

# High Speed Efficient Karatsuba- of man Pipelined Multiplier for Low Contrast Image Enhancement

Triveni.K.S, E. Elavarasi

**Abstract:** Multiplication is one of the supreme operation in many high performance systems such as microprocessor, FIR filters, Digital Signal Processor, Image Processing etc. Since multiplication dominates the execution time of most operations, there is a huge demand in increasing the speed of the multipliers, which has been subject of interest over years. . Multiplier based on Vedic Mathematics is one of the fast and low power multiplier. This proposed Karatsuba-Ofman Pipelined Multiplier (KOPM) is designed mainly to optimize speed of the multiplier which is the major requirement in many applications. Karatsuba-Ofman algorithm [3] is employed to optimize the speed. Pipelining will enhance the performance of the multiplier. The multiplier is implemented for the low contrast image enhancement. This multiplier has been synthesized on Spartan3E.

**Keywords:** Karatsuba-Ofman, Low contrast image, Pipelining, Vedic Mathematics.

## I. INTRODUCTION

Multipliers are the basic and essential building blocks of many high performance systems. Multiplication is frequently used operation which is currently implemented in many processors. In today’s market there is a huge demand for high speed multipliers, since these are the slowest elements in the systems. The speed of the multiplier decides the speed of the system; hence the speed of the multiplier has to be improved. The performance of the system depends on the multiplier’s speed which is optimized by the proposed multiplier. Vedic mathematics is the ancient system of mathematics. The word ‘Vedic’ is derived from the word ‘Veda’, which means store-house of all knowledge.. Karatsuba Ofman is the fast multiplication algorithm which helps to increase the speed of the multiplier. Pipelining has long been known as efficient technique for optimising the computational time. Significant speed-up in computational time is increased by the application of pipelining. Pipelining comes as a rescue for speeding-up the clock frequency. With the aid of Pipelining and with Karatsuba-Ofman algorithm, will double the speed of the proposed KOPM multiplier. Section II gives a detailed explanation on Karatsuba-Ofman Algorithm. Section III and IV explains the design of blocks of the multiplier. Concept of Pipelining is explained in section V. Application of the multiplier is given in section VI.

## II. KARATSUBA-OFMAN ALGORITHM

Karatsuba-Ofman’s algorithm is one of the fastest methods to multiply long integers. This multiplication algorithm multiplies every digit of a multiplicand by every digit of the multiplier and adds the result to the partial product.

Manuscript published on 30 June 2013.

\* Correspondence Author (s)

Triveni. K. S, Electronics and communication, PG Student (DEC), AMC College of Engineering, VTU, Bangalore, India.

E. Elavarasi, Electronics and communication, Asst. Prof., AMC College of Engineering, VTU,, Bangalore, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

It has  $O(n^2)$  complexity, where  $n$  is the operand size (number of digits).Karatsuba-Ofman algorithm has  $O(n^{1.585})$  complexity and thus it multiplies large numbers faster. Let  $A$  and  $B$  be two  $n$ -digit numbers in radix  $z$  where  $n$  is even. Divide these two numbers into two parts and can be written as,

$$A = A_H Z^{n/2} + A_L$$

$$B = B_H Z^{n/2} + B_L$$

where  $Z=2$  for binary number system,  $A_L$  and  $A_H$  are the lower digits (first  $n/2$ ) and higher digits(last  $n/2$ ) of number  $A$  respectively. Similarly  $B_L$  and  $B_H$  are the lower digits (first  $n/2$ ) and higher digits(last  $n/2$ ) of number  $B$ . The product  $P$  can be calculated as,

$$P = A \cdot B = (A_H Z^{n/2} + A_L) \cdot (B_H Z^{n/2} + B_L)$$

$$P = A_L B_L + (A_L B_H + A_H B_L) Z^{n/2} + (A_H B_H) Z^n \dots \dots \dots (1)$$

Illustration:

Let  $A = 1001$  (9) and

$B = 0010$  (2)

Now,  $A = (10)2^2 + (01)$

$B = (00)2^2 + (10)$

$$P = (01)(01) + [(10)(00)] 2^4 + [(01)(00) + (10)(00)] 2^2$$

$$= 0010 + 00000000 + [0000 + 0100] 2^2$$

$$= 0010 + 00000000 + 010000$$

$$= 00010010$$
 (18)

