© 2018 IJSRSET | Volume 4 | Issue 9 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

# Implementation of High Speed 16x16 Vedic Multiplier using Verilog HDL Coding Technique

**Choksi vandana M.**
Department of E&C, FoT, Dharmsinh Desai University, Nadiad, Gujarat, India

## ABSTRACT

Multiplication is an operation much needed in Digital Signal Processing for various applications. This paper puts forward a high speed Vedic multiplier which is efficient in terms of speed, making use of Urdhva Tiryagbhyam, a sutra from Vedic Math for multiplication and MAC unit for performing addition of partial products and also compares it with the characteristics of existing algorithms. It enables parallel generation of intermediate products, eliminates unwanted multiplication steps with zeros and scaled to higher bit levels using Karatsuba algorithm with the compatibility to different data types. The below algorithms aids to parallel generation of partial products and faster carry generation respectively, leading to better performance. The code is written in Verilog HDL and implemented on ModelSim.

**Keywords:** Vedic mathematics, Urdhva triyakbhyam sutra, Vedic Multiplier

## I. INTRODUCTION

Multiplication is an important fundamental function in arithmetic operations. Multiplication-based operations such as Multiply and Accumulate(MAC) and inner product are among some of the frequently used Computation - Intensive Arithmetic Functions(CIAF) currently implemented in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform(FFT), filtering and in microprocessors in its arithmetic and logic unit. Since multiplication dominates the execution time of most DSP algorithms, so there is a need of high speed multiplier. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip.

The demand for high speed processing has been increasing as a result of expanding computer and signal processing applications. Higher throughput arithmetic operations are important to achieve the desired performance in many real-time signal and image processing applications. One of the key arithmetic operations in such applications is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. Reducing the time delay and power consumption are very essential requirements for many applications. This work presents different multiplier architectures. Multiplier based on Vedic Mathematics is one of the fast and low power multiplier. Minimizing power consumption for digital systems involves optimization at all levels of the design. This optimization includes the technology used to implement the digital circuits, the circuit style and topology, the architecture for implementing the circuits and at the highest level the algorithms that are being implemented. Digital multipliers are the most commonly used components in any digital circuit design. They are fast, reliable and efficient components that are utilized to implement any operation. Depending upon the arrangement of the components, there are different types of multipliers available. Particular multiplier architecture is chosen based on the application.

In many DSP algorithms, the multiplier lies in the critical delay path and ultimately determines the performance of algorithm. The speed of multiplication operation is of great importance in DSP as well as in general processor. In the past multiplication was implemented generally with a sequence of addition, subtraction and shift operations. There have been many algorithms proposals in literature to perform multiplication, each offering different advantages and having tradeoff in terms of speed, circuit complexity, power consumption and area.

The multiplier is a fairly large block of a computing system. The amount of circuitry involved is directly proportional to the square of its resolution i.e. A multiplier of size n bits has n2 gates. For multiplication algorithms performed in DSP applications latency and throughput are the two major concerns from delay perspective. Latency is the real delay of computing a function, a measure of how long the inputs to a device are stable is the final result available on outputs. Throughput is the measure of how many multiplications can be performed in a given period of time; multiplier is not only a high delay block but also a major source of power dissipation. That's why if one also aims to minimize power consumption, it is of great interest to reduce the delay by using various delay optimizations.
Digital multipliers are the core components of all the digital signal processors (DSPs) and the speed of the DSP is largely determined by the speed of its multipliers. Two most common multiplication algorithms followed in the digital hardware are array multiplication algorithm and Booth multiplication algorithm. The computation time taken by the array multiplier is comparatively less because the partial products are calculated 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. Booth multiplication is another important multiplication algorithm. 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 recorder adder stages. Thus, a large propagation delay is associated with this case. 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, Urdhva tiryakbhyam Sutra is first applied to the binary number system and is used to develop digital multiplier architecture. This is shown to be very similar to the popular array multiplier architecture. This Sutra also shows the effectiveness of to reduce the NXN multiplier structure into an efficient 4X4 multiplier structures. Nikhilam Sutra is shown to be much more efficient in the multiplication of large numbers as it reduces the multiplication of two large numbers to that of two smaller ones. The proposed multiplication algorithm is then illustrated to show its computational efficiency by taking an example of reducing a 16X16-bit multiplication to a single 2X2-bit multiplication operation. This work presents a systematic design methodology for fast and area efficient digit multiplier based on Vedic mathematics .The Multiplier Architecture is based on the Vertical and Crosswise algorithm of ancient Indian Vedic Mathematics.

