



# FPGA IMPLEMENTATION OF 32-BIT VEDIC MULTIPLIER AND SQUARE ARCHITECTURES

**T.SRAVANTHI DEVI**

MTECHVLSI Design, Dept of ECE, ANURAG GROUP OF INSTITUTIONS, TS, India.

**A.MADHU SUDHAN**

Assistant Professor, Dept of ECE, ANURAG GROUP OF INSTITUTIONS, TS, India.

## ABSTRACT

The main objective of the project is to design Vedic Multiplier and Square Architectures based upon ancient Indian Vedic mathematics sutras for DSP applications. In this work, all the partial products are adjusted using concatenation operation and are added by using single carry save adder instead of two adders at different stages. The high speed Vedic multiplier architecture is then used in the squaring modules. The key to our success is that only one Vedic Multiplier is used instead of four multipliers. The reduced number of computations in multiplication due to adjusting using concatenation operation and one carry save adder only, the designed multiplier offers significant improvement in speed, reduces delay. The Proposed method is Coded in Verilog, Synthesized and Simulated using Xilinx ISE Tool 13.2. We are designing our architecture in Verilog HDL code and also simulated by using Vivado 2015.2 tool and Hardware Implementation on ZYNQ Board (FPGA). The Project functionality input and output to the system is in digital data. And the process expected to be done in the system is Multiplication and Squaring. The specifications of vedic multiplier and square architectures are of 32bits digital data input bit size and 64bits digital data output bit size.

**KEYWORDS** : VLSI, Vedic, Urdhava-Tiryakbhayam, Dwandwa-yoga, Multiplier, Square, IXI.

## I. INTRODUCTION

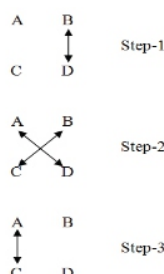
The word 'Vedic' was driven from the word 'Veda' which is ancient store-house of all knowledge. Vedic mathematics provides the solution to the problem of long computation time by reducing the time delay needed for the operations to be performed. It has originated from "Atharva Vedas" the fourth Veda. Atharva Veda mainly deals with the branches like engineering, mathematics, sculpture, medicines and all other sciences. Vedic mathematics deals with all areas of mathematics either it is pure or applied. It was rediscovered from the Vedas between 1911 and 1918 by Sri Bharati Krishna Tirtha Ji. Vedic mathematics has been formulated on sixteen sutras and thirteen sub-sutras. These sutras offer magical short cut methods to all basic mathematical operations. All the advantages drives from the fact that Vedic mathematics approach is totally different and considered very close to the way a human mind works. Vedic mathematics can be applied to every branch of mathematics including arithmetic, algebra and geometry. The powerful applications of Vedic mathematics are in the fields of Digital Signal Processing (DSP), Chip Designing, Discrete Fourier Transform (DFT), High Speed Low Power VLSI Arithmetic and Algorithms and encryption systems [1].

Hence it is imperative to have faster additions, multiplications, squaring and cubing etc. These Vedic mathematics based modules along with other modules can be integrated in ALU.

## II. DESIGN APPROACH

### 1. VEDIC MULTIPLIER ARCHITECTURE

This section introduces multiplication operation using Vedic IXI Methodology and then illustrates architecture of 2x2 multiplier module and finally architecture of n-bit multiplier. Urdhava-Tiryakbhayam sutra has been used for multiplication purpose. The figure 1 explains multiplication of two decimal numbers using IXI technique:-

