Volume 4, Issue 6, June 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Polynomial Multiplication Using Karatsuba and Nikhilam Sutra

Mr. Dharmendra Madke
M.Tech Scholar
(EC DEPARTMENT) P.C.S.T,(R.G.P.V).
Bhopal (M.P.), India

Prof. Sameena Zafar
Associate Professor
(EC DEPARTMENT)P.C.S.T,(R.G.P.V.)
Bhopal (M.P.), India

Abstract – This paper analysed the design of polynomial Karatsuba multiplication using nikhilam sutra of Vedic mathematics. In this we are going to design a polynomial multiplication techniques that have been time to time been modified as per the problem requirement to improve performance. Know a day Polynomial multiplication is a required factor that will improve the efficiency of processors. Addition and multiplication are the hole and sole of any processors. This new method is divided in to two part first split numbers in Karatsuba forms and second is multiply them using nikhilam sutra. In this paper we are going to implement polynomial multiplication using Spartan..... FPGA device. The code were written in VHDL and they were simulated and synthesized using Xilinx ISE 9.1

.

Keywords— Vedic multiplication, urdhva triyakbhyam sutra (algorithm), Nikhilam Sutra,karatsuba- ofman algorithm

I. INTRODUCTION

Multiplication is an important arithmetic operations which is used frequently in hardware level in digital filtering where currently implementations applied in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform (FFT), filtering and in microprocessors in its arithmetic and logic unit [1]. In Electronics, for instance, such polynomials have an important role in the design of analog and digital filters with characteristics close to the ideal ones.[13] A CPU (central processing unit) devotes a considerable amount of processing time in performing arithmetic operations. Multiplication requires substantially more hardware resources and processing time than addition and subtraction. Digital signal processors (DSPs) are the technology that is omnipresent in engineering Discipline. Fast multiplication is very important in DSPs for digital filter, convolution, Fourier transforms etc [4].

Fast polynomial multiplication is a requirement of multi-core as well as DSP processors. Know a day for fast processing DSP processors come with separate adder unit as well as multiplication unit in order to reduce delay and to improve speed of processing. 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 [12].In this paper we are going to develop a polynomial multiplication using Karatsuba multiplication method and nikhilam sutra .Karatsuba method is one of the fastest multiplication methods that are used by Intel and other company's because of its less number of steps. It required three steps to complete their multiplication. Vedic mathematics is research topic now a day .tiryakbhyam and nikhilam are the popular methods for multiplication. This new method is a method where we divide into multiplication steps as in karatsuba method and multiply with nikhilam sutra of Vedic mathematics.

II. NIKHILAM SUTRA OF VEDIC MULTIPLIER

1. NIKHILAM SUTRA

Vedic mathematics is an ancient math which was discovered by "Shri.Bharati Krishna Tirthaji"."vedas" stands for source of knowledge. It contain all mental calculation related to mathematics which covers trigonometry and geometry[15]. A 1500BC vedic math was rediscovered by Shri. Bharati Krishna Tirthaji maharaj between 1911 - 1918 who was a great mathematician and philosopher[15].

This mental calculations i.e vedic mathematics are organized in 16 simple sutras. Sutras means formula, it is an Sanskrit word because Shri.Bharati krishnaji Tirthaji maharaj is also an Sanskrit scholar. Large amount of research has to be going on Vedic mathematics in order to implement in DSP processors. Most of the study and simulation work prove that it require low power consumption and less area.

Nikhilam Sutra stands for "all from 9 and last from 10"[16].Basically this method is use in all type of multiplication, but most efficiently use in large number multiplications. This method take a nearest base of number, larger the number lesser the complexity in multiplication[16].Lets take an example of 96x 93 wherewe take base 100 which is near to number and grater to number as shown below fig.5.

