INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

DSP Based Vedic Multiplier

Abdul Lateef Haroon.P.S[1], Mamatha.G.M[2]

[1]*Assistant Professor, Department of EEE, Bellary Institute of Technology & Management,Ballari, Karnataka.*

[2]*Assistant Professor, Department of EEE, Sri Sai Ram College of Engineering, Bangalore, Karnataka*

Abstract— Digital signal processors (DSPs) are very imperative in various methodological field in the at hand scenario. In today's world Fast burgeoning is one of the very important methods in DSPs. Fast multiplication is used for intricacy, Fourier Analysis etc. A fastest technique for multiplication based on ancient Indian Vedic arithmetic is proposed in this paper. Among the poles apart methods of multiplications in Vedic mathematics, Urdhva tiryakbhyam will be discussed in detail. Urdhva tiryakbhyam is one of the widespread multiplication formulae pertinent to different cases of multiplication. This is a highly modular drawing in which smaller blocks can be used to build higher blocks. The proposed architecture is done for two 8-bit numbers; the multiplier and multiplicand, each are grouped as 4-bit statistics so that it decomposes into 4×4 multiplication modules. It is also illustrated that the further hierarchical decomposition of 4×4 modules into 2×2 modules will not have a significant effect in improvement of the multiplier efficiency or in other words multiplier decomposition nearly reaches a saturation level in its efficiency at 4×4 decomposition. The coding is done in VHDL (very high speed integrated circuits hardware description language) and synthesis is done using Xilinx ISE series [1]. The combinational delay obtained after synthesis is compared with the performance of the modified Booth Wallace multiplier which is a fast multiplier. This Vedic multiplier can bring about great improvement in DSP performance.
Keywords— Urdhva Tiryakbhyam, DSP, CSA Carry Save Array.

I.   INTRODUCTION

High speed arithmetic operations are very important in many signal processing applications. Speed of the digital signal processor (DSP) is largely determined by the speed of its multipliers. In fact the multipliers are the most important part of all digital signal processors; they are very important in realizing many important functions such as fast Fourier transforms and convolutions. Since a processor spends considerable amount of time in performing multiplication, an improvement in multiplication speed can greatly improve system performance. Multiplication can be implemented using many algorithms such as array, booth, carry save, and Wallace tree algorithms. The computational time required by the array multiplier is less because the partial products are computed independently in parallel. The delay associated with the array multiplier is the time taken by the signals to propagate through the gates that form the multiplication array. Arrangement of adders is another way of improving multiplication speed. There are two methods for this: Carry save array (CSA) method and Wallace tree method. In the CSA method, bits are processed one by one to supply a carry signal to an adder located at a one bit higher position. The CSA method has got its own limitations since the execution time depends on the number of bits of the multiplier. In the Wallace tree method, three bit signals are passed to a one bit full adder and the sum is supplied to the next stage full adder of the same bit and the carry output signal is passed to the next stage full adder of same number of bit and then the formed carry is supplied to the next stage of the full adder located at a one bit higher position. In this method, the circuit layout is not easy [1]. Booth algorithm reduces the number of partial products. However, large booth arrays are required for high speed multiplication and exponential operations which in turn require large partial sum and partial carry registers. Multiplication of two n-bit operands using a radix-4 booth recording multiplier requires approximately n/ (2m) clock cycles to generate the least significant half of the final product, where m is the number of booth recoded adder stages. Thus, a large propagation delay is associated with this case. The modified booth encoded Wallace tree multiplier uses modified booth algorithm to reduce the partial products and also faster additions are performed using the Wallace tree. This paper proposes a novel fast multiplier adopting the sutra of ancient Indian Vedic mathematics called Urdhva tiryakbhyam. The design of the multiplier is faster than existing multipliers reported previously.

In this paper, after a gentle introduction of Urdhva tiryakbhyam Sutra, multiplier architecture is proposed and the architecture is illustrated with two 8-bit numbers; the multiplier and multiplicand, each are grouped as 4-bit numbers so that it decomposes into 4×4 multiplication modules. After decomposition, vertical and crosswise algorithm is applied to carry out the multiplication of first 4×4 multiply modules. The results of first 4×4 multiplication module are utilized after getting the sub product bits parallel from the subsequent module to generate the final 16-bit product. Hence any complex N×N multiplication can be efficiently implemented by

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

using small 4×4 multiplier using the proposed architecture where N is a multiple of 4 such as 8, 12, 16, 20, 24…...4N. Therefore efficient multiplication algorithm implementation with small numbers such as 4-bits, can be easily extended and embedded for implementing efficient N×N multiply operation. This paper emphasizes that the further hierarchical decomposition of 4×4 modules into 2×2 modules will not have a significant effect in improvement of the multiplier efficiency n terms of area and speed [2].

