

Performance Analysis of Various Vedic Techniques for Multiplication

Kavita^{#1}, Umesh Goyal^{#2}

^{#1}Electronics and Electrical Communication Department, PEC University of Technology, India

^{#2}Electronics and Electrical Communication Department, PEC University of Technology, India

Abstract— Multiplication is an important factor in arithmetic operations and is carried out in a number of Digital Signal Processing applications. As Multipliers take a long time for execution so there is a need of fast multiplier to save the execution time. This paper describes the multiplication using Ancient Indian Vedic Mathematics multiplication techniques. Vedic Mathematics is the ancient system of mathematics which has a unique technique of calculations based on 16 Sutras. The techniques described in this paper are Nikhilam Sutra, Urdhva Tiryakbhyam and Karatsuba-ofman and the performance analysis of these techniques is obtained. Modelsim tool is used for simulation and the results obtained are compared on the basis of time delay of multiplication. This paper also describes the deliberateness and the fastness of different multiplier techniques compared with each other.

Keywords— Vedic Mathematics, Multiplication, Nikhilam Sutra, Urdhva Tiryakbhyam, Karatsuba-ofman

I. INTRODUCTION

Due to the growth of computer and signal processing applications the demand for high speed processing has been increased. In order to achieve the desired performance in many real-time signal and image processing applications higher throughput arithmetic operations are important. Multiplication is one of the key arithmetic operations in such applications. Speed of multiplication operation is of great importance in DSP. The multiplier is a large block of computing system [1]. The amount of circuitry involved is directly proportional to the square of its resolution. From delay perspective latency and throughput are the two major concerns in DSP applications for multiplication algorithm. Latency is the real delay of computing a function. Throughput is the measure of how many multiplications can be performed in a given period of time. Multiplier based on Vedic Mathematics is one of the fast multiplier as compared to other methodologies used for multiplication such as Array multiplier, Wallace tree and Booth multiplier. This paper describes the different techniques used in Ancient Vedic Mathematics for multiplication and also compares them.

II. VEDIC MATHEMATICS

The word “Vedas” which literally means knowledge has derivational meaning as principle and limitless store-house of all knowledge. The word Veda also refers to the sacred ancient Hindu literature which is divided into four volumes.

Vedas initially were passed from previous generation to next generation orally. Later they were transcribed in Sanskrit [2].

Vedas include information from many subjects such as from religion, astronomy, architecture, mathematics, medicine etc. Vedic mathematics is not only a mathematical wonder but also it is logical. That’s why Vedic mathematics has such a degree of prominence which cannot be disapproved. Due to these characteristics, Vedic mathematics has already crossed the boundaries of India and has become a leading topic of research abroad. Vedic mathematics deals with various mathematical operations.

The system of Vedic mathematics is based on 16 Sutras – formulas and 13 Up-sutras or Corollaries [3].

The 16 Sutras are:

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

The 13 Up-sutras are described as below:

1. Anurupyena
2. Shishyate Sheshsamjnah
3. Adyamadye Nantyamantyena
4. Kevalaih Saptakam Gunyat
5. Vestanam
6. Yavadunam Tavadunam
7. Yavadunam Tavadunikutya Varganka ch Yojayet
8. Antyayordhshakepi
9. Antyatoreva
10. Samucchayagunitah
11. Lopanasthapanabhyam
12. Vilokanam
13. Gunitasamucchyah Samucchayagunitah

III. DESIGN OF VEDIC MULTIPLIER

This section describes the techniques used for multiplication of two numbers in Ancient Vedic Mathematics. These techniques are: Nikhilam Sutra, Urdhva Tiryakbhyam and Karatsuba-ofman.