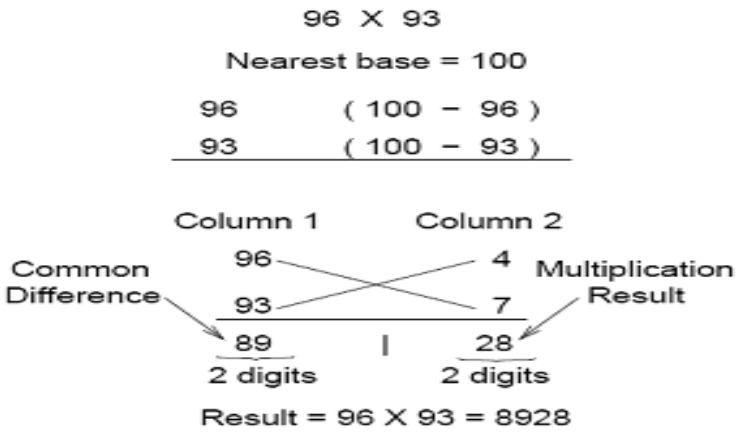

Figure5. Nikhiam Line Diagram of 4 bit number

From above we can see that we take base as 100 which is near to 96 as well as 93, here we subtract number with base. After subtracting number from base we get 4 and 7.we place this number on left hand side we see on column 2,Know multiply this number we get an 28.In column 1 we placed a numbers ,for LSB we have to subtract number with their opposite number remainder which obtain during subtraction. Let's see 96-7=89 or 93-4=89.Know merges this answer with the answer that obtains during first multiplication. from the above we can see that this method required only two multiplication steps as well as less complexity. Nikhilam power will make pupils play with power of base like 10,100...etc

III. THE KARATSUBA MULTIPLICATION

If we want to multiply large numbers, it's possible to use other techniques than the traditional multiplication, which is time expensive. In this case the Karatsuba Multiplication Algorithm [8], [9] is one of the choices. But this method is one of the fastest as well as mostly uses method in processors, because this method required only three multiplications and most of time this method is recursive in nature. For convenience, we will focus on long integers in binary representation [9]. If we have the numbers

$U = (u_{2n-1}...u_1u_0)_2$ and $V = (v_{2n-1}...v_1v_0)_2$ (1)

Represented on 2n bits, we can write then

$U = 2^nU_1 + U_0$ and $V = 2^nV_1 + V_0$, where,

$U_1 = (u_{2n-1}...u_n)$, $U_0 = (u_{n-1}...u_1u_0)$,

$V_1 = (v_{2n-1}...v_n)$, $V_0 = (v_{n-1}...v_1v_0)$.

Now, we have:

$$UV = (2^nU_1 + U_0)(2^nV_1 + V_0) =$$
$$= 2^{2n}U_1V_1 + 2n(U_1V_0 + U_0V_1) + U_0V_0 =$$
$$= 2^{2n}U_1V_1 + 2^n((U_1 + U_0)(V_1 + V_0) - U_1V_1 - U_0V_0) + U_0V_0 \quad (2)$$

This formula reduce the original 2n bits operands multiplication to three n bits operands multiplication: U_1V_1, $(U_1 + U_0)(V_1 + V_0)$ and U_0V_0, and few simple operations (like shift and addition). The main advantage of using the formula (2) is that we can define a recursive process for multiplication, which is faster than the traditional multiplication. The traditional multiplication has the complexity $O(n^2)$ [9]. Proceeding recursively we obtain the bit complexity $O(n^{\log 3})$, where

$$\log 3 = 1.58... < 2. \text{ [10]}$$

Lets take two numbers Xa Xb and Ya Yb Then

$$A = Xb.\ Yb$$
$$B = Xa\ Ya$$
$$C = (Xa+Xb)(Ya+Yb) - A - B$$

From above it is clear that we required three multiplication and thus increase the speed of multiplication.It is similar to Urdhva Tiryakbhyam, but it reduce multiplication steps because A and B multiplication is already performed and we have to take value in to C as an input.When this technique is recursively applied to ultidigit numbers, a point is reached in the recursion when the overhead of addition and subtraction makes it more efficient to use the usual $O(n^2)$ multiplication algorithm to evaluate the partial products. We call this point "Karatsuba threshold". The most efficient overall method therefore relies on a combination of Karatsuba and conventional multiplication [6]. Namely, we'll have $n^{\log 3/\log 2}$ digit products for operands of length n, not n^2 like in the traditional multiplication. (See [9], p. 233)It's convenient to see this algorithm in terms of a ternary tree. Each node has three children that compute the partial products and at each level the input length is divided with two. The leaves perform the classical multiplication. The threshold for switching to the traditional multiplication method is determined experimentally [3].

