficient. It is demonstrated that this architecture is quite efficient in terms of silicon area/speed [10, 4].

1) Multiplication of two decimal numbers- 325*738

To illustrate this multiplication scheme, let us consider the multiplication of two decimal numbers (325 * 738). Line diagram for the multiplication is shown in Fig.2.2. 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. To make the methodology more clear, an alternate illustration is given with the help of line diagrams in figure 2.2 where the dots represent bit „0“ or „1“.

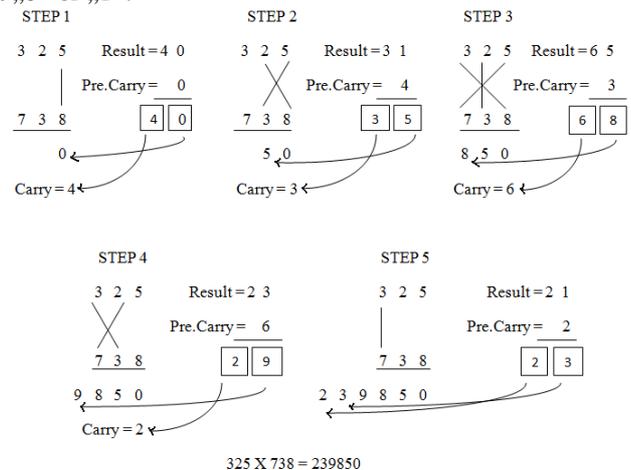


Figure 1: Multiplication of two decimal numbers by Urdhva Tiryakbhyam.

2) Algorithm for 4 x 4 bit Vedic multiplier Using Urdhva Tiryakbhyam (Vertically and crosswise) for two Binary numbers [10]-

CP = Cross Product (Vertically and Crosswise)
 X3 X2 X1 X0 Multiplicand
 Y3 Y2 Y1 Y0 Multiplier

 H G F E D C B A

P7 P6 P5 P4 P3 P2 P1 P0 Product

PARALLEL COMPUTATION METHODOLOGY

1. CP X0 = X0 * Y0 = A

Y0

2. CP X1 X0 = X1 * Y0 + X0 * Y1 = B

Y1 Y0



International Journal of Ethics in Engineering & Management Education

Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 10, October 2014)

3. CP X2 X1 X0 = X2 * Y0 + X0 * Y2 + X1 * Y1 = C
Y2 Y1 Y0
4. CP X3 X2 X1 X0 = X3 * Y0 + X0 * Y3 + X2 * Y1
+ X1 * Y2 = D
Y3 Y2 Y1 Y0
5. CP X3 X2 X1 = X3 * Y1 + X1 * Y3 + X2 * Y2 = E
Y3 Y2 Y1
6. CP X3 X2 = X3 * Y2 + X2 * Y3 = F
Y3 Y2
- 7 CP X3 = X3 * Y3 = G
Y3

3) Algorithm for 8 X 8 Bit Multiplication Using Urdhva Triyakbhyam (Vertically and crosswise) for two Binary numbers [11]-

A = A7A6A5A4 A3A2A1A0
X1 X0
B = B7B6B5B4 B3B2B1B0
Y1 Y0
X1 X0
* Y1 Y0

F E D C
CP = X0 * Y0 = C
CP = X1 * Y0 + X0 * Y1 = D
CP = X1 * Y1 = E
Where CP = Cross Product.

Note: Each Multiplication operation is an embedded parallel 4x4 Multiply module.

To illustrate the multiplication algorithm, let us consider the multiplication of two binary numbers a3a2a1a0 and b3b2b1b0. As the result of this multiplication would be more than 4 bits, we express it as... r3r2r1r0. Line diagram for multiplication of two 4-bit numbers is shown in Fig. 2.2 which is nothing but the mapping of the Fig.2.1 in binary system. For the simplicity, each bit is represented by a circle. Least significant bit r0 is obtained by multiplying the least significant bits of the multiplicand and the multiplier. The process is followed according to the steps shown in Fig.