## II. METHODOLOGY

The word "Vedic" is derived from the word "Veda" which means the store-house of all knowledge. Vedic mathematics is mainly based on 16 Sutras (or aphorisms) dealing with various branches of mathematics like arithmetic, algebra, geometry etc. These Sutras along with their brief meanings are enlisted below alphabetically.

1) (Anurupye) Shunyamanyat – If one is in ratio, the other is zero.
2) Chalana-Kalanabyham – Differences and Similarities.
3) Ekadhikina Purvena – By one more than the previous One.
4) Ekanyunena Purvena – By one less than the previous one.
5) Gunakasamuchyah – The factors of the sum is equal to the sum of the factors.
6) Gunitasamuchyah – The product of the sum is equal to the sum of the product.
7) Nikhilam Navatashcaramam Dashatah – All from 9 and last from 10.
8) Paraavartya Yojayet – Transpose and adjust.
9) Puranapuranabyham – By the completion or noncompletion.
10) Sankalana- vyavakalanabhyam – By addition and by subtraction.
11) Shesanyankena Charamena – The remainders by the last digit.
12) Shunyam Saamyasamuccaye – When the sum is the same that sum is zero.
13) Sopaantyadvayamantyam – The ultimate and twice the penultimate.
14) Urdhva-tiryagbhyam – Vertically and crosswise.
15) Vyashtisamanstih – Part and Whole.
16) Yaavadunam – Whatever the extent of its deficiency.

The beauty of Vedic mathematics lies in the fact that it reduces the otherwise cumbersome-looking calculations in conventional mathematics to a very simple one. This is so because the Vedic formulae are claimed to be based on the natural principles on which the human mind works. This is a very interesting field and presents some effective algorithms which can be applied to various branches of engineering such as computing and digital signal processing.

The multiplier architecture can be generally classified into three categories. First is the serial multiplier which emphasizes on hardware and minimum amount of chip area. Second is parallel multiplier (array and tree) which carries out high speed mathematical operations. But the drawback is the relatively larger chip area consumption. Third is serial - parallel multiplier which serves as a good trade-off between the times consuming serial multiplier and the area consuming parallel multipliers.

**2.1 Urdhva Tiryagbhyam:**

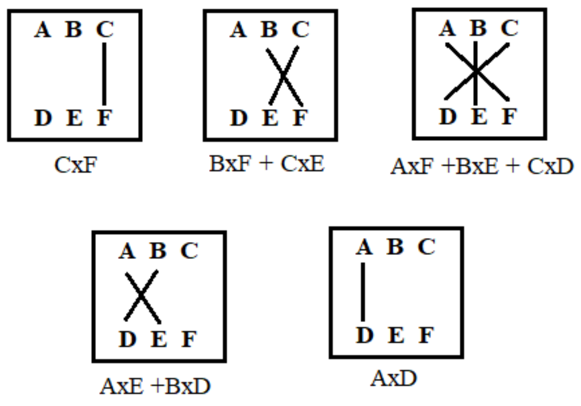

**Figure 1.** Urdhva Tiryagbhyam Sutra

Consider ABC as multiplicand and DEF as the multiplier. The steps of multiplication are descriptive in the figure 1 above for better understanding. The intermediate carry generated is appended to the very next bit.