A. Nikhilam Sutra

Nikhilam Sutra literally means “all from 9 and last from 10” [2]. This technique is applicable to all the cases of multiplication but it is more efficient in case of multiplication of large numbers. In this method, the compliment of the large number is found from its nearest base to perform the multiplication operation on it, hence larger the original number, lesser the complexity of the multiplication. Let us understand this technique with the help of some example. If we want to multiply two numbers i.e. 88 x 97, it can be done as follows using Nikhilam Sutra technique.

$$88 \times 97$$

Select near Base as 100,

TABLE I
SELECTING THE NEAREST BASE AS 100

88	=100 - 12
97	=100 - 3

Common	88	X	12	Multiplication
Difference	97	X	3	Result

TABLE III
PRODUCT RESULT

85	36
----	----

Hence the result is $88 \times 97 = 8536$

As shown above in Table I, we write the multiplier and the multiplicand in two rows followed by the differences of each of them from the chosen base, i.e., their compliments. We can now write two columns of numbers, one consisting of the numbers to be multiplied (Column 1) and the other consisting of their compliments (Column 2). The product also consists of two parts which are written in table II for the purpose of illustration. The right hand side (RHS) of the product can be

obtained by simply multiplying the numbers of the Column 2 ($12 \times 3 = 36$). The left hand side (LHS) of the product can be found by cross subtracting the second number of Column 2 from the first number of Column 1 or vice versa, i.e., $88 - 3 = 85$ or $97 - 12 = 85$. The final result is obtained by concatenating RHS and LHS i.e. 8536.

Architecture of Vedic Multiplier Using Nikhilam Sutra:

Broadly this architecture is divided into three parts.

- (i) Radix Selection Unit
- (ii) Exponent Determinant
- (iii) Multiplier.

Hardware implementation of this multiplier is shown in Fig. 1 [4]. The architecture can be decomposed into three main subsections: (i) Radix Selection Unit (RSU) (ii) Exponent Determinant (ED) and (iii) Multiplier. The RSU is required to select the proper radices corresponding to the input numbers.

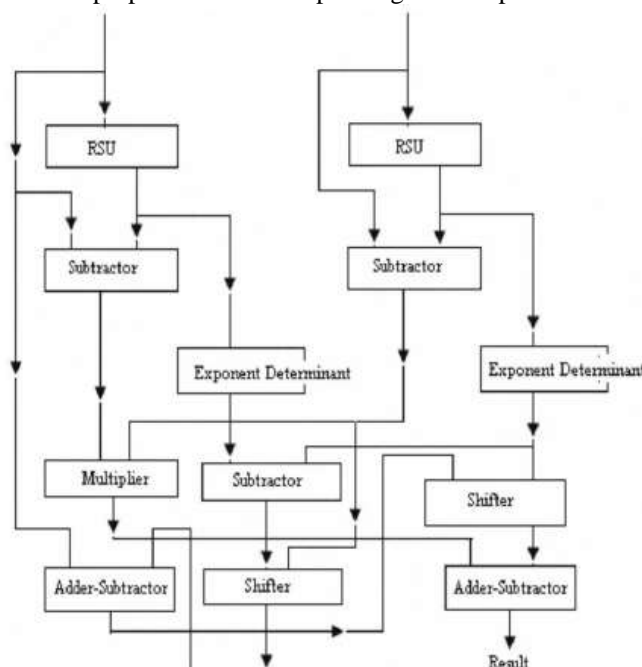


Fig. 1 Architecture of Vedic Multiplier using Nikhilam Sutra

B. Urdhva-Tiryakbhyam Sutra

Urdhva-tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means “Vertically and Crosswise” [5]. To illustrate this multiplication scheme, let us consider the multiplication of two decimal numbers (5498×2314). The conventional methods already know to us will require 16 multiplications and 15 additions [6].

An alternative method of multiplication using Urdhva-Tiryakbhyam Sutra is shown in Fig. 2. The numbers to be multiplied are written on two consecutive sides of the square as shown in the figure. The square is divided into rows and

columns where each row/column corresponds to one of the digit of either a multiplier or a multiplicand. Thus, each digit of the multiplier has a small box common to a digit of the multiplicand. These small boxes are partitioned into two halves by the crosswise lines. Each digit of the multiplier is then independently multiplied with every digit of the multiplicand and the two-digit product is written in the common box. All the digits lying on a crosswise dotted line are added to the previous carry. The least significant digit of the obtained number acts as the result digit and the rest as the carry for the next step. Carry for the first step (i.e., the dotted line on the extreme right side) is taken to be zero.

