

32-BIT VEDIC MULTIPLIER

# INTRODUCTION

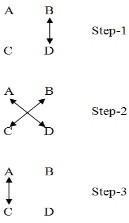
The word 'Vedic' was driven from the word 'Veda' which is ancient store-house of all knowledge. Vedic mathematics provides the solution to the problem of long computation time by reducing the time delay needed for the operations to be performed. It has originated from “Atharva Vedas” the fourth Veda. Atharva Veda mainly deals with the branches like engineering, mathematics, sculpture, medicines and all other sciences.. All the advantages drive 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.

# DESIGN APPROACH

* 1. **VEDIC MULTIPLIER ARCHITECTURE**

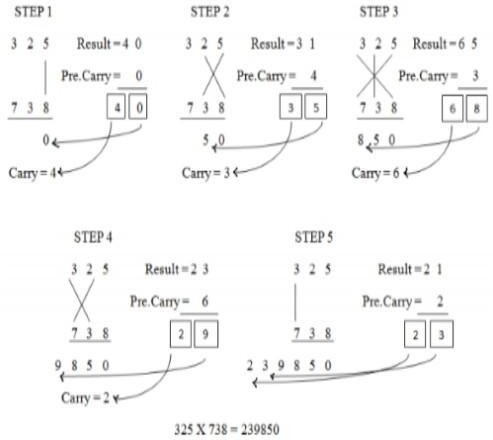
This section introduces multiplication operation using Vedic IXI Methodology and then illustrates architecture of 2x2 multiplier module and finally architecture of n-bit multiplier. Urdhava- Tiryakhayam sutra has been used for multiplication purpose. The figure 1 explains multiplication of two decimal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

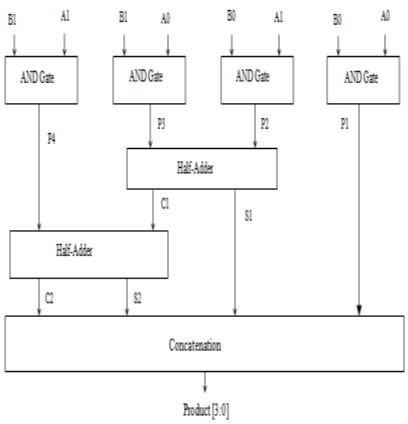


# Figure 2: Example for UT sutra

## 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

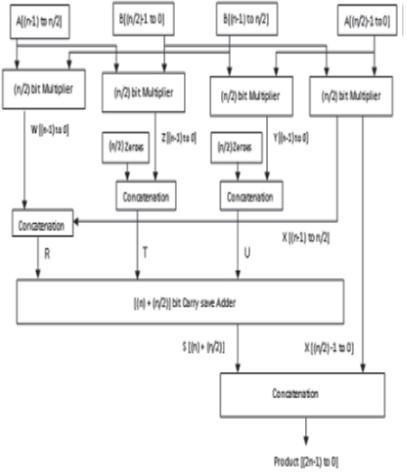
It consists of four AND gates used for AND operation of single bit numbers, two half adders for the addition of numbers obtained in the previous section to provide the sum and carry. At the end concatenation of all the final results is done to obtain the final product of 4-bits.

This architecture is basically implementation of IXI 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

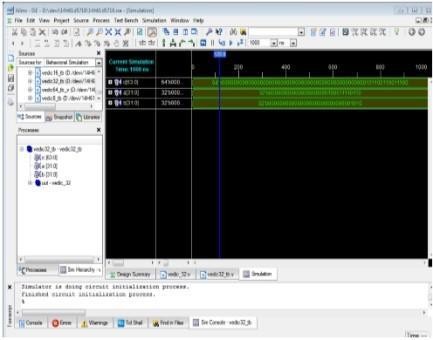


most multiplier to get the required final product.

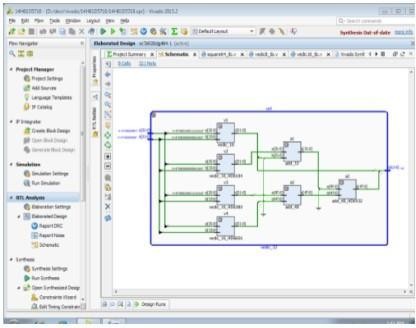
# TEST RESULTS

**A. Simulation Results**

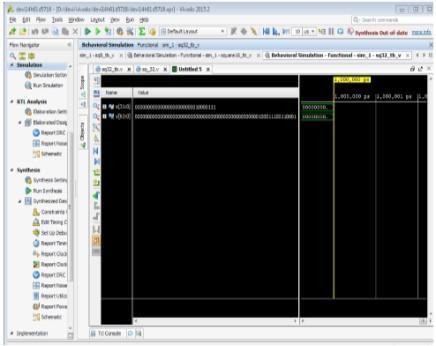
## i. Vedic Multiplier

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

# 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 Vivado Tool.

# 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.



# RESULT ANALYSIS

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

# Table1: Comparision Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Paramete r** | **Proposed Method** | | **Conventional**  **Method** | |
| **Vedic Multiplie r** | **Squar e Modul**  **e** | **Vedic Multiplie r** | **Squar e Modul**  **e** |
| **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



The proposed Vedic Multiplier and Squaring Architectures has been designed by using Vedic mathematics sutras for DSP application. And Simulated in Vivado tool.

# 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,“VedicMathematics”,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. Dilip 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, VishwanathAmbli, VishwanathMunavali,RavindraPatil,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 ICCCNT, July 4-6, 2013.