## III. PRIOR ARTWORK

Urdhva tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means 'Vertically and crosswise'. Design 2X2 Multiplier using this multiplication scheme.

### 3.1. 2X2 Multiplier:

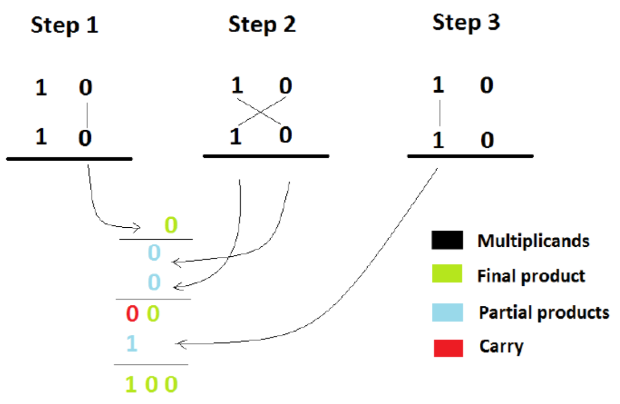

**Figure 2.** 2x2 multiplier

Figure 2 illustrates the steps to multiply two 2 bit numbers. Converting the above figure 2 to a hardware equivalent we have 3 and gates which will act as 2 bit multipliers and two half adders to add the products to get the final product. Here is the hardware detail of the multiplier shown in figure 3.

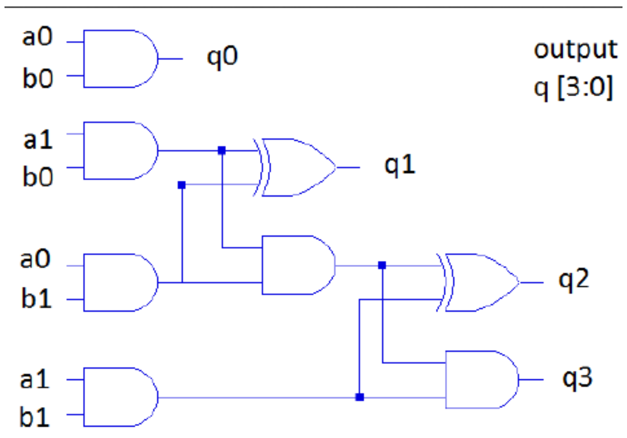

**Figure 3.** Hardware Implementation of 2x2 Multiplier

Where "a" and "b" are two numbers to be multiplied and "q" is the product. To make the design more modular we write the code for HA (Half Adder) first and then instantiate it to have the final product.

### 3.2. 4X4 Multiplier:

Using 4 such 2x2 multipliers and 3 adders we can build 4x4 bit multipliers as shown in the figure 4. First, we write code for 4-bit and 6-bit adders. It's our choice to choose our adders. If in case we want to have better performance you can replace these normal adders with compressors. For a simpler design we have used the "+" operator which is supported by the tool which by default selects a low hardware adder. This architecture follows Wallace tree which reduces the addition levels from 3 to just 2 stages as shown in figure 5. Arrangement of the adders and the addition is explained in the figure 4.