II. VEDIC MULTIPLICATION ALGORITHMS

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.

A. 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.

*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".
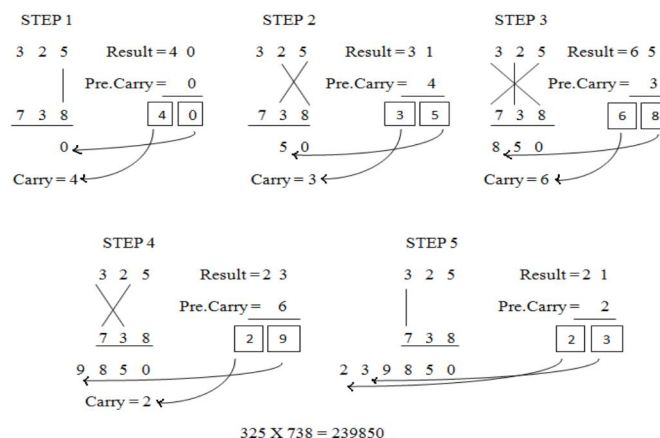

Fig 2.1: Multiplication of two decimal numbers by Urdhva Tiryakbhyam.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

2) Algorithm For 4 x 4 Bit Vedic Multiplier Using Urdhva Tiryakbhyam (Vertically And Crosswise) For Two Binary Numbers

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 $\begin{array}{c} X0 \\ Y0 \end{array}$ $= X0 * Y0 = A$

