

# FPGA IMPLEMENTATION OF KARATSUBA VEDIC MULTIPLIERS

Shankar R<sup>1</sup>, Sundhararajan G<sup>2</sup>, Vignesh S<sup>3</sup>, Aravind AR<sup>4</sup>

<sup>1</sup>UGStudent, Electronics and Communication Engineering, Prince Shri Venkateshwara Padmavathy Engineering College, Tamil Nadu, India

<sup>2</sup>UGStudent, Electronics and Communication Engineering, Prince Shri Venkateshwara Padmavathy Engineering College, Tamil Nadu, India

<sup>3</sup>UGStudent, Electronics and Communication Engineering, Prince Shri Venkateshwara Padmavathy Engineering College, Tamil Nadu, India

<sup>4</sup>Associate Professor, Electronics and Communication Engineering, Prince Shri Venkateshwara Padmavathy Engineering College, Tamil Nadu, India

## ABSTRACT

An area efficient multiplier design is conferred. This design is based on Karatsuba algorithm founded by Anatoly Karatsuba and ancient algorithm of Vedas, proposed within the Vedic mathematics. Multiplication is a very important mathematical operation to be performed with high speed and less power consumption in high-speed systems like Digital Signal Processing. The Karatsuba algorithm speed up the multiplication of large numbers by splitting the operands into two parts of equal length. In the Vedic multipliers the partial product generation and the also the sums are generated in few steps which reduces the carry propagation from LSB to MSB. In this proposed design the area and the delay is reduced. Finally, the results of Karatsuba Vedic multiplier is compared with Vedic multiplier.

**Keyword:** Karatsuba Multiplier, Vedic multiplier, Urdhva Triyakbhyam sutra, area efficient architecture, array multiplier

## 1. INTRODUCTION

The general procedure for multiplying two n-digit numbers requires a number of elementary operations proportional to  $n^2$ , or  $o(n^2)$  in bi-O notation. Andrey Kolmogorov conjectured that it would require  $n^2$  elementary operations for any algorithm for that task. Karatsuba algorithm reduces the multiplication of two n-digit numbers at most  $n^{\log_2 3} \cong n^{1.58}$  single-digit multiplications in. It is therefore faster than the conventional algorithm.

Array multiplier consumes a lot of power and occupies more area because of many full adders. The delay of array multiplier is high as one full adder has to wait until the carry is propagated from the full adder to next full adder. One way to improve the speed as well as to reduce the power is to employ the ancient methods of mathematics known as Vedic mathematics. The Vedic in Sanskrit means store-house of knowledge. The Vedic mathematics offers shortcuts to quickly perform the arithmetic operations. There is a total of 16 Vedic sutras or formulas and 13 sub-sutras to perform different mathematical operation. For performing the multiplication operation there are two sutras- urdhva tiryakbhyam and nikhilam sutra. Out of this two, the urdhva tiryakbhyam sutra is discussed in this paper to design a multiplier.

The paper is organized as follows. Section 2, demonstrate the algorithm of Karatsuba multiplier. Section 3 talks about the algorithm of Vedic multiplier. The Karatsuba-Vedic architecture is shown in Section 4. The results and comparisons are included in section 5, while section 6 contains conclusion.

## 2. KARATSUBA MULTIPLIER

Karatsuba multiplication algorithm was introduced by Anatoli Karatsuba in 1962. It is best suited for multiplication of large numbers. It is a divide and conquer method, in which the numbers are divided into their Most Significant half and Least Significant half and then multiplication is performed. The multiplication

operation is replaced by addition operation in Karatsuba algorithm hence reducing the number of multipliers used. This indeed increases the speed of operation as addition is faster than multiplication. algorithm becomes more efficient when the number of bits increases. It is suitable for multiplication operations of 16 bits and more.

Product = X.Y

X and Y can be written as,

$$X = 2^{\frac{n}{2}}.X_l + X_r \tag{2.1}$$

$$Y = 2^{\frac{n}{2}}.Y_l + Y_r \tag{2.2}$$

Where  $X_l, Y_l$  and  $X_r, Y_r$  are the Most Significant half and Least Significant half of X and Y respectively, and n is the number of bits.

Then,

$$\begin{aligned} XY &= (2^{\frac{n}{2}}.X_l + X_r). (2^{\frac{n}{2}}.Y_l + Y_r) \\ &= 2^n.X_lY_l + (2^{\frac{n}{2}}.X_lY_r + X_rY_l) + X_rY_r \end{aligned} \tag{2.3}$$

The Second term in equation (2) can be optimized to reduce the number of multiplication operations. i.e.

$$X_lY_r + X_rY_l = (X_l + X_r).(Y_l + Y_r) - X_lY_l - X_rY_r \tag{2.4}$$

The equation (2) can be re-written as,

$$X.Y = 2^n.X_lY_l + X_rY_r + 2^{\frac{n}{2}}((X_l + X_r).(Y_l + Y_r) - X_lY_l - X_rY_r) \tag{2.5}$$

This formula requires only four multiplications and observed that at the cost of a few extra additions, X.Y could be found with only three multiplications. Figure 1 shows the block diagram of Karatsuba multiplier.