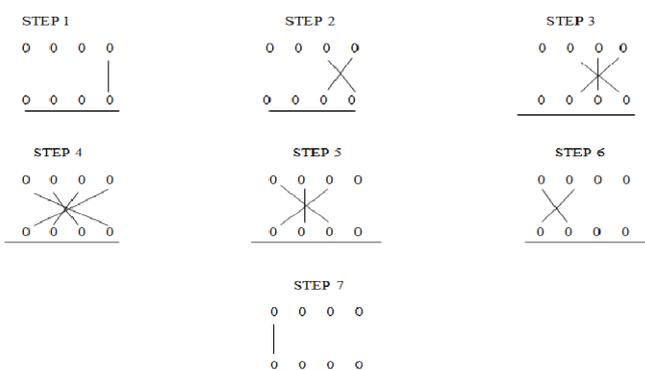


Figure 2: Line diagram for multiplication of two 4 - bit numbers.

Firstly, least significant bits are multiplied which gives the least significant bit of the product (vertical). Then, the LSB of the multiplicand is multiplied with the next higher

bit of the multiplier and added with the product of LSB of multiplier and next higher bit of the multiplicand (crosswise). The sum gives second bit of the product and the carry is added in the output of next stage sum obtained by the crosswise and vertical multiplication and addition of three bits of the two numbers from least significant position. Next, all the four bits are processed with crosswise multiplication and addition to give the sum and carry. The sum is the corresponding bit of the product and the carry is again added to the next stage multiplication and addition of three bits except the LSB. The same operation continues until the multiplication of the two MSBs to give the MSB of the product. For example, if in some intermediate step, we get 110, then 0 will act as result bit (referred as rn) and 11 as the carry (referred as cn). It should be clearly noted that cn may be a multi-bit number.

Thus we get the following expressions:

- r0=a0b0; (1)
- c1r1=a1b0+a0b1; (2)
- c2r2=c1+a2b0+a1b1 + a0b2; (3)
- c3r3=c2+a3b0+a2b1 + a1b2 + a0b3; (4)
- c4r4=c3+a3b1+a2b2 + a1b3; (5)
- c5r5=c4+a3b2+a2b3; (6)
- c6r6=c5+a3b3 (7)

With c6r6r5r4r3r2r1r0 being the final product. Hence this is the general mathematical formula applicable to all cases of multiplication.

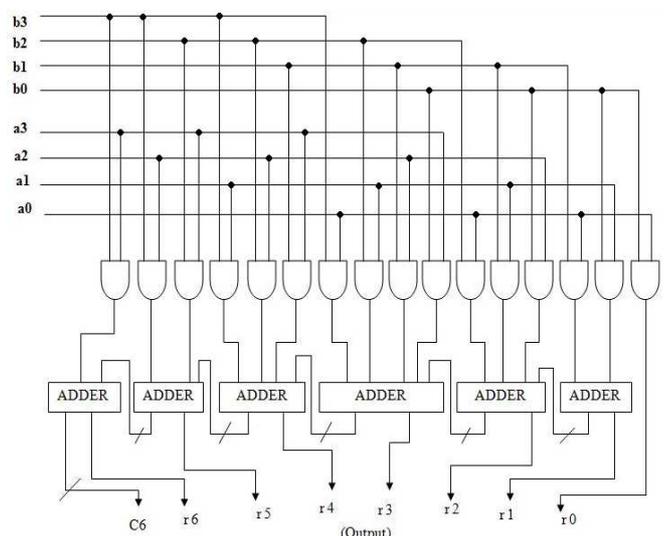


Figure 3: Hardware architecture of the Urdhva triyakbhyam multiplier

The hardware realization of a 4-bit multiplier is shown in figure2.3. This hardware design is very similar to that of the famous array multiplier where an array of adders is required to arrive at the final product. All the partial products are calculated in parallel and the delay associated is mainly the time taken by the carry to propagate through the adders which form the multiplication array. Clearly, this is not an efficient algorithm for the multiplication of large numbers as a lot of propagation delay is involved in such cases. To deal with this



problem, we now discuss Nikhilam Sutra which presents an efficient method of multiplying two large numbers.

3. PROPOSED MULTIPLIER ARCHITECTURE

The hardware architecture of 2X2, 4x4 and 8x8 bit Vedic multiplier module are displayed in the below sections. Here, “Urdhva-Tiryagbhyam” (*Vertically and Crosswise*) sutra is used to propose such architecture for the multiplication of two binary numbers. The beauty of Vedic multiplier is that here partial product generation and additions are done concurrently. Hence, it is well adapted to parallel processing. The feature makes it more attractive for binary multiplications. This in turn reduces delay, which is the primary motivation behind this work.