IV. METHODOLOGY

1. *New Methodology For polynomial multiplication using Karatsuba Algorithm and Nikhilam Sutra.*

Karatsuba Algorithm is the Algorithm which is fastest and most preferred by processors Developer because of less multiplication steps i.e. is 3.most probably it call itself i.e. recursive Karatsuba ,in our method while calling Karatsuba we going to call nikhilam sutra method of vedic mathematics.

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be represented as n-digit strings in some base B. For any positive integer m less than n, one can write the two given numbers as

$$x = x_1B^m + x_0$$
$$y = y_1B^m + y_0,$$

where x_0 and y_0 are less than B^m. The product is then

$$xy = (x_1B^m + x_0)(y_1B^m + y_0)$$
$$= z_2B^{2m} + z_1B^m + z_0$$

where

$$\boldsymbol{z_2 = x_1 * y_1} \quad \text{(a)}$$
$$\boldsymbol{z_1 = x_1 * y_0 + x_0 * y_1} \quad \text{(b)}$$
$$\boldsymbol{z_0 = x_0 * y_0.} \quad \text{(c)}$$

The above steps are same to Urdhva trikyabhayam sutra of Vedic mathematics, where above equation shows that this three equation require 4 steps of multiplication, which is very expensive as well as time consuming in the field of computer.

Karatsuba reduce these steps in such a way that 4 steps of multiplication get reduce to 3 steps. In Karatsuba multiplication equation (a) and (b) is performed first. Then it performs third steps in such a way that these steps require only one multiplication step.

Let see.

$\boldsymbol{z_1 = x_1 * y_0 + x_0 * y_1}$ this is over basic step, which can be simplified as

$\boldsymbol{z_1 = (x_1 + x_0) * (y_0 + y_1) - (a) - (b)}$ (d)

from above equation we can see that equation (b) is simplified into (d) steps, where we see that only one multiplication is require to be done. While performing a large operation it performed recursive operation, so in recursion it again performs the three steps. so in order to reduce multiplier we are going to use Nikhilam sutra inside this Karatsuba steps. While calling Karatsuba our method call nikhilam sutra. Let's take an example

x= 9 Bm + 8.....(1)

y = 7 Bm + 8.....(2)

in simplified form we can write x = 98 and y=78 lets see simplification x*y using nikhilam sutra in karatsuba method

X * Y = 98 * 78

from above equation (a) of karatsuba method first step is $\boldsymbol{z_2 = x_1 * y_1}$ .the nikhilam sutra solution is given below .

Z2= 9 * 7 = 63 (A)

9	10-9=1
7	10-7=3
7-1/9-3=6	1*3=3
Z2=63	

from above equation (c) of karatsuba method first step is $\boldsymbol{z_0 = x_0 * y_0}$. the nikhilam sutra solution is given below.

Z0 = 8 * 8 = 64 (B)

8	10-8=2
8	10-8=2
8-2/8-2=6	2*2=4
Z0=64	

from above equation (d) of Karatsuba method first step is $\boldsymbol{z_1 = (x_1 + x_0) * (y_0 + y_1) - (a) - (b).}$ the nikhilam sutra solution is given below

Z1= (9 + 8) * (7 + 8) – A - B (C)

17	10-17=-7
15	10-15=-5
17-(-5)/15-(-7)=22	7*5=35
2(2+3)5=225	

Know this 225 get subtract equation 1 and 2 then we get answer of step 3

From equation (c)

Z0 = 225-64-63=128 (4)