2. CP $\begin{array}{c} X1\ X0 \\ Y1\ Y0 \end{array}$ $= X1 * Y0 + X0 * Y1 = B$

3. CP $\begin{array}{c} X2\ X1\ X0 \\ Y2\ Y1\ Y0 \end{array}$ $= X2 * Y0 + X0 * Y2 + X1 * Y1 = C$

4. CP $\begin{array}{c} X3\ X2\ X1\ X0 \\ Y3\ Y2\ Y1\ Y0 \end{array}$ $= X3 * Y0 + X0 * Y3 + X2 * Y1 + X1 * Y2 = D$

5. CP $\begin{array}{c} X3\ X2\ X1 \\ Y3\ Y2\ Y1 \end{array}$ $= X3 * Y1 + X1 * Y3 + X2 * Y2 = E$

6. CP $\begin{array}{c} X3\ X2 \\ Y3\ Y2 \end{array}$ $= X3 * Y2 + X2 * Y3 = F$

7 CP $\begin{array}{c} X3 \\ Y3 \end{array}$ $= X3 * Y3 = G$

3) Algorithm For 8 X 8 Bit Multiplication Using Urdhva Tiryakbhyam (Vertically And Crosswise) For Two Binary Numbers

```
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.

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

380

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

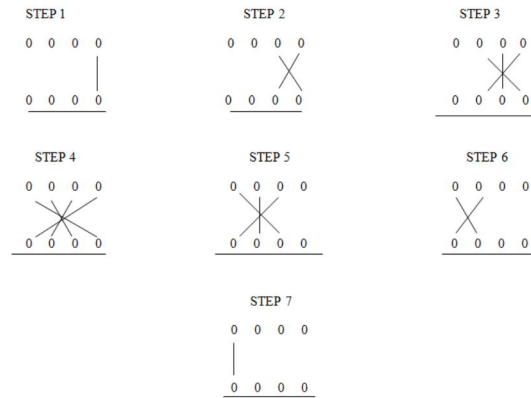multiplier. The process is followed according to the steps shown in Fig. 2.1.

FIG 2.2: 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.

Vedic multiplier is faster than array multiplier and Booth multiplier. As the number of bits increases from 8x8 bits to 16x16 bits, the timing delay is greatly reduced for Vedic multiplier as compared to other multipliers. Vedic multiplier has the greatest advantage as compared to other multipliers over gate delays and regularity of structures. Delay in Vedic multiplier for 16 x 16 bit number is 32 ns while the delay in Booth and Array multiplier are 37 ns and 43 ns respectively. Thus this multiplier shows the highest speed among conventional multipliers. It has this advantage than others to prefer a best multiplier.

The area needed for Vedic square multiplier is very small as compared to other multiplier architectures i.e. the number of devices used in Vedic square multiplier are 259 while Booth and Array Multiplier is 592 and 495 respectively for 16 x 16 bit number when implemented on Spartan FPGA. Thus the result shows that the Vedic square multiplier is smallest and the fastest of the reviewed architectures.

III. THE MULTIPLIER ARCHITECTURE