A. Vedic Multiplier for 2x2 bit Module

The method is explained below for two, 2 bit numbers A and B where $A = a1a0$ and $B = b1b0$ as shown in Fig. 2. Firstly, the least significant bits are multiplied which gives the least significant bit of the final product (vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with, the product of LSB of multiplier and next higher bit of the multiplicand (crosswise). The sum gives second bit of the final product and the carry is added with the partial product obtained by multiplying the most significant bits to give the sum and carry. The sum is the third corresponding bit and carry becomes the fourth bit of the final product.

$$s_0 = a_0b_0; (1)$$

$$c_1s_1 = a_1b_0 + a_0b_1; (2)$$

$$c_2s_2 = c_1 + a_1b_1; (3)$$

The final result will be $c_2s_2s_1s_0$. This multiplication method is applicable for all the cases.

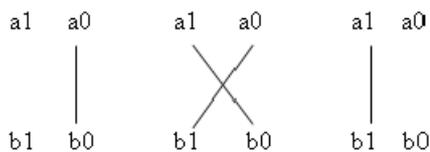


Fig. 2 The Vedic Multiplication Method for two 2-bit Binary Numbers

The 2X2 Vedic multiplier module is implemented using four input AND gates & two half-adders which is displayed in its block diagram in Fig. 3. It is found that the hardware architecture of 2x2 bit Vedic multiplier is same as the hardware architecture of 2x2 bit conventional Array Multiplier [2]. Hence it is concluded that multiplication of 2 bit binary numbers by Vedic method does not made significant effect in improvement of the multiplier’s efficiency. Very precisely we can state that the total delay is only 2-half adder delays, after final bit products are generated, which is very similar to Array multiplier. So we switch over to the implementation of 4x4 bit Vedic multiplier which uses the 2x2 bit multiplier as a basic building block. The same method can be extended for input bits 4 & 8. But for higher no. of bits in input, little modification is required.

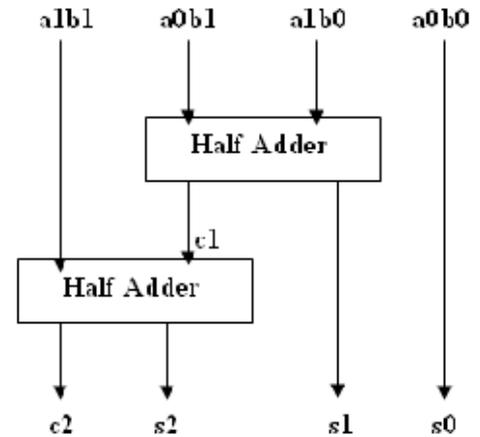


Fig. 4 Block Diagram of 2x2 bit Vedic Multiplier

B. Vedic Multiplier for 4x4 bit Module

The 4x4 bit Vedic multiplier module is implemented using four 2x2 bit Vedic multiplier modules as discussed in Fig. 3. Let’s analyze 4x4 multiplications, say $A = A_3 A_2 A_1 A_0$ and $B = B_3 B_2 B_1 B_0$. The output line for the multiplication result is $S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0$. Let’s divide A and B into two parts, say $A_3 A_2$ & $A_1 A_0$ for A and $B_3 B_2$ & $B_1 B_0$ for B. Using the fundamental of Vedic multiplication, taking two bit at a time and using 2 bit multiplier block, we can have the following structure for multiplication as shown in Fig. 4.