Know we get add equation 4, 2 and 1 answer and we the final answer

$$63*100+128*10+64=63\ z^2+128\ z+64=7644 \quad (5)$$

3.2.2. Flowchart

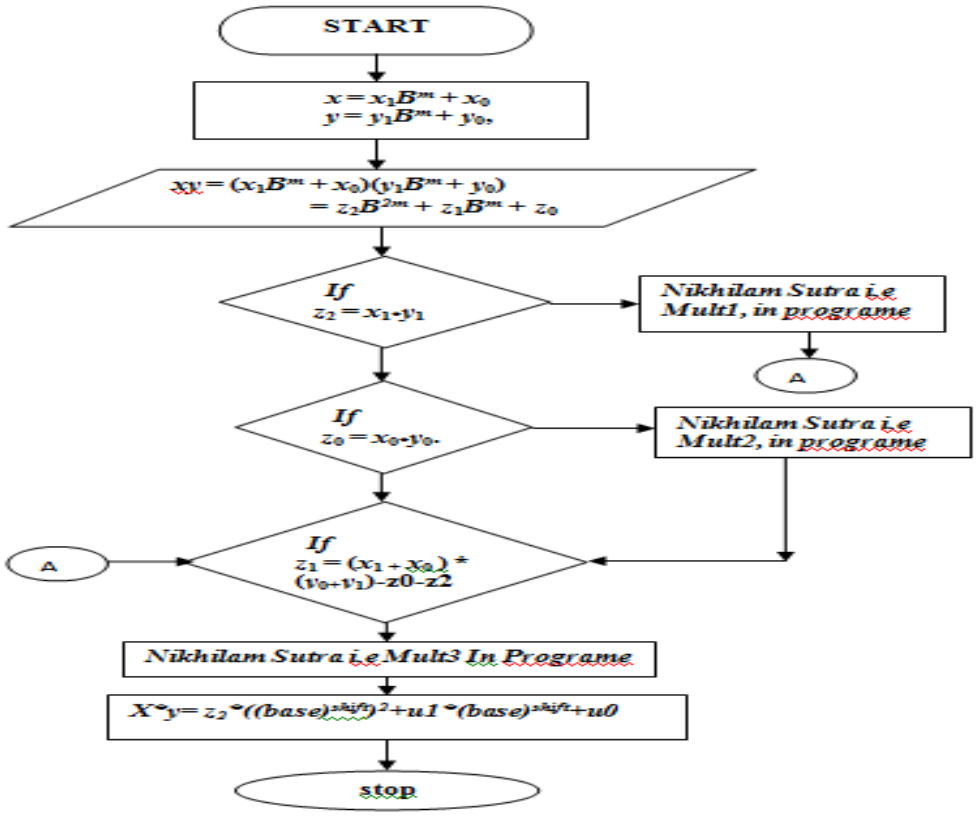

Figure 3.1.FlowChart of proposed polynomial multiplication

The flow chart of proposed methodology is given above where we can see that for each step we call nikhilam sutra programme i.e MULT1,MULT2, MULT3 .In above flow chart we can see that for each step there is separate programme of nikhilam sutra .In this method in place of calling Karatsuba method after an all we call nikhilam sutra method for fastest processing .means in recursive Karatsuba we can call nikhilam sutra of vedic mathematics.