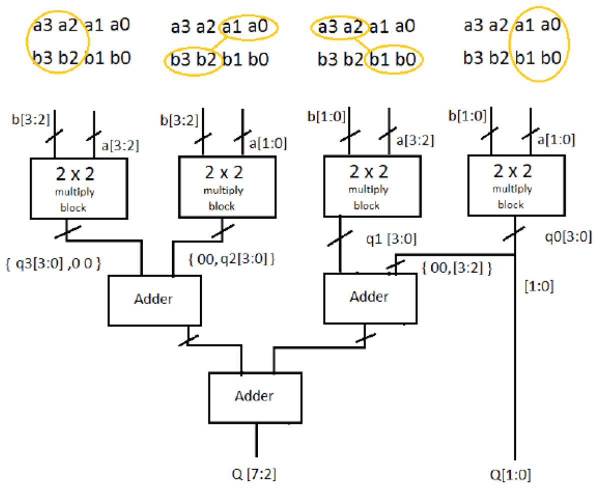

**Figure 4.** Arrangement of the adders and the addition for 4x4 multiplier

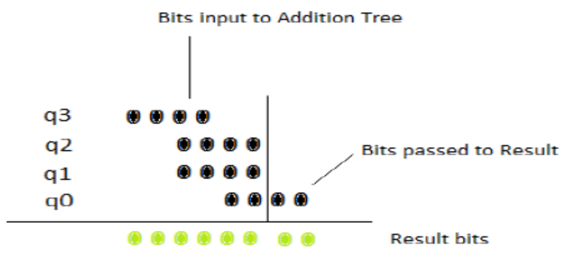

**Figure 5.** Architecture follows Wallace tree

### 3.3. 8X8 Multiplier:

Similar to the design of 4x4 multiplier, we need 4 such 4x4 multipliers to develop 8x8 multipliers. Here we need to first design 8bit and 12 bit adders and by proper instantiating of the module and connections as shown in the figure 6. Refer the addition tree diagram to know the process of 8x8 multiplier shown in figure 7.

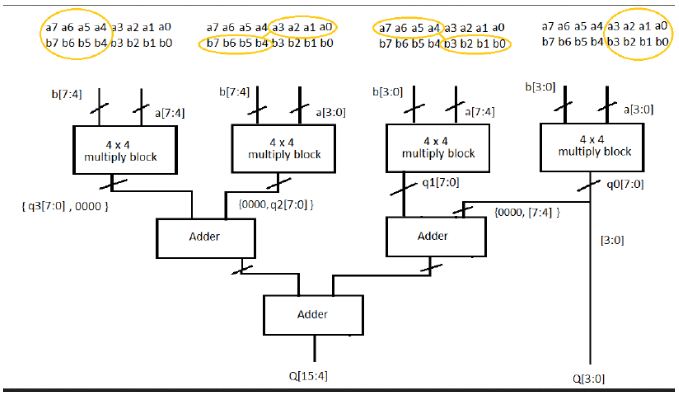

**Figure 6.** Arrangement of the adders and the addition for 8x8 multiplier

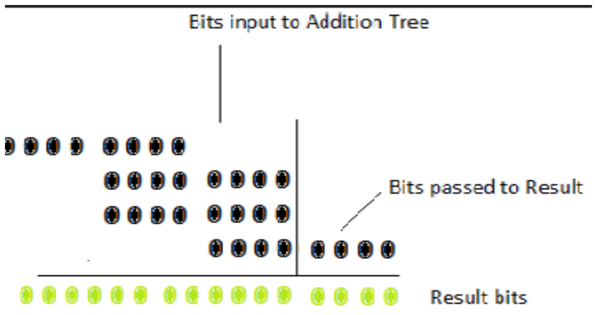

**Figure 7.** Architecture follows Wallace tree

### 3.4. 16X16 Multiplier:

Follow the same steps as in case of previous multipliers and develop 16x16 multipliers. Refer the adder tree diagram shown in figure 9.