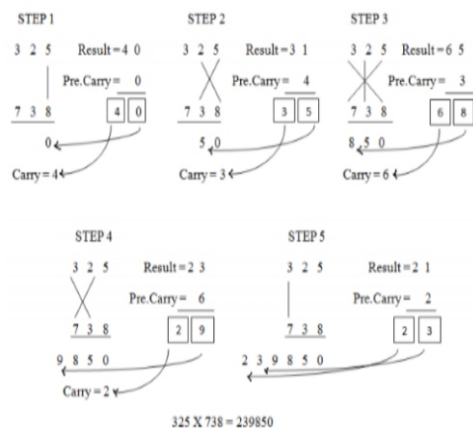


**Figure 1: IXI Methodology or Multiplication**

In this method initially multiplication of the rightmost digit of multiplier is performed with rightmost digit of the multiplicand giving the LSB of the product term as shown in step-1 of figure 1. Then multiplication of the rightmost digit of the multiplier is performed with leftmost digit of the multiplicand and leftmost digit of the multiplier is performed with rightmost digit of the multiplicand and then added together. Thus forming the middle part of the product term as shown in step-2. At the last, in step-3 the leftmost digit of the multiplier is performed with leftmost digit of the multiplicand giving the product term. In this way the multiplication process is carried out. Similar logic of cross-multiplication and addition can be extended to implement any number of bits. Each iteration gives the coefficient of the final product.

### Ex. Multiplication of two decimal numbers: 325x738

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 as shown in figure 2, 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.

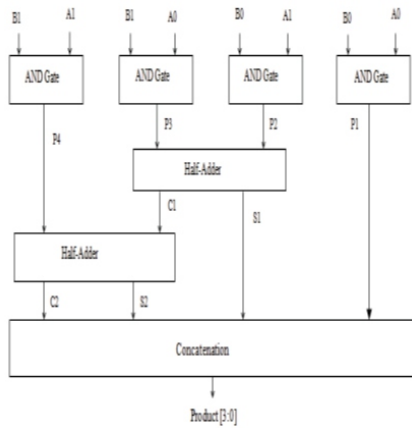


**Figure 2: Example for UT sutra**

### A. 2x2 Multiplier Module

The 2x2 multiplier module can be implemented by using four AND

gates and two Half-adder modules as shown in figure 3. The total delay for 2x2 multiplier is only two half-adder delay.



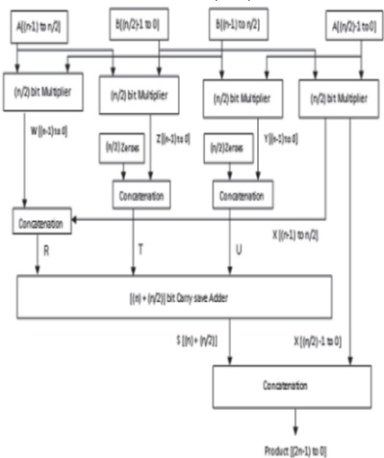
**Figure 3: Architecture of 2x2 Multiplier Module**

It consists of four AND gates used for AND operation of single bit numbers, two half adders for the addition of numbers obtained in the previous section to provide the sum and carry. At the end concatenation of all the final results is done to obtain the final product of 4-bits.

This architecture is basically implementation of IX1 technique. Firstly, right most digits are multiplied to give the LSB of the final product. Then LSB of first digit is multiplied with the MSB of second digit and MSB of first digit with the LSB of the second digit and are added together using half adder block. The sum obtained from this adder block is the second digit of the product. At the next step MSB's of both the numbers are ANDed together and the result is added with the previous carry through half adder. So, the obtained sum makes the third digit of the final product and carry will be the MSB of the product.

**B. n-bit Vedic Multiplier Module**

The architectural concept of n-bit Vedic Multiplier is shown in figure 4. This architecture consists of four n/2 bit multipliers used for calculating the partial products. Next, the results of these n/2 bit multipliers are adjusted using concatenation operation to have all the partial product terms of equal bit-length. The partial product of right most multiplier is concatenated with the partial product of leftmost multiplier and the partial products of middle two multipliers are concatenated with (n/2) zeroes each.



**Figure 4: Architecture of n-bit Multiplier**

All the numbers obtained after concatenations are added together using single carry save adder. At the end, the sum obtained from the carry save adder is concatenated with LSB partial product of right

most multiplier to get the required final product.

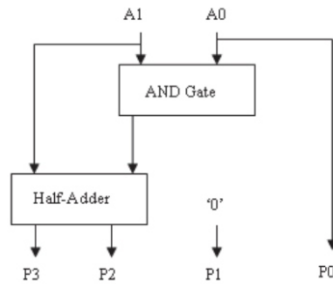
**2. SQUARING ARCHITECTURE**

In most of the computations, the multiplier unit is used to compute the square of an operand. Since square is a special case of multiplication, a dedicated square hardware will significantly improve the computation time. The squaring algorithm makes use of the Duplex or Dwandwa (D) operator. In the Duplex, we take twice the product of the outermost pair, and then add twice the product of the next outermost pair, and so on till no pairs are left. When there are odd number of bits in the original sequence there is one bit left by itself in the middle, and this enters as its square.