V. RESULTS AND DISCUSSION

3. SIMULATION

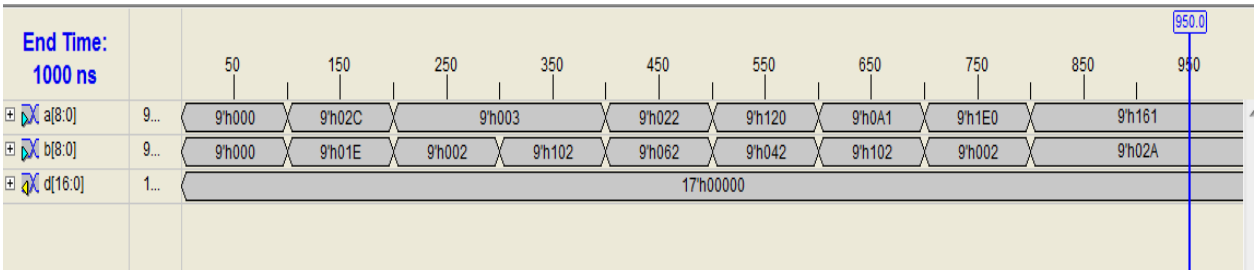

Figure 3.Simulation waveform

3.1. *Simulated input a=000000011*

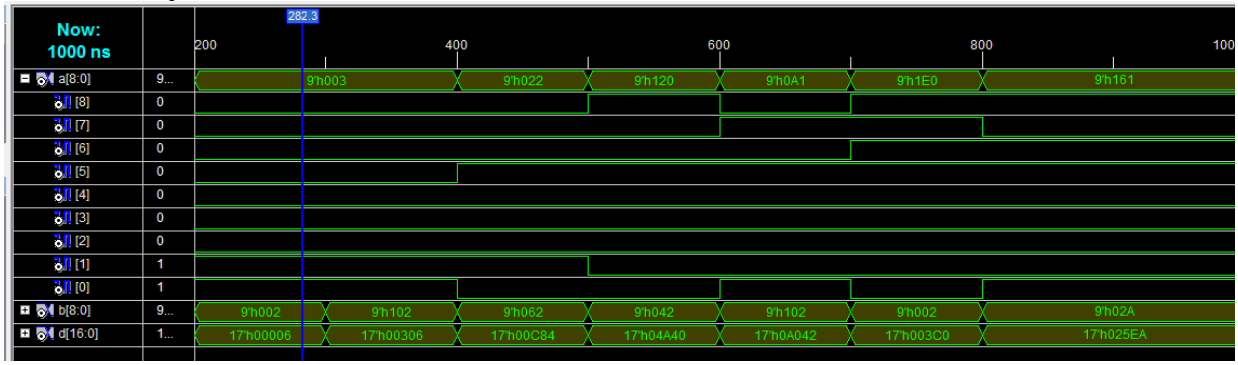

Figure 3.1simulated Waveform For A = 00000011

3.2. *Simulated input b=00000010*

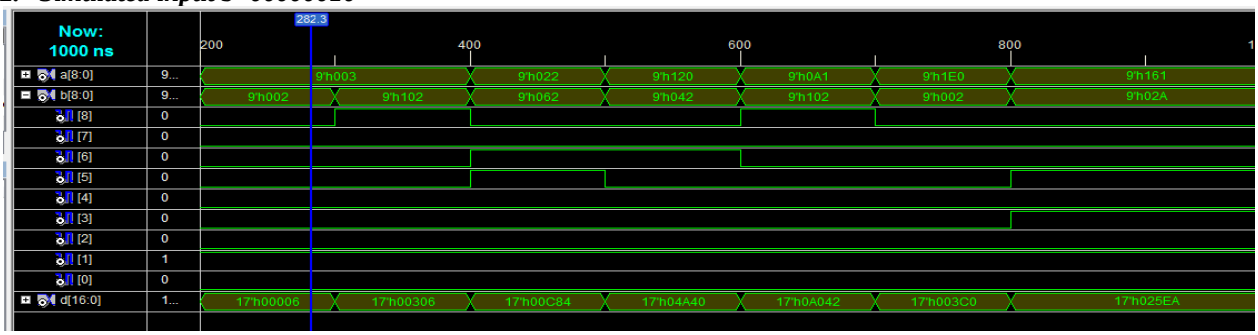

Figure 3.2.simulated Waveform For A = 00000010

3.3. *Simulated Output=0000000000000110*

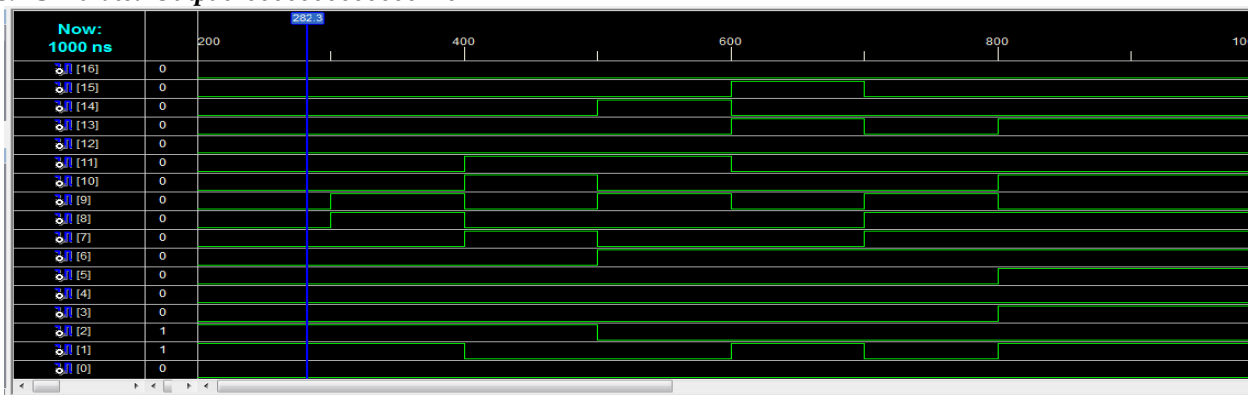

Figure 3.3simulated Waveform For OUTPUT = 0000000000000110

4. SYNTHESIS RESULTS AND COMPARISONS OF NEW METHODS

DEVICE(Spartan 2xc2s200 pq208)	NUMBER OF SLICES	NUMBER OF 4 INPUT LUTs	NUMBER OF BOUNDED IOBs	MAXIMUM COMBINATIONAL DELAY
KARATSUBA MULTIPLIER	26 out of 2352	45 out of 4704 (0%)	31out of 140 (22%)	12.33ns
KARATSUBA USING NIKHILAM SUTRA	30 out of 2352	57 out of 4704	35 out of 140	14.20ns
VEDIC MULTIPLIER	62 out of 2352	113 out of 4704 (2 %)	32 out of 140 (22%)	37.34 ns

Table.4.Synthesis Results And Comparisons Of New Methods With Other

IV. CONCLUSION

This is found out that Karatsuba Algorithm and Nikhilam Sutra is giving the most efficient multiplier output than the other sutras in terms of speed, space and less power consumption. As the need of low power devices is mounting in digital world due to battery operation and device's operational longevity is the main concern these multipliers are the main focuses of researches than the conventional multipliers.Karatsuba along with the Vedic shows it even more efficient type of algorithm than its individual approaches where speed of throughput is drastically improved over other methods.

REFERENCES

[1] Wallace, C.S., "A suggestion for a fast multiplier," IEEE Trans. Elec. Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.

[2] Booth, A.D., "A signed binary multiplication technique," Quarterly Journal of Mechanics and Applied Mathematics, vol. 4, pt. 2, pp. 236– 240, 1951.

[3] Jagadguru Swami Sri Bharath, Krsna Tirathji , "Vedic Mathematics or Sixteen Simple Sutras from the Vedas", Motilal Banarsidas, Varanasi (India), 1986.