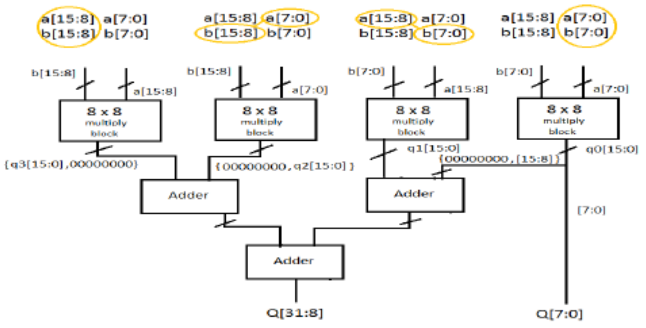

**Figure 8.** Arrangement of the adders

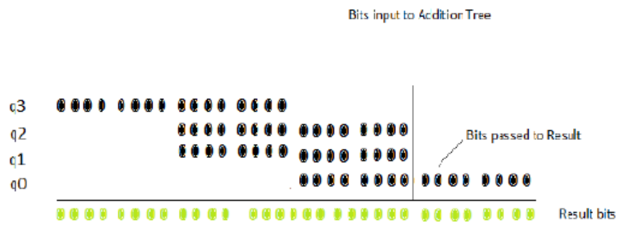

**Figure 9.**Adder Tree Diagram

### 3.5. MAC Design:

To being with MAC design first we design an accumulator which adds two number. One of which is the output of the previous stage and the other is the output from the multiplier module. Figure 10 below shows the implementation design for MAC.

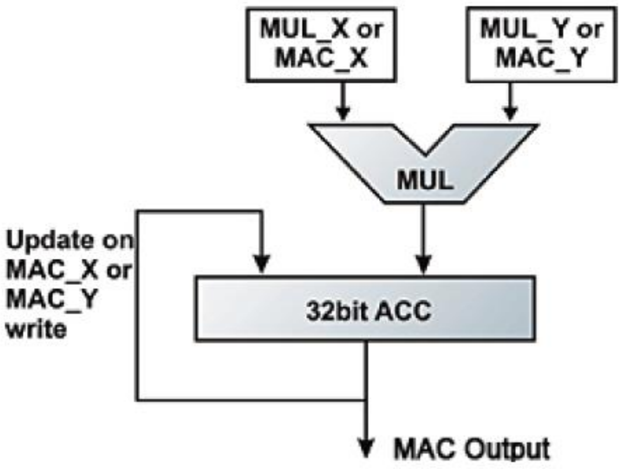

**Figure 10.** Implementation design for MAC

It can be seen from the block diagram that the accumulator module has one input (we have designed this module be synchronous so we have used Clock as second input). Few more control signals are required to clear the ACC unit and enable signal to initiate the process of accumulation.

## IV. RESULT

The results are as our expectation shown in figures.

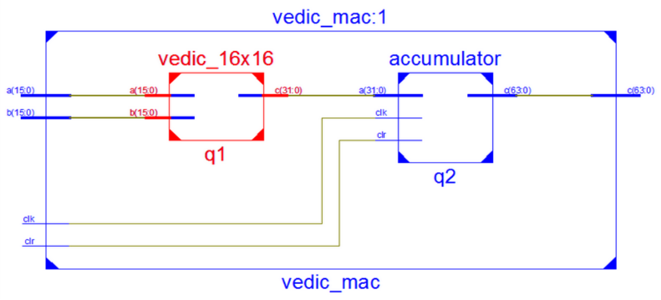

**Figure 11.** RTL view of 16X16 Multiplier

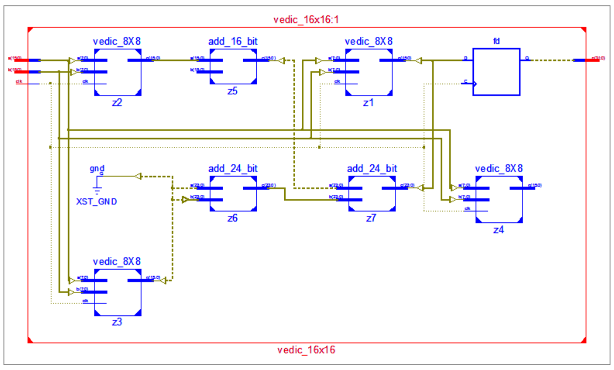

**Figure 12.** RTL view of Vedic calculator

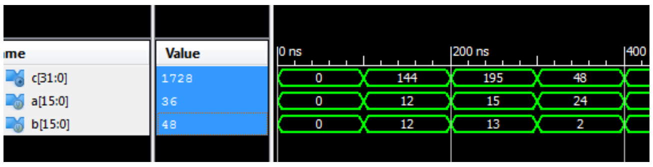

**Figure 13.** Simulation Window

## V. HOW IS IT BETTER THAN THE CONVENTIONAL METHOD?

In conventional method, partial products are summated only after every partial product is obtained. Whereas, in Vedic technique, partial products are obtained vertically as shown in the figure 14 and simultaneously once all the elements of a column are obtained, respective partial products are added. Hence, leads to advancement in speed over the conventional method.