The multiplier architecture is based on this Urdhva tiryakbhyam sutra. The advantage of this algorithm is that partial products and their sums are calculated in parallel. This parallelism makes the multiplier clock independent [2]. The other main advantage of this multiplier as compared to other multipliers is its regularity. Due to this modular nature the lay out design will be easy. The architecture can be explained with two eight bit numbers i.e. the multiplier and multiplicand are eight bit numbers. The multiplicand and the multiplier are split into four bit blocks. The four bit blocks are again divided into two bit multiplier blocks. According to the multiplier the 8×8 (A×B) bit multiplication will be as follows:

A = AH - AL, B = BH - BL

A = A7A6A5A4A3A2A1A0

B = B7B6B5B4B3B2B1B0

AH = A7A6A5A4, AL = A3A2A1A0

BH = B7B6B5B4, BL = B3B2B1B0

By the algorithm, the product can be obtained as follows.

Product of A x B = AL x BL + AH x BL + AL x BH + AH x BH

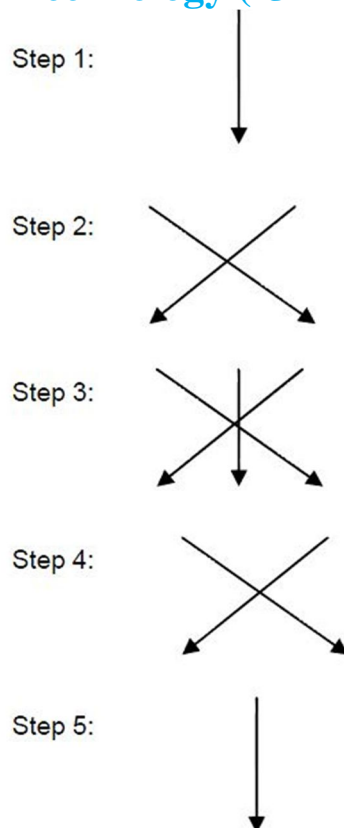## International Journal for Research in Applied Science & Engineering Technology (IJRASET)
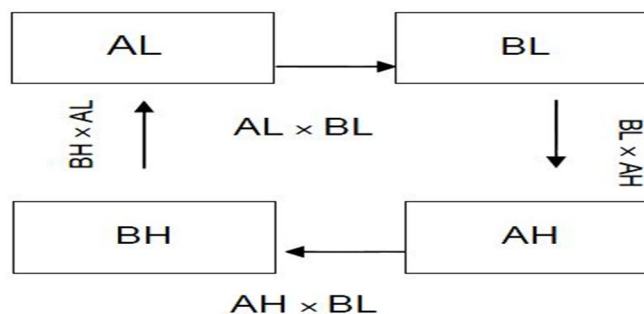

Fig 3.1: Methodology of multiplication

Fig 3.2: Parallel Multiplication of 8 x8 bits

The 4 x 4 bit multiplication can be again reduced to 2 x 2 bit multiplications. The 4 bit multiplicand and the multiplier are divided into two-bit blocks.

AH = AHH - AHL

BH = BHH - BHL

AH x BH = AHL x BHL + AHH x BHL + AHL x BHH + AHH x BHH
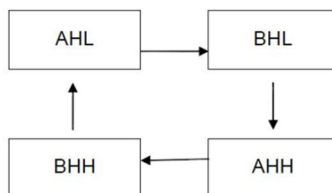
Here the parallel multiplications are

Fig 3.3: Parallel Multiplication of 4 x4 bits

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

THUS 8 X 8 MULTIPLICATIONS CAN BE DECOMPOSED INTO 2 X 2 MULTIPLICATION UNITS. BY USING THIS ALGORITHM ANY COMPLEX N X N MULTIPLICATION CAN BE IMPLEMENTED USING THE BASIC 2×2 MULTIPLIER UNITS

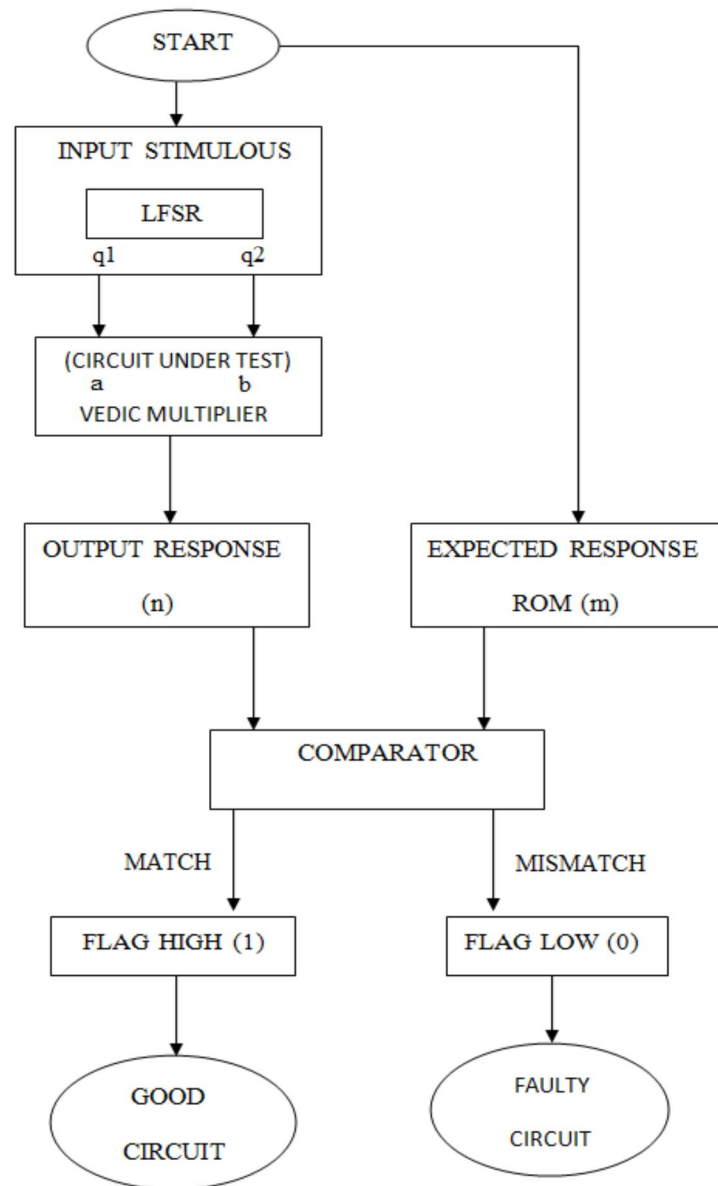## IV. VERIFICATION AND IMPLEMENTATION

Fig 4: Verification Flow Chart

In this work the algorithms are implemented in VHDL and logic simulations are done in Modelsim simulator and the synthesis is done using Xilinx - project navigator.