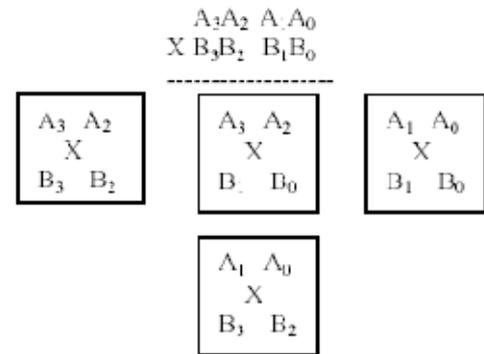


Fig. 5 Sample Presentation for 4x4 bit Vedic Multiplication

Each block as shown above is 2x2 bit Vedic multiplier. First 2x2 bit multiplier inputs are A_1A_0 and B_1B_0 . The last block is 2x2 bit multiplier with inputs $A_3 A_2$ and $B_3 B_2$. The middle one shows two 2x2 bit multiplier with inputs $A_3 A_2$ & B_1B_0 and A_1A_0 & $B_3 B_2$. So the final result of multiplication, which is of 8 bit, $S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0$. To understand the concept, the Block diagram of 4x4 bit Vedic multiplier is shown in Fig. 5. To get final product ($S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0$), four 2x2 bit Vedic multiplier (Fig. 3) and three 4-bit Ripple-Carry (RC) Adders are required. The proposed Vedic multiplier can be used to reduce delay. Early literature speaks about Vedic multipliers based on array multiplier structures. On the other hand, we proposed a new architecture, which is efficient in terms of speed. The arrangements of RC Adders shown in Fig. 5, helps us to



reduce delay. Interestingly, 8x8 Vedic multiplier modules are implemented easily by using four 4x4 multiplier modules.

D. Generalized Algorithm for N x N bit Vedic Multiplier

We can generalize the method as discussed in the previous sections for any number of bits in input. Let, the multiplication of two N-bit binary numbers (where $N = 1, 2, 3, \dots, N$, must be in the form of $2N$) A and B where $A = A_N \dots A_3 A_2 A_1$ and $B = B_N \dots B_3 B_2 B_1$. The final multiplication result will be of $(N + N)$ bits as $S = S(N + N) \dots S_3 S_2 S_1$. **Step 1:** Divide the multiplicand A and multiplier B into two equal parts, each consisting of $[N$ to $(N/2)+1$] bits and $[N/2$ to $1]$ bits respectively, where first part indicates the MSB and other represents LSB. **Step 2:** Represent the parts of A as A_M and A_L , and parts of B as B_M and B_L . Now represent A and B as $A_M A_L$ and $B_M B_L$ respectively. **Step 3:** For $A \times B$, we have general format as shown in Fig.

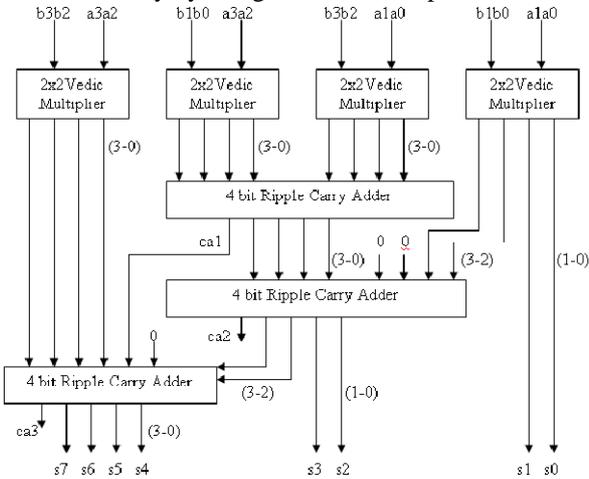


Fig. 6 Block Diagram of 4x4 bit Vedic Multiplier

C. Vedic Multiplier for 8x8 bit Module

The 8x8 bit Vedic multiplier module as shown in the block diagram in Fig. 6 can be easily implemented by using four 4x4 bit Vedic multiplier modules as discussed in the previous section. Let's analyze 8x8 multiplications, say $A = A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$ and $B = B_7 B_6 B_5 B_4 B_3 B_2 B_1 B_0$. The output line for the multiplication result will be of 16 bits as $S_{15} S_{14} S_{13} S_{12} S_{11} S_{10} S_9 S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0$. Let's divide A and B into two parts, say the 8 bit multiplicand A can be decomposed into pair of 4 bits $A_H A_L$. Similarly multiplicand B can be decomposed into $B_H B_L$. The 16 bit product can be written as:

$$P = A \times B = (A_H A_L) \times (B_H B_L) = A_H \times B_H + (A_H \times B_L + A_L \times B_H) + A_L \times B_L$$

Using the fundamental of Vedic multiplication, taking four bits at a time and using 4 bit multiplier block as discussed we can perform the multiplication. The outputs of 4x4 bit multipliers are added accordingly to obtain the final product. Here total three 8 bit Ripple-Carry Adders are required as shown in Fig.