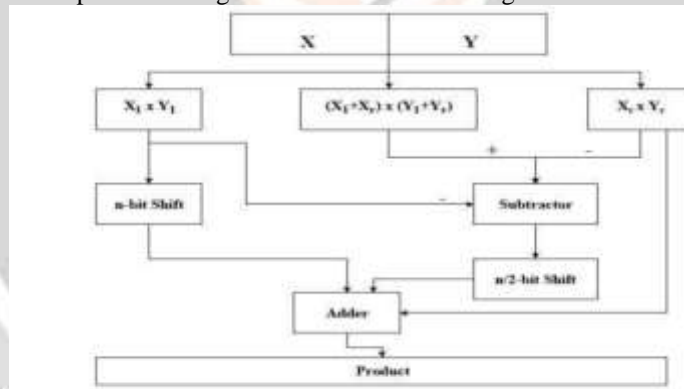


Fig - 1 Block diagram of Karatsuba multiplier

### 3. VEDIC MULTIPLIER

#### 3.1 2X2 bit Vedic Multiplier

Consider two 2-bit numbers A=a1a0 and B=b1b0. Initially, the least significant bits (LSB) a0 and b0 is multiplied which gives the LSB (s0) of the final product. Then, the LSB of the multiplicand a0 is multiplied with the next higher bit of the multiplicand b1. The result gives second bit s1 of the final product and carry c1 which was generated gets added with the partial product obtained by multiplying the most significant bits a1 and b1 to give the sum s2 and carry c2. The sum s2 is the third bit and carry c2 becomes the fourth bit of the final product.

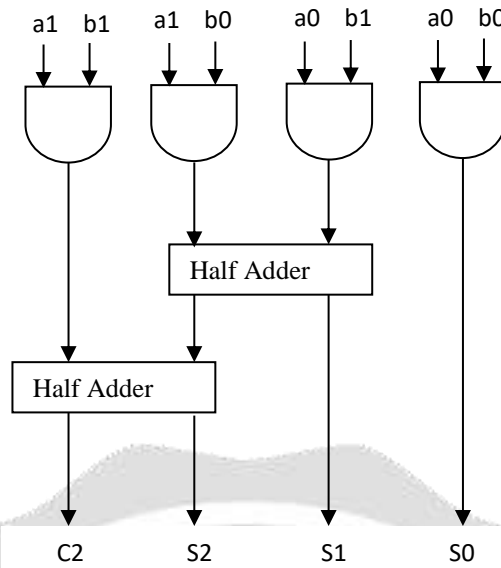


Fig - 2 Block diagram of 2x2 Vedic Multiplier

**3.2 4X4 bit Vedic Multiplier**

Consider two 4-bit binary numbers  $a_3a_2a_1a_0$  and  $b_3b_2b_1b_0$ . Divide the multiplicand into two parts each consisting of two bits as  $a_3a_2$  and  $a_1a_0$ . similarly, divide the multiplier into two parts as  $b_3b_2$  and  $b_1b_0$ . Taking two bits at a time and using 2 x 2-bit Vedic multiplier, the architectures for 4x4 bit Vedic multiplier using four 2x2 bit Vedic multipliers are shown in figure 3. The final product is  $s_7s_6s_5s_4s_3s_2s_1s_0$ .

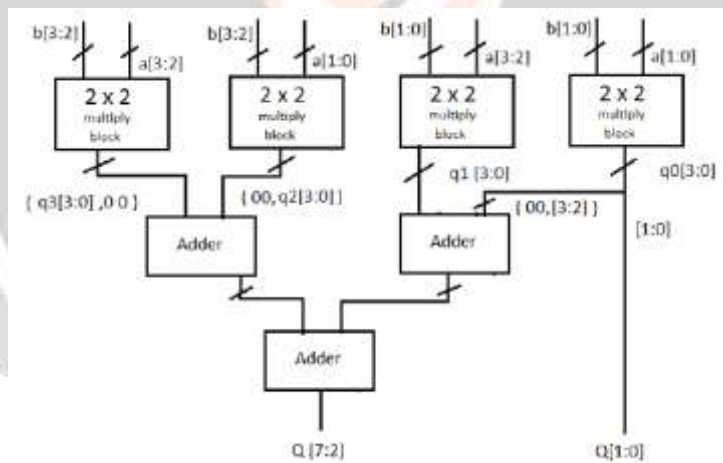


Fig - 3 Block diagram of 4x4 Vedic Multiplier

**4. KARATSUBA-VEDIC ARCHITECTURE**

The Karatsuba algorithm is suitable for computing the higher order bits by divide and conquer method. The Vedic multiplier is suitable for multiplying bits less than 16. So, a suitable multiplier is proposed by efficiently using these two methods. A higher order bit is first reduced into fewer lower order bits by using the Karatsuba algorithm and the lower order bits are multiplied by using Vedic multipliers. With this Multiplier the effective area and the delay is reduced.