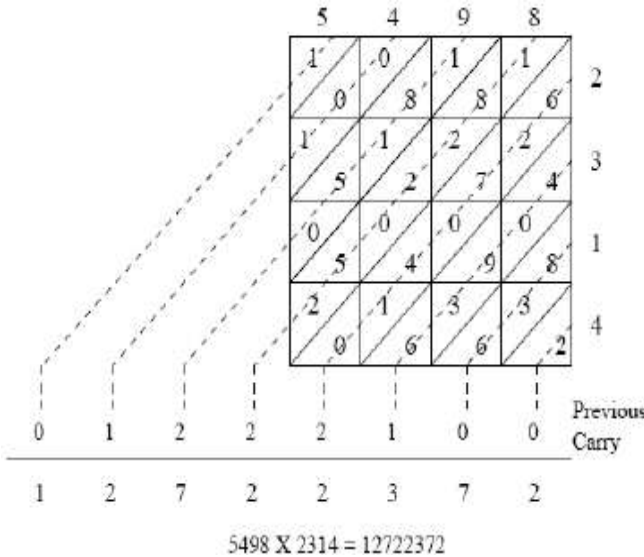


Fig. 2 Multiplication of two large integers

Another way of multiplication of two numbers using this technique is as shown below in Fig. 3.

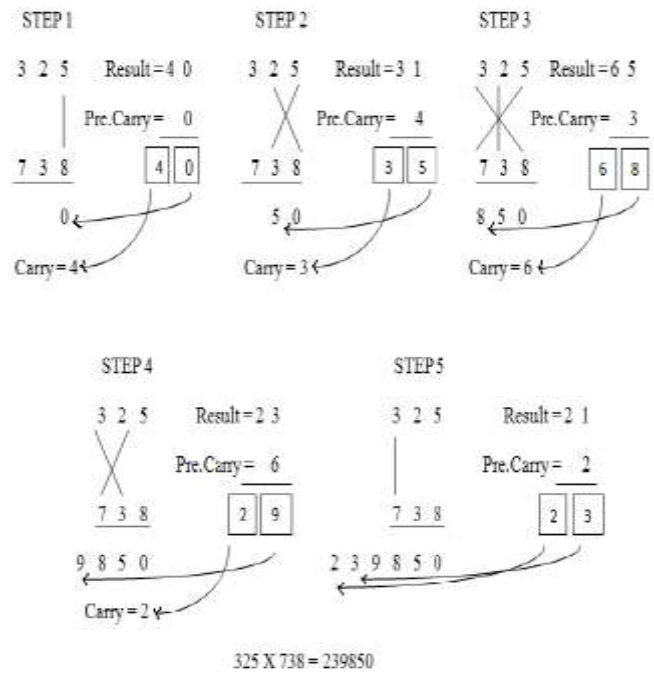


Fig. 3 Alternative method of Multiplication of two large integers

To illustrate this multiplication scheme, let us consider the multiplication of two decimal numbers (325 * 738). Line diagram for the multiplication is shown in Fig.2.2. 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, 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 [5].

C. Karatsuba-Ofman

Karatsuba-Ofman algorithm is considered as one of the fastest ways to multiply long integers. It is based on the divide and conquers strategy [3]. A multiplication of 2n digit integer is reduced to two n digit multiplications, one (n+1) digit multiplication, two n digit subtractions, two left shift operations, two n digit additions and two 2n digit additions.