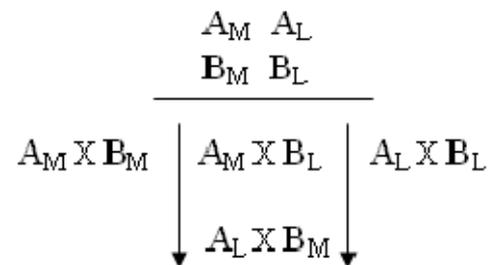


Fig. 8 General Representation for Vedic multiplication

Step 4: The individual multiplications product can be obtained by the partitioning method and applying the basic building blocks. By adopting the above generalized algorithm we can implement Vedic Multiplier for any number of bits say 16, 32, 64, and so on, as per the requirement. Therefore, it could be possible to implement this Vedic multiplier in the ALU (Arithmetic Logic Unit) which will reduce the computational speed drastically & hence improves the processors efficiency.

4. CONCLUSION

This paper presents a highly efficient method of multiplication "Urdhva Tiryakbhyam Sutra" based on Vedic mathematics. It gives us method for hierarchical multiplier design and clearly indicates the computational advantages offered by Vedic methods. The computational path delay for proposed 8x8 bit Vedic multiplier is found to be 21.679 ns. Hence our motivation to reduce delay is finely fulfilled. Therefore, we observed that the Vedic multiplier is much more efficient than Array and Booth multiplier in terms of execution time (speed). An awareness of Vedic mathematics can be effectively increased if it is included in engineering education. In future, all the major universities may set up appropriate research centers to promote research works in Vedic mathematics. References