The result is grouped in [Table4.1] for different bit multiplications of the Vedic multiplier. [Table 4.1] and [Table 4.2] shows the difference in combinational delays between the Vedic multiplier and Booth Wallace multiplier for 8 x 8 and 16 x 16 bit multiplication. The highest performances for both multipliers are seen on the device Virtex2p with a speed grade of -7. The

383

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

combinational delays for both the multipliers are same for 8 x 8 multiplications. But for the 16 x 16 multiplication the Vedic multiplier shows a very improved performance over the modified Booth Wallace multiplier. The results suggest that Vedic multiplier is an extreme fast multiplier and is well ahead of the modified Booth Wallace multiplier………………………………………………………………………….

The Simulation Waveforms Of Various Vedic Multipliers Is As Shown Below For 2×2 Multiplier, For 4×4 Multiplier, For 8×8 Multiplier
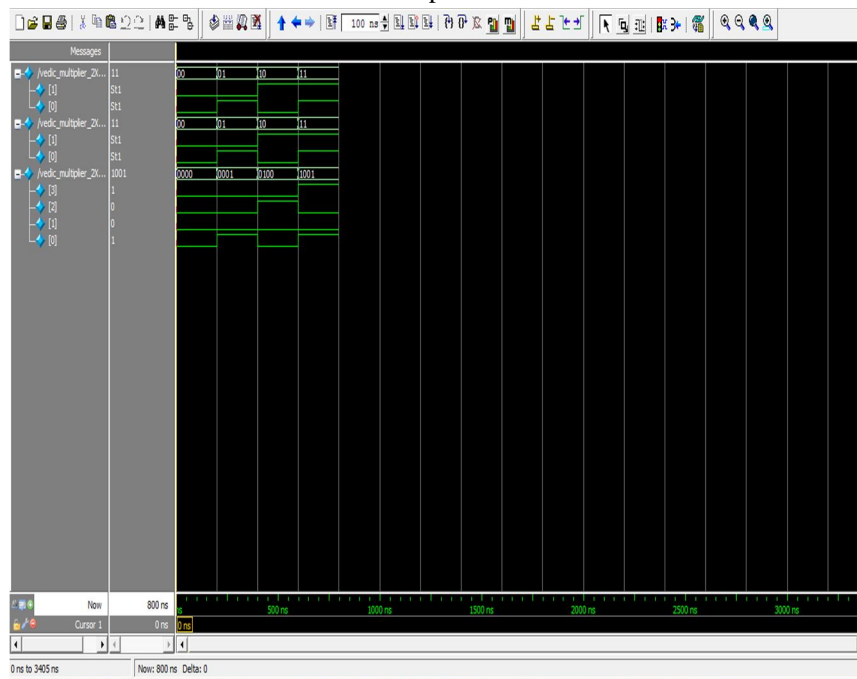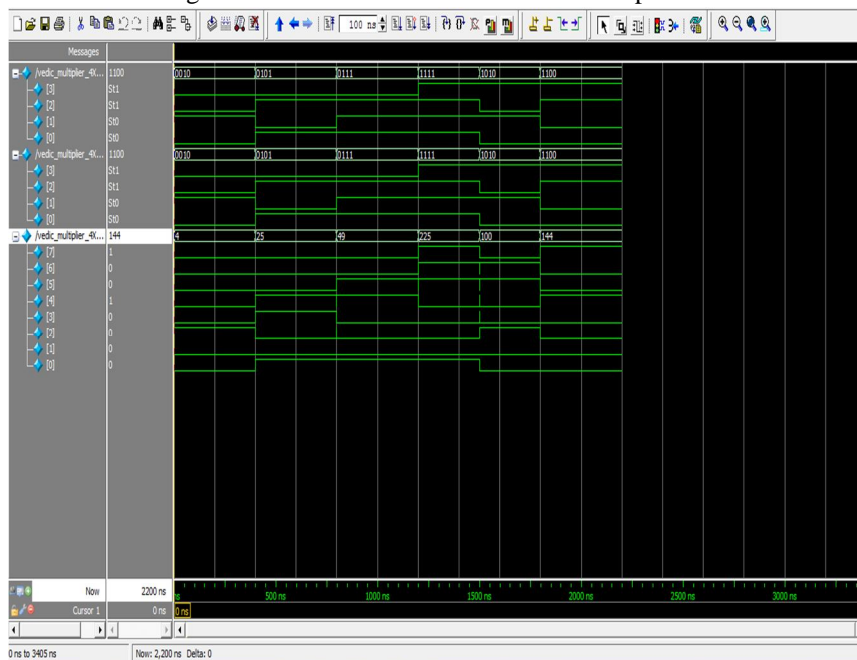

Fig: 5.1 Simulation Waveform of 2×2 Multiplier

Fig 5.2 Simulation Waveform of 4×4 Multiplier

384

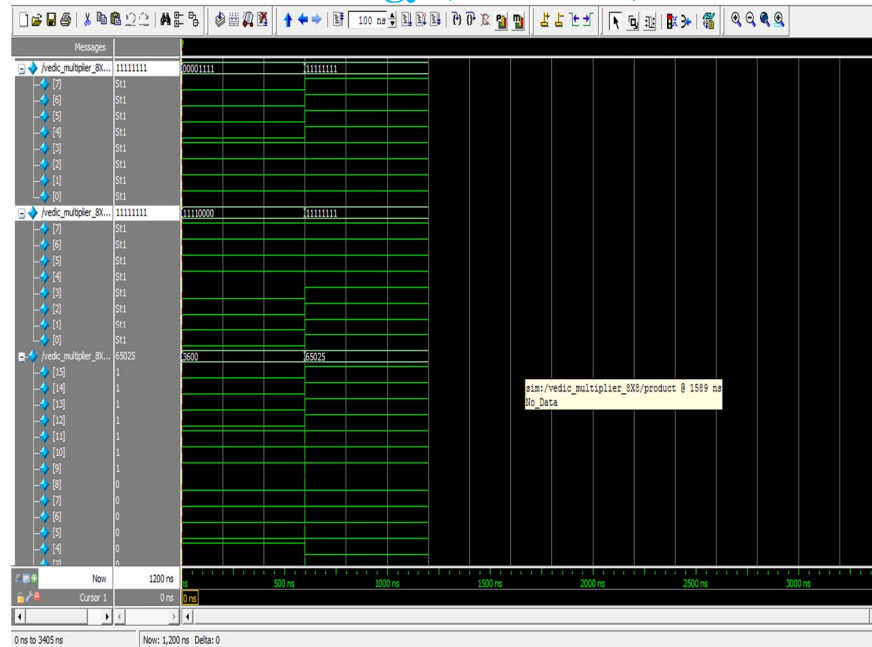# International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Fig 5.3 Simulation Waveform of 8×8 Multiplier

V. ADVANTAGES

The main advantage is its regularity when compared to other multipliers.
It is easy, simple, direct and straightforward.
The partial products and their sums are calculated in parallel.
The **Sutras** are easy to understand, easy to apply and easy to remember, and the whole work can be truthfully summarized.

VI. APPLICATIONS

Mainly used in Digital Signal Processor.
Can be applied in those designs in which smaller blocks are used to build higher blocks.
Can be efficiently applied in any Signal Processing applications.

VII. CONCLUSION