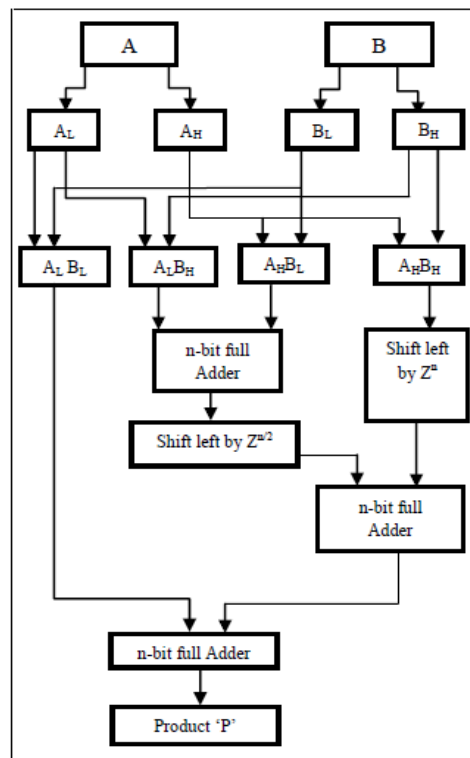


Figure - 1 Karatsuba-Ofman Algorithm



Thus product P can be calculated by solving equation (1) with four partial products:  $A_L B_L$ ,  $A_L B_H$ ,  $A_H B_L$  and  $A_H B_H$ . The block diagram of Karatsuba-Ofman Algorithm is shown in figure 1 which requires three adders.

### III. 2X2 MULTIPLIER

2x2 is the elemental building block of this large bit size multiplier. This fundamental block is designed by using the truth table of 2x2 multiplier as shown in table 1, which can be solved using K-map. This generates a set of equations. 2x2 multiplier can be realized using these equations.

TABLE 1- TRUTH TABLE FOR 2X2

A <sub>1</sub>	A <sub>0</sub>	B <sub>1</sub>	B <sub>0</sub>	P <sub>3</sub>	P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

$P_0 = A_0$  and  $B_0$

$P_1 = A_0$  and  $B_1$  [ (not  $A_1$ ) or (not  $B_0$ )] or  $A_1$  and  $B_0$  [ (not  $A_0$ ) or (not  $B_1$ )]

$P_2 = A_1$  and  $B_1$  [ (not  $A_0$ ) or (not  $B_0$ )]

$P_3 = A_0$  and  $A_1$  and  $B_0$  and  $B_1$

Using these equations 2X2 multiplier has been designed and implemented.

### IV. 4X4 AND HIGHER ORDER MULTIPLIER

Following the design of 2x2 multiplier, 4X4 and higher order multipliers are designed and implemented using Karatsuba-Ofman Algorithm. Because of the use of Karatsuba-Ofman Algorithm in the proposed KOPM multiplier, the usage of memory is reduced as compared with the memory that is required for the normal multiplier. nxn multiplier is decomposed into smaller order multipliers, i.e.,  $[(n/2) \times (n/2)]$ ,  $[(n/4) \times (n/4)]$  and so on up to 4X4.

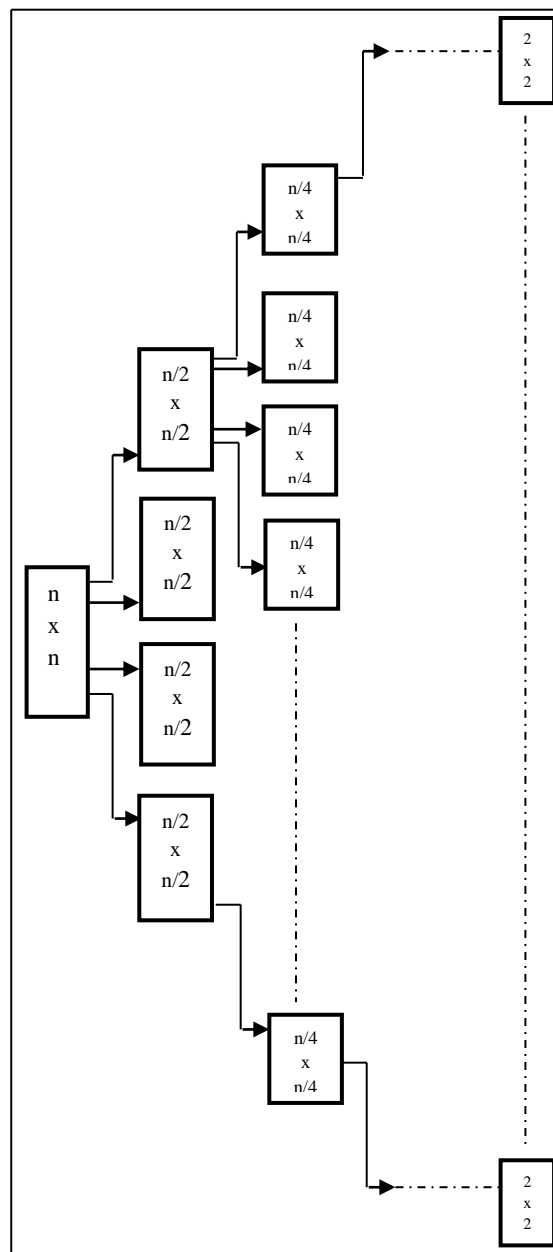


Figure-2- nxn Multiplier Architecture

2x2 multiplier is designed as mentioned in section III. Higher order multipliers are designed using Karatsuba-Ofman Algorithm in the proposed model is shown in fig. 2.