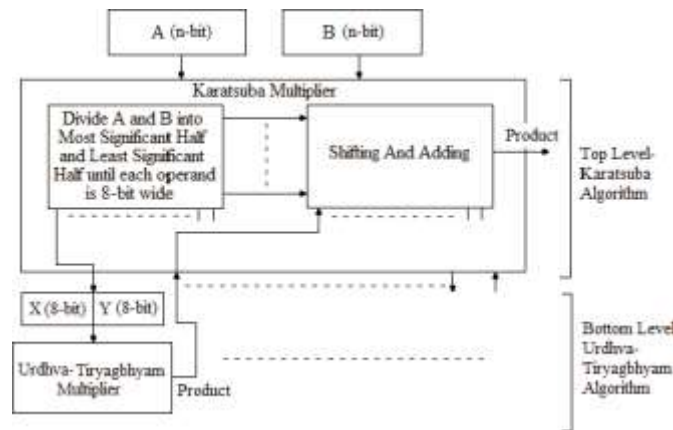


Fig - 4 Block diagram of Karatsuba -Vedic multiplier

5. SIMULATION RESULTS

The basic 8x8 bit Karatsuba - Vedic multiplier is implemented in Xilinx Verilog 9.1. The simulation result of two numbers (211x149), (255x255) and (128,126) is shown in the figure 5. The Proposed system and Vedic multiplier are compared in terms of number of slices and delay. From the table 1, there is a decrease in the area and delay of the proposed system.

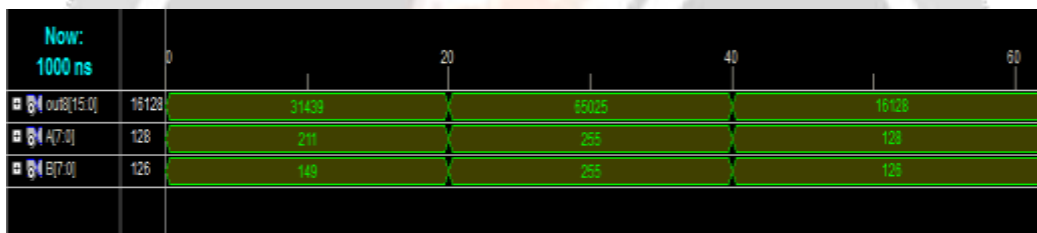


Fig - 5 Simulation of 8x8 Proposed multiplier

The below table represent the comparison of Karatsuba-Vedic and Vedic multiplication.

Table - 1 Comparison of Vedic and Karatsuba-Vedic multipliers

Parameters	Proposed System	8-bit Vedic	Percentage improvement
Path Delay	22.475ns	23.18ns	3.05 (decrease)
Number of 4 input (LUTs)	154 out of 9312 1%	497 out of 9312 5%	69.10 (decrease)
Number of bonded IOBs	32 out of 190 16%	34 out of 190 16%	5.89 (decrease)

6. CONCLUSIONS

We report on a unique multiplier design based on Karatsuba algorithm combined with formulas of the ancient Indian Vedic mathematics which is extremely suitable for High Power Computing (HPC), Image processing and Signal processing which is based on binary multiplication. The effective delay and the area of the circuit is thus reduced by the combination of Karatsuba-Vedic multipliers.

7. REFERENCES

[1]. Anand Mehta, C. B. Bidhul, Sajeevan Joseph, Jayakrishnan. P, "Implementation of Single Precision Floating Point Multiplier using Karatsuba Algorithm", International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), pp.254-256, 2013.

- [2]. Arish S, R.K.Sharma, “An efficient binary multiplier design for high speed applications using Karatsuba algorithm and Urdhva-Tiryagbhyam algorithm”, Global Conference on Communication Technologies (GCCT 2015) ,pp.192-196, 2015.
- [3]. Bathija R. K, R.S. Meena, S. Sarkar, Rajesh Sahu, “Low Power High Speed 16x16 bit Multiplier using Vedic Mathematics”, International Journal of Computer Applications (0975 – 8887), Volume 59– No.6, pp. 41-44, December 2012.
- [4]. Chow, G. C. T., Eguro K, Luk W, & Leong P, “A Karatsuba-based Montgomery Multiplier”, International Conference on Field Programmable Logic and Applications, 2010.
- [5]. Ganesh Kumar G, V. Charisma, “Design of High-Speed Vedic Multiplier using Vedic Mathematics Techniques”, International Journal of Scientific and Research Publications, Volume 2, Issue 3, 2012.