- For 1-bit number (X0),  $D=X0*X0$ .
- For 2-bit number (X0X1),  $D=2*X1*X0$ .
- For 3-bit number (X2X1X0),  $D=2*X2*X0+X1*X1$
- For 4-bit number (X3X2X1X0),  $D=2*X3*X0+2*X2*X1$ .

**A. 2-bit Squaring Circuit**

For calculating the square of 2-bit number, one AND gate and one Half-adder are used as shown in figure 5.



**Figure 5: Architecture of 2-bit Squaring Circuit**

The LSB of the input is directly taken as LSB of the final output. The next bit of the output will always be '0'. So, it is by default set to zero value always. Then both the inputs are ANDed together. The result is added to the MSB of the input numbers. The sum of the half-adder is taken as the third bit of the final result and the carry bit of the half-adder is the MSB of the final output. Thus, we get the final output of 4-bit for squaring module of two-bit number. This two-bit squaring circuit is the base circuit for developing the n-bit squaring module.

**B. n-bit Squaring Circuit**

The architecture for n-bit squaring module is shown in figure 6.



**Figure 6: Architecture of n-bit Square module**

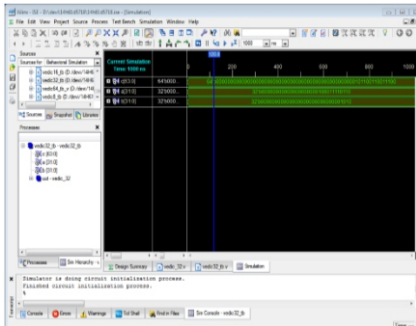
In this architecture, one multiplier of n/2 bits along with two squaring circuits of n/2 bits is used for the calculation of the partial products. At the last step all the calculated partial products are adjusted accordingly and are added together using the single carry save adder to get the final output.

**III. TEST RESULTS**

**A. Simulation Results**

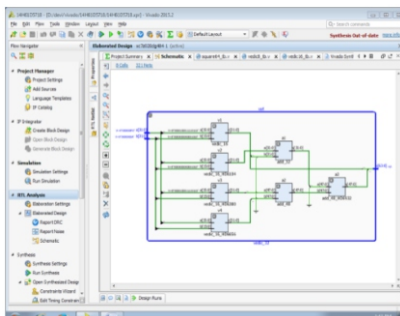
**i. Vedic Multiplier**

The Vedic Multiplier is coded in Verilog and Simulated by using VivadoTool.



**Figure 7: Simulation result for 32-bit Vedic Multiplier**

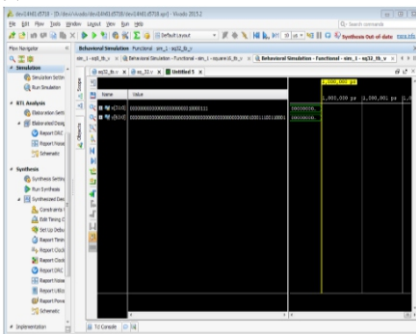
**Interpretation:** From the above simulation waveform, when the inputs are  $a=2294$ ,  $b=10$  then the output of the 32-bit Vedic Multiplier is  $c=22940$ . The representation of inputs and output is in the binary form as seen in figure 7.



**Figure 8: RTL Schematic View for 32-bit Vedic Multiplier**

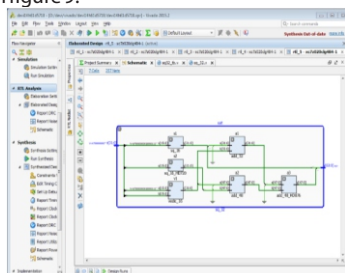
**ii. Square Module**

The Square module is coded in Verilog and Simulated by using VivadoTool.