### V. PIPELINING

Pipelining is the possible approach to speed-up the clock frequency. It is an important technique used in several applications such as digital signal processing (DSP) systems, microprocessors, etc. Accordingly, it results in speed enhancement for the critical path in most systems. Pipelining comes from the idea of water pipe: continue sending water without waiting the water in the pipe to be exit. The beauty of pipelining design is that new data can be processing before the processing of the prior data has finished. Pipelining is very similar to the assembly line in auto industry. The performance of the proposed multiplier is advanced by the implementation of pipelined concept. Figure 3 illustrates the pipelined architecture concept which enhances the performance of the multiplier.



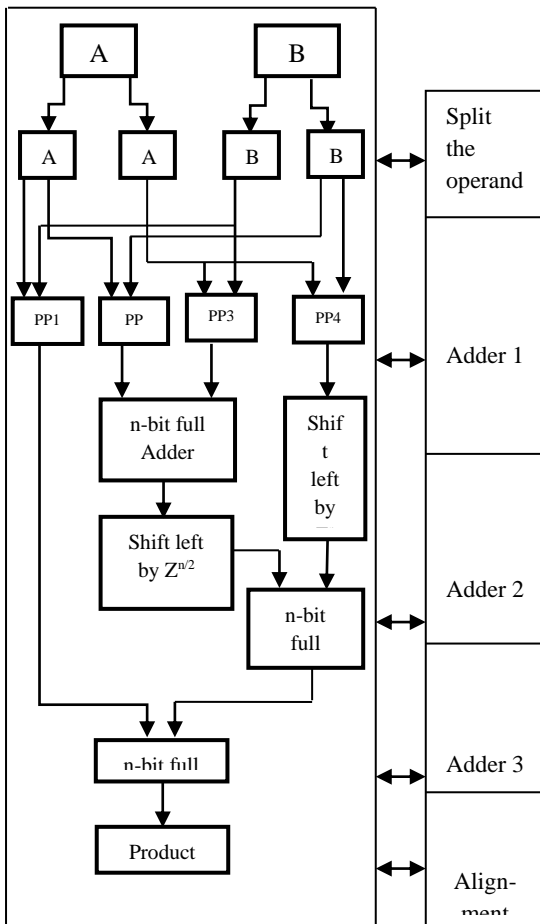


Figure 3 – Pipelined Architecture

As shown in figure 3, the Pipelined architecture is classified into 5 stages, (1) splitting stage, (2) Adder 1 stage, (3) Adder 2 stage, (4) Adder 3 stage and (5) Alignment stage. At each stage one clock period is considered. Speed of the pipelined multiplier is double the speed of the non-pipelined multiplier.

For each operation in pipelined architecture, one clock period is considered. Pipelined multiplier requires 7 clock pluses, whereas the non-pipelined multiplier requires 9 clock pluses. Since the number of clock pluses required by pipelined multiplier is less by 2 clock pluses as compared with non-pipelined multiplier, the speed of the multiplier is enhanced.

VI. APPLICATION OF KOPM ON LOW CONTRAST IMAGES

The proposed Karatsuba-Ofman Pipelined multiplier has been applied for the enhancement of low contrast images. On a monochrome display with eight bits per pixel, each pixel can have any of 256 different grey scale intensities, ranging from 0(black) to 255 (white).

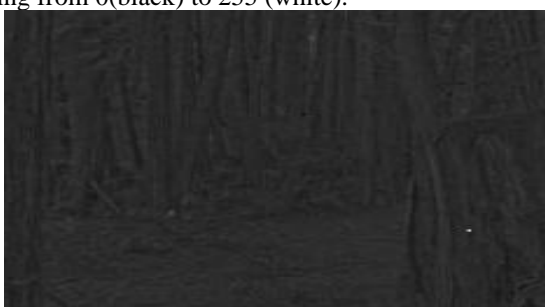


Figure – 4 Low contrast image

Consider the low contrast image shown above in figure – 4. In order to simulate real image data in VHDL it is important to create a method of transferring image to a matrix form. MATLAB is used to do this functionality which is quite efficient in manipulating matrix data. Then the matrix is multiplied with a fixed threshold value by using the proposed Karatsuba Ofman Pipelined multiplier. The obtained matrix is again converted back to image with the help of MATLAB. The enhanced image is shown in fig.-5.