Let A and B are the binary representation of two long integers. They can be written as follows:

$$A = \sum_{t=0}^{k-i} a_t 2^t$$

$$B = \sum_{t=0}^{k-1} b_t 2^t$$

We wish to compute the product AB. Using the divide and conquer strategy, the operands A and B can be decomposed

into equal size parts A_H and A_L , B_H and B_L , where subscripts H and L represent high and low order bits of A and B respectively.

Let $k=2n$. If k is odd, it can be right padded with a zero

$$A = 2^n \sum_{i=0}^{n-1} a_{i+n} 2^i + \sum_{i=0}^{n-1} a_i 2^i = A_H 2^n + A_L$$

$$B = 2^n \sum_{i=0}^{n-1} b_{i+n} 2^i + \sum_{i=0}^{n-1} b_i 2^i = B_H 2^n + B_L$$

The product AB can be computed as follows:

$$P = A * B$$

$$P = (A_H 2^n + A_L) (B_H 2^n + B_L)$$

For Multiplier, first the basic blocks, that are the 2x2 bit multipliers have been made and then, using these blocks, 4x4 block has been made by adding the partial products using carry save adders and then using this 4x4 block, 8x8 bit block, 16x16 bit block and then finally 32 x 32 bit Multiplier as shown in figure 4 has been made.

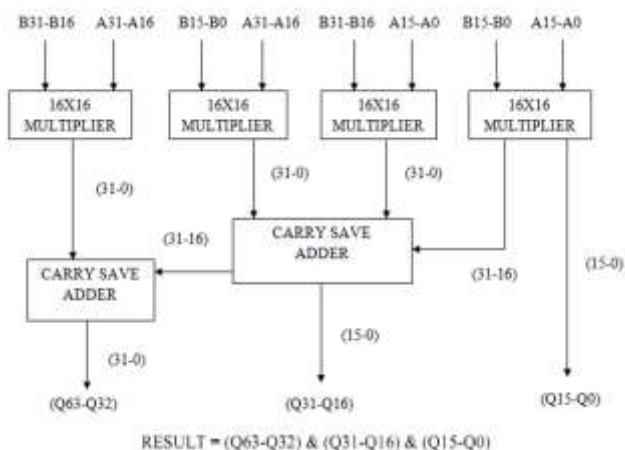


Fig. 4 32x32 bit multiplier

IV. IMPLEMENTATION AND ANALYSIS OF MULTIPLIERS

The implementation of different multipliers is done using VHDL and the simulation results are obtained on ModelSim. Fig. 5 shows the multiplication of two 8 bit numbers using Karatsuba-Ofman technique and Fig. 6 shows the multiplication of two 16 bit numbers.

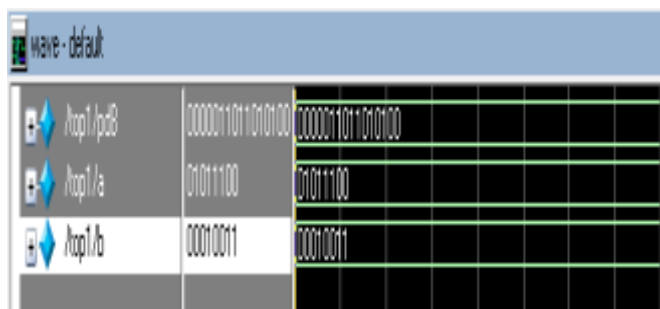


Fig. 5 8x8 bit multiplier (Karatsuba-Ofman)

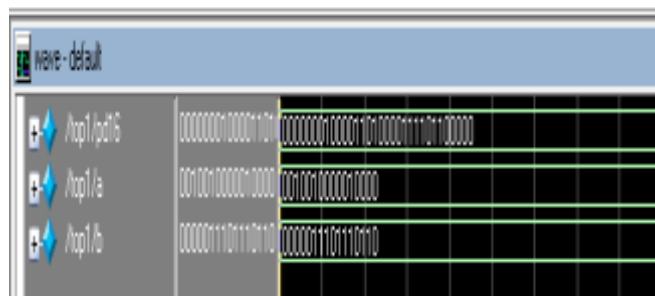


Fig. 6 16x16 bit multiplier (Karatsuba-Ofman)

Similarly the waveforms for other techniques are also drawn for both 8 bit and 16 bit numbers. Their timing delays are found that tells the time taken by a specific multiplier for performing its task of multiplication. Table III describes the data regarding the timing delays in case of Karatsuba-Ofman technique. Similarly Table IV and Table V shows the values of time delays in case of Nikhilam Sutra and Urdhva Tiryakbhyam Sutra algorithms.