The proposed Vedic multiplier proves to be highly efficient in terms of speed. Due to its regular and parallel structure it can be realized easily on silicon as well. The main advantages delay increases slowly as input bits increase. The designs of 16x16 bits, Vedic multiplier have been implemented on Spartan XC3S500-5-FG320 and XC3S1600-5-FG484 device. The computation delay for 16x16 bits Booth multiplier was 20.09 ns and for 16x16 bits Vedic multiplier was 6.960 ns. Also computation delays for 32x32 bits and 64x64 bits Vedic multiplier was obtained 7.784 ns and 10.241 ns respectively. It is therefore seen that the Vedic multipliers are much faster than the conventional multipliers. The algorithms of Vedic mathematics are much more efficient than of conventional mathematics.
Urdhva tiryakbhyam sutra algorithm can reduce the delay, power and hardware requirements for multiplication of numbers.
Built in Self Tests of the implemented algorithms further proves that the design of Vedic multiplier circuit is fault free.

REFERENCES

[1] International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011 Proceedings published by International Journal of Computer Applications® (IJCA).
[2] International Journal of Technology and Engineering System (IJTES):Jan-March 2011-vol2.No1.
[3] Purushottam D. Chidgupkar and Mangesh T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Eng. Educ., Vol.8, No.2 © 2009 UICEE Published in Australia.

385

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

[4]   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 Centre Athens, Georgia 30602-7404, U.S.A.

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

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

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

[8]   Charles E. Stroud, "A Designer's Guide to Built-In Self-Test", University of North Carolina at Charlotte, ©2002 Kluwer Academic Publishers New York, Boston, Dordrecht, London, Moscow.

[9]   Douglas Densmore, "Built-In-Self Test (BIST) Implementations An overview of design tradeoffs", University of Michigan EECS 579 – Digital Systems Testing by Professor John P. Hayes 12/7/01.

[10] Shripad Kulkarni, "Discrete Fourier Transform (DFT) by using Vedic Mathematics", report, vedicmathsindia.blogspot.com, 2010.

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

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 ⟲ (24*7 Support on Whatsapp)