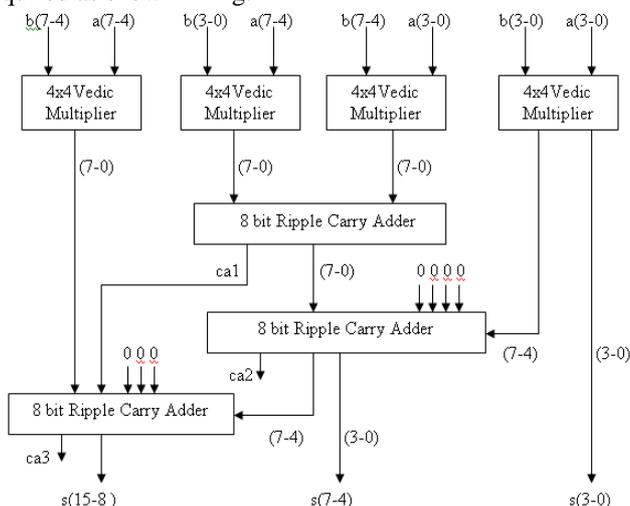


Fig. 7 Block Diagram of 8x8 bit Vedic Multiplier

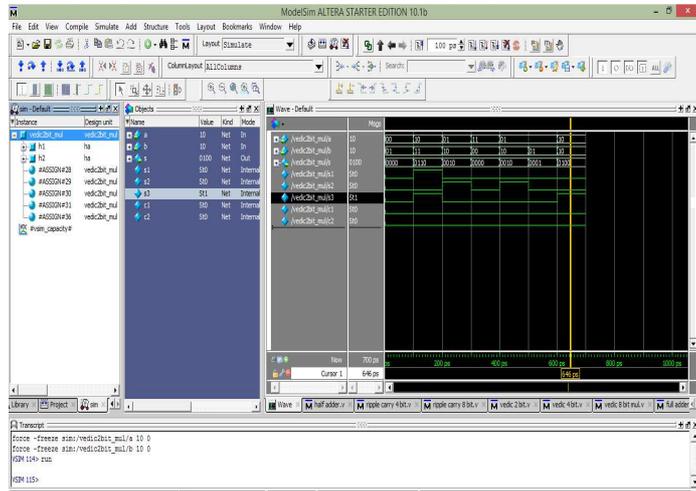


International Journal of Ethics in Engineering & Management Education

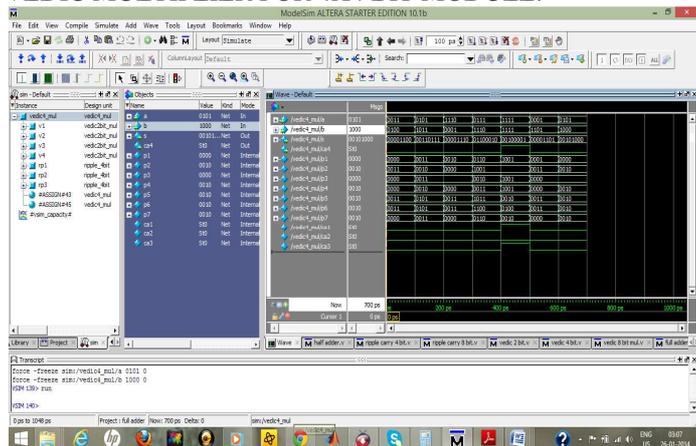
Website: www.ijee.in (ISSN: 2348-4748, Volume 1, Issue 10, October 2014)

5. SCREEN SHOTS

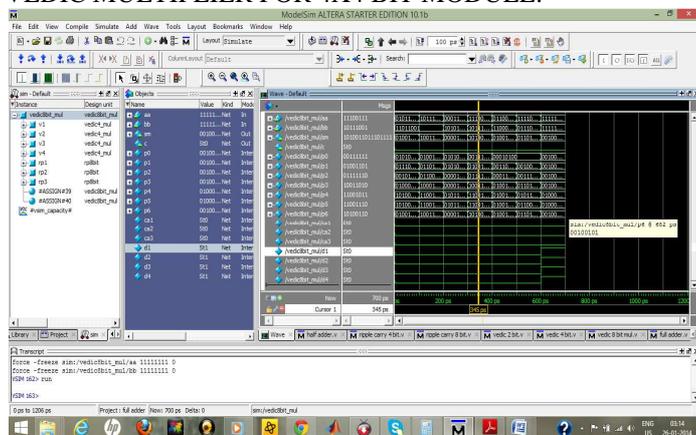
VEDIC MULTIPLIER FOR 2X2 BIT MODULE:



VEDIC MULTIPLIER FOR 4X4 BIT MODULE:



VEDIC MULTIPLIER FOR 4X4 BIT MODULE:



REFERENCES

[1]. Jagadguru Swami, Sri Bharati Krisna, Tirthaji Maharaja, "Vedic Mathematics or Sixteen Simple Mathematical Formulae from the Veda, Delhi (1965)", Motilal Banarsidas, Varanasi, India, 1986.

[2]. M. Morris Mano, "Computer System Architecture", 3rd edition, Prentice-Hall, New Jersey, USA, 1993, pp. 346-348.

[3]. H. Thapliyal and H.R Arbania, "A Time-Area-Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics", Proceedings of the 2004 International Conference on VLSI (VLSI'04), Las Vegas, Nevada, June 2004, pp. 434-439.

[4]. P. D. Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engg. Edu, Vol.8, No.2, 2004, UICEE Published in Australia.

[5]. Thapliyal H. and Srinivas M.B, "High Speed Efficient NxN Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics", Transactions on Engineering, Computing and Technology, 2004, Vol.2.

[6]. 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.

[7]. Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim and Yong Beom Cho, "Multiplier design based on ancient Indian Vedic Mathematician", International SoC Design Conference, pp. 65- 68, 2008.

[8]. Parth Mehta and Dhanashri Gawali, "Conventional versus Vedic mathematics method for Hardware implementation of a multiplier", International conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 640-642, 2009.

[9]. Ramalatha, M.Dayalan, K D Dharani, P Priya, and S Deborah, "High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques", International Conference on Advances in Computational Tools for Engineering Applications (ACTEA) IEEE, pp. 600-603, July 15-17, 2009.

[10]. Sumita 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, pp. 47-56, July 2010.

[11]. S.S.Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V A "Implementation of Vedic Multiplier For Digital Signal Processing" International conference on VLSI communication & instrumentation (ICVCI), 2011.

[12]. Asmita Haveliya, "A Novel Design for High Speed Multiplier for Digital Signal Processing Applications (Ancient Indian Vedic mathematics approach)", International Journal of Technology and Engineering System (IJTES), Vol.2, No.1, pp. 27-31, Jan-March, 2011.