TABLE III
TIMING RESULTS OF KARATSUBA-OFMAN

Size	Delay in ns
8-bit (Karatsuba-Ofman)	15.418
16-bit (Karatsuba-Ofman)	22.604

TABLE IV
TIMING RESULTS OF NIKHILAM SUTRA

Size	Delay in ns
8-bit (Nikhilam Sutra)	18.699
16-bit (Nikhilam Sutra)	20.094

TABLE V
TIMING RESULTS OF URDHVA TIRYAKBHYAM SUTRA

Size	Delay in ns
8-bit (Urdhva Tiryakbhyam)	13.455
16-bit (Urdhva Tiryakbhyam)	25.083

V. CONCLUSION

From the timing delay analysis of different ancient Vedic mathematics techniques, the following conclusion can be made regarding the relative speed of the multipliers. These relative speeds are summarized in Table VI as follows

TABLE VI
RELATIVE SPEED COMPARISON

Size→	8x8	16x16
Karatsuba-Ofman	Faster than Nikhilam by 21.28 %	Faster than urdhva Tiryakbhyam by 10.96%

Nikhilam Sutra	Slowest	Faster than Karastuba-Ofman by 12.49%
Urdhva Tiryakbhyam	Faster than Karatsuba-Ofman by 14.58%	Slowest

REFERENCES

- [1] Kabiraj Sethi, Rutuparna Panda, An Improved Squaring Circuit for Binary Numbers, International Journal of Advanced Computer Science and Applications, Vol 3, No. 2, 2012.
- [2] D.Kishore Kumar, A.Rajakumari, Modified Architecture of Vedic Multiplier for High Speed Applications, International Journal of Engineering Research & Technology, Vol. 1 Issue 6, August – 2012.
- [3] Abhishek Gupta, Arithmetic Unit Implementation Using Delay Optimized Vedic Multiplier with BIST Capability, International Journal of Engineering and Innovative Technology, Volume 1, Issue 5, May 2012.
- [4] M.Nagaraju, R.Surya Prakash, B.Vijay Bhaskar, High Speed ASIC design of Complex Multiplier Using Vedic Mathematics, International Journal of Engineering Research and Application, Vol. 3, Issue 1, January-February 2013, pp. 1079-1084.
- [5] Pushpalata Verma, K. K. Mehta, Implementation of efficient multiplier based on Vedic Mathematics using EDA tool, International Journal of Engineering and Advance Technology, Volume-1, Issue-5, June 2012.
- [6] G.Ganesh Kumar, V.Charishma, Design of high Speed Vedic Multiplier using Vedic Mathematic Techniques, International Journal of Scientific and Research Publication, Vol 2, Issue 3, March 2012.
- [7] Ramachandran.S*, Kirti.S.Pande, Design, Implementation and Performance Analysis of an Integrated Vedic Multiplier Architecture, International journal of Computational Engineering Research.
- [8] P. Saha, A. Banerjee, A. Dandapat, P. Bhattacharyya , Vedic Mathematics Based 32-bit Multiplier Design for High Speed Low Power Processor, International Journal on smart sensing and Intelligent Systems.
- [9] G.Ganesh Kumar*, C. Venkata Sudhakar**, M.Naresh Babu***, Design of High Speed Vedic Square by using Vedic Multiplication Techniques, International Journal of scientific & Engineering Research, Volume 4, Issue 1, January 2013.