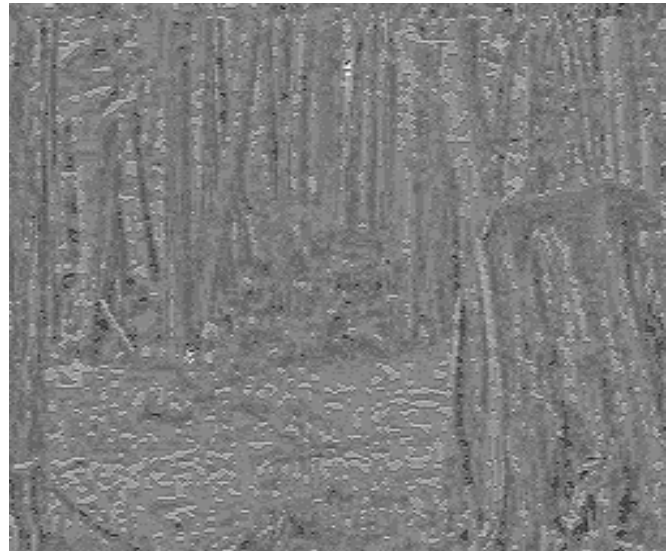


Figure – 5 Enhanced image

VII. CONCLUSION & RESULTS

TABLE – II Comparison Between Pipelined And Non-Pipelined Multiplier

Blocks	Non-Pipelined [1]	Pipelined
4x4	170.366	320.682
8x8	104.404	237.662
16x16	78.90	169.166
32x32	57.521	135.352

Non-Pipelined Multiplier

Table II gives a comparison of clock frequency between pipelined and non-pipelined multiplier. The proposed multiplier has been implemented up to 32x32 stage and the speed is advanced at each stage which shown in table II.

REFERENCES

- [1] Jagadguru Swami, Sri Bharati Krsna Tirthji Maharaja, "Vedic Mathematics", Motilal Banarsidas, Varanasi, India, 1986.
- [2] Dr. K. S. Gurumurthy, M.S. Prahalad, "Fast and Power Efficient 16x16 Array of Array Multiplier using Vedic Multiplication", IMPACT-IEEE, 2010
- [3] Serdar S. Erdem, Cetin K. Koc, "A Less Recursive Variant of Karatsuba-Ofman Algorithm for Multiplying Operands of Size a Power of Two", Computer Arithmetic Proceedings, 16th IEEE Symposium on Digital Object Identifier, 2003, Page(s): 28 – 35.
- [4] Abhijit Asati, Chandrashekar "A High-Speed, Hierarchical 16x16 Array of Array Multiplier Design", IMPACT 2009.



- [5] Gang Zhou, Harald Michalik, and László Hinsenkamp, "Complexity Analysis and Efficient Implementations of Bit Parallel Finite Field Multipliers Based on Karatsuba-Ofman Algorithm on FPGAs", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 7, July 2010
- [6] M. Ramalatha, K. Deena Dayalan, P. Dharani, S. Deborah Priya, "High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques", July 15-17, 2009, ACTEA
- [7] Matthias Fugger, Andreas Dielacher, Ulrich Schmid, "How to Speed-up Fault-Tolerant Clock Generation in VLSI Systems-on-Chip via Pipelining", 2010 European Dependable Computing Conference.
- [8] C.Prema, C.S.Mainkanda babu, "Enhanced high speed modular multiplier using Karatsuba algorithm" , 2013 International Conference on Computer Communication and Informatics (ICCCI - 2013), Jan. 04 – 06, 2013, Coimbatore, INDIA.
- [9] Prof J M Rudagil, Vishwanath Amble, Vishwanath Munavalli, Ravindra Patil, Vinaykumar Sajjan, "Design and Implementation of Efficient Multiplier using Vedic mathematics" *Proc. of Int. Con/, on Advances in Recent Technologies in Communication and Computing 2011*.
- [10] Digital VLSI "A Pipelining & Parallel Processing" Mahdi S habany, Sharif University of Technology.
- [11] Circuit design with VHDL by Pedroni.
- [12] Introduction to Xilinx ISE 8.2i. University of Pennsylvania, Digital Design Laboratory.
- [13] VHDL programming by example by Douglas L.Perry.