**Figure 9: Simulation result for 32-bit Square Module**

**Interpretation:** From the above simulation waveform, when the input  $x=135$  then the output of the 32-bit Square Module is  $y=18225$ . The representation of input and output is in the binary form as seen in figure 9.



**Figure 10: RTL Schematic view for 32-bit Square Module**

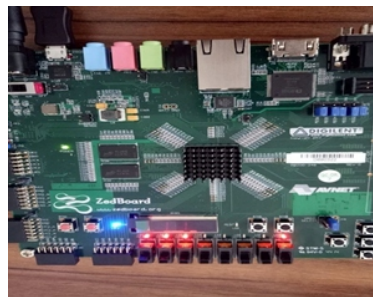
**B. Hardware Implementation Results**

ZYNQ Board(FPGA) 7000 series was implemented in family XC7Z020 with package number CLG484 and speed grade is -1. Vendor is Xilinx.



**Figure 11: Result showing for 32-bit Vedic Multiplier**

**Interpretation:** Above figure 11 shows 32-bit Vedic Multiplier where  $i/p a=32'd15$ ,  $b=32'd15$  and  $o/p c=225$ . The above zed board kit contains only 8 switches so, towards output 'c' we see only LSB bits of  $c7-c0(11100001)$ .



**Figure 12: Result showing for 32-bit Square Module**

**Interpretation:** Above figure 12 shows 32-bit Square Architecture where  $i/p x=32'd15$  and  $o/p y=225$ . The above zed board kit contains only 8 switches so, towards output 'y' we see only LSB bits of  $y7-y0(11100001)$ .

**IV. RESULT ANALYSIS**

The following are Proposed Vedic Multiplier and Square Module results compared with previous results.

**Table 1: Comparison Table**

| Parameter | Proposed Method  |               | Conventional Method |               |
|-----------|------------------|---------------|---------------------|---------------|
|           | Vedic Multiplier | Square Module | Vedic Multiplier    | Square Module |
| Delay     | 34.240ns         | 34.794ns      | 41.562ns            | 36.979ns      |
| Speed     | more             | More          | less                | less          |

The result shows that better performance in proposed Vedic Multiplier and Squaring architecture in various aspects like delay, speed. Therefore the proposed method shows the better results compared to conventional method.

**Advantages**

- Improvement in delay.
- To acquire good efficiency
- Vedic multiplier is faster than the array multiplier and booth multiplier.
- Increases Speed.

**Applications**

- Digital Signal Processing(DSP)
- Discrete Fourier Transform(DFT)
- Image Processing

**V. CONCLUSION**

The proposed Vedic Multiplier and Squaring Architectures has been designed by using Vedic mathematics sutras for DSP application. And Simulated in Vivado tool and Implemented on Zynq board. As the number of bit size increases these architectures offer great improvement in delay. And improves the speed.

#### Future Scope

Future scope of the work is to increase the bits, improvement in delay, analysis of power consumption and area.

#### REFERENCES

- [1] Jagadguru Swami Sri Bharati Krishna Tirtha Ji Maharaja, "Vedic Mathematics", Motilal Banarsidas, Varanasi, India, 1986.
- [2] G. Vaithiyanathan, K. Venkatesan, S. Sivaramakrishnan, S. Siva, S. Jayakumar, "Simulation And Implementation of Vedic multiplier using VHDL code", International Journal of Scientific & Engineering Research, Vol. 4, Issue 1, January 2013.
- [3] Dillip Kumar, Abhijeet Kumar, Siddhi, "Hardware Implementation of 16\*16 bit Multiplier and Square using Vedic Mathematics", International Conference on Signal, Image and Video Processing (ICSIVP), 2012.
- [4] Prof J M Rudagi, Vishwanath Ambli, Vishwanath Munavali, Ravindra Patil, Vinaykumar Sajjan, "Design and Implementation of Efficient Multiplier Using Vedic Mathematics", IET Proc. Of Conf. on Advances in Recent Technologies in Communication and Computing, pp. 162- 166, 2011.
- [5] L. Sriraman, K. Saravana Kumar, T.N. Prabakar, "Design and FPGA Implementation of Binary Squarer Using Vedic Mathematics", IEEE ICCNT, July 4-6, 2013.