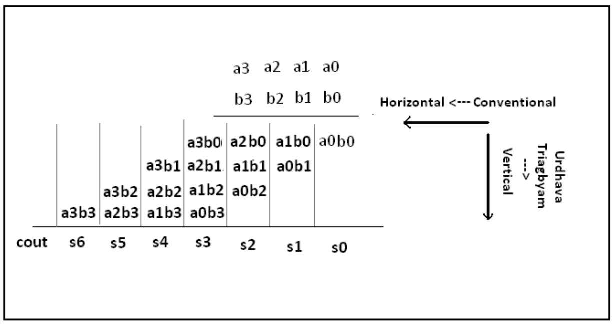

**Figure 14.** Difference between Conventional Multiplication and Vedic technique

## VI. CONCLUSION & FUTURE ASPECTS

As conclusion it can be said that, designs of 16x16 bits Vedic multiplier have been simulated inn ModelSim. The design is based on Vedic method of multiplication. The worst case propagation delay in the Optimized Vedic multiplier case is 31.526ns. It is therefore seen that the Vedic multipliers are much faster than the conventional multipliers. This gives us method for hierarchical multiplier design. So the design complexity gets reduced for inputs of large no of bits and modularity gets increased. Urdhva tiryakbhyam, Nikhilam and Anurupye sutras are such algorithms which can reduce the delay, power and hardware requirements for multiplication of numbers. FPGA implementation of this multiplier shows that hardware realization of the Vedic mathematics algorithms is easily possible. The high speed multiplier algorithm exhibits improved efficiency in terms of speed.

## VII. REFERENCES

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

[2]. Ramchandran.S and Kruthi.S.Pande. May-June 2012. "Design, Implementation and Performance Analysis of an Integrated Vedic Multiplier Architecture." Amrita college of engineering, Bangalore.

[3]. Samir Palnitkar. "Verilog HDL, A guide to Digital Design and Synthesis." 2nd edition. Dorling Kindersley and Pearson Education, Inc. 2008.

[4]. Poornima M., Shivaraj Kumar Patil, Shivukumar, Shridhar K P and Sanjay H. May, 2013. "Implementation of Multiplier using Vedic Algorithm". MVJCE, Bangalore.

[5]. PushpalataVerma and K.K.Mehta. June 2012. "Implementation of an efficient Multiplier based on Vedic Mathematics using EDA tool."

[6]. Peter M. Kogge and Harold S. Stone. August 1973. "A Parallel algorithm for the efficient

solution of a general class of recurrence equations."

[7]. Pavan Kumar U.C.S, Saiprasad Goud A and A.Radhika. March 2013. "FPGA Implementation of High Speed 8-bit Vedic Multiplier Using Barrel Shifter."

[8]. Neil H.E. Weste and Kamran Eshraghian. "Principles of CMOS VLSI Design." 3rd edition. Addision-Wesley Reading, MA.

[9]. Pakkiraiah Chakali, Madhu Kumar Patnala. February 2013. "Design of High Speed Kogge-Stone Based Carry Select Adder."

[10]. P.Annapurna Bai, M.Vijaya Laxmi. August 2013. "Design of 128-bit Kogge-Stone Low Power Parallel Prefix VLSI Adder for High Speed Arithmetic Circuits."

[11]. Abhijith Kini G. September 2011. "Asynchronous Hybrid Kogge-Stone Structure Carry Select Adder Based IEEE-754 Double-Precision Floating-Point Adder." Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, NITK-Surathkal. Surathkal, Karnataka 575025, India.