[4] Arushi Somani, Dheeraj Jain, Sanjay Jaiswal, Kumkum Verma and Swati Kasht, "Compare Vedic Multipliers with Conventional Hierarchical array of array multiplier", International Journal of Computer Technology and Electronics Engineering (IJCTEE),Volume 2, Issue 6, December 2012.

[5] Neil H.E Weste, David Harris, Ayan anerjee, "CMOS VLSI Design, A Circuits and Systems Perspective", Third Edition, Published by Person Education, PP-327-328].

[6] Mrs. M. Ramalatha, Prof. D. Sridharan , "VLSI Based High Speed Karatsuba Multiplier for Cryptographic Applications Using Vedic Mathematics", IJSCI, 2007.

[7] Thapliyal H. and Srinivas M.B. "High Speed Efficient N x N Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics", Transactions on Engineering, Computing and Technology, 2004, Vol.2.

[8] Knuth, D. E, "The art of computer programming, volume 2", Addison-Wesley, 2 editions, 1981.

[9] Karatsuba, A.; Ofman, Yu. (1962) , "Multiplication of multidigit numbers on automata", Sov. Phys. Dokl., 7:595-597.

[10] D. Zuras, "On squaring and multiplying large integers", In Proceedings of International Symposium on Computer Arithmetic, IEEE Computer Society Press, pp. 260-271, 1993.

[11] Shripad Kulkarni, "Discrete Fourier Transform (DFT) by using Vedic Mathematics", Papers on implementation of DSP algorithms/VLSI structures using Vedic Mathematics, 2006, www.edaindia.com, IC Design portal.

[12] Triveni.K.S, E. Elavarasi, "High Speed Efficient Karatsuba- Ofman Pipelined Multiplier for Low Contrast Image Enhancement", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-5, June 2013.

[13] Juliano B. Lima Daniel Panario and Qiang Wang, "A Karatsuba-based Algorithm for Polynomial Multiplication in Chebyshev", IEEE Trans. Computers 59(6):835-841 (2010)

[14] Himanshu Thapliyal, "Vedic Mathematics for Faster Mental Calculations and High Speed VLSI Arithmetic" , Invited talk at IEEE Computer Society Student Chapter, University of South Florida, Tampa, FL, Nov 14 2008.

[15] Jeganathan Sriskandarajah, "Secrets of Ancient Maths: Vedic Mathematics", Journal of Indic Studies Foundation, California, pages 15 and 16.

[16] Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho, "Multiplier design based on ancient Indian Vedic Mathematics," in IEEE International SoC Design Conference, pp. II-65 - II-68, November 2008.

