ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Study of Vedic Multiplier Algorithms using Nikhilam Method

Prof S. B. Somani[1], Nikhil R. Mistri[2]

Associate Professor, Dept. of ECE, MIT College of Engineering, Pune, India[1]

PG Student [VLSI], Dept. of ECE, MIT College of Engineering, Pune, India [2]

ABSTRACT: In High speed arithmetic logic units, Multiplier and accumulator unit, digital signal processing units, multipliers are the key block. In computer arithmetic, multiplication is the most important operation. Multiplication operation is very significant for arithmetic operation like correlation, convolution, frequency analysis, image processing etc. hence Process time of multiplication operations must be very efficient. As constraints on delay increases, there is a need to design for faster multiplication. Many modifications over the standard algorithm has been made to enhance the speed. Methods like Wallace tree, Booth algorithm and several others techniques are being worked upon to increase speed. Amongst this Vedic multipliers based on Vedic sutras are under focus for being fastest and low power multiplier. Vedic mathematics contains 16 sutras[5] out of which “UrdhvaTiryakbhyam” and “Nikhilam Navatashcaramam Dashatah” sutras are most efficient one in terms of speed. In this Paper “Nikhilam Navatashcaramam Dashatah” sutras is presented with its scope to multiply two binary numbers.

KEYWORDS: Nikhilam Navatashcaramam Dashatah, Vedic mathematics, multiplier, Karatsuba algorithm.

I.INTRODUCTION

Operations like multiply and accumulate (MAC), Digital signal processing applications like correlation, convolution, and Digital image processing applications mostly use Multipliers. Since in Digital signal processing, the operation time is mostly dominated by multiplication time, there is a need of high speed multiplier.

As signal processing and computer arithmetic operations are expanding, to achieve desired performance and higher throughput, there is a demand for high speed, low power multiplier design. Hence development of high speed low power multiple has been subject of focus over recent years.

Vedic Multipliers are one of the fast and low power multipliers. Vedic mathematics is believed to be reconstructed by “Shri Bharti Krishna Tirathaji” between the years 1911 to 1918[5]. Vedic mathematics is divided into sixteen sutras which can be applied to any branch of mathematics like algebra, trigonometry, geometry etc. Vedic mathematics sutras reduces the complex calculations using simple methods similar to the working of human minds. Multipliers based on Vedic mathematics are used in various applications like ALU, MAC etc.

Vedic sutras are divided into sixteen sutras which are briefed alphabetically. Out of this sutras “Nikhilam NavatascaramamDasatah” has been discussed in this paper.

Sr No.	Sutras	Meaning
1	AnurupyeShunyamanyat	If one is in ratio, the other is zero
2	Chalana-Kalanabyham	Differences and Similarities
3	EkadhikinaPurvena	By one more than the previous one
4	EkanyunenaPurvena	By one less than the previous one
5	Gunakasamuchyah	The factors of the sum is equal to thesum of the factors
6	Gunitasamuchyah	The product of the sum is equal to thesum of the product

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Sr. No.	Sutras	Meaning
7	Nikhilam NavatashcaramamDashatah	All from 9 and the last from 10
8	ParaavartyaYojayet	Transpose and adjust
9	Puranapuranabyham	By the completion or non-completion.
10	Sankalana-vyavakalanabhyam	By addition and by subtraction
11	ShesanyankenaCharamena	The remainders by the last digit
12	ShunyamSaamyasamuccaye	When the sum is the same that sum is zero
13	Sopaantyadvayamantyam	The ultimate and twice the penultimate
14	UrdhvaTiryakbyham	Vertically and crosswise
15	Vyashtisamanstih	Part and Whole
16	Yaavadunam	Whatever the extent to fits deficiency

Table 1: List of Vedic Sutras[1]

NIKHILAM NAVATASHCARAMAM DASHATAH

Nikhilam NavatashcaramamDashatah is one of the 16 sutra from Vedic Mathematics. This sutra is used to convert large- digit multiplication to small digit multiplication using few subtract, add, and shift operation. Steps for multiplication of 2 digit number which are less than the nearest base using Nikhilam sutra are as shown in table 2.

Step 1	$a \times b$
Step 2	$A = (\text{Nearest base}) - a$
Step 3	$B = (\text{Nearest base}) - b$
Step 4	$C = A \times B$
Step 5	$D = a - B = b - A$
Step 6	$Result = 100 \times D + C$

Table 2:Steps for Nikhilam sutra (number less than nearest base) [4]

Using Nikhilam Sutra, 2-digit multiplication operation can be performed in 1 multiplication where classical multiplication needs four multiplications. Table 3 below is the multiplicationexample of2 digit number which are less than the nearest base using Nikhilam algorithm.

Step 1	98×99
Step 2	$A = 100 - 98 = 2$
Step 3	$B = 100 - 99 = 1$
Step 4	$C = 2 \times 1 = 2$
Step 5	$D = 98 - 1 = 99 - 2 = 97$
Step 6	$Result = 100 \times 97 + 2 = 9702$

Table 3:Steps for 96 x 97 multiplication using Nikhilam algorithm

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Steps for multiplication of 2 digit number which are greater than the nearest base using Nikhilam sutra are as shown in table 4.

Step 1	$\boldsymbol{a \times b}$
Step 2	$\mathbf{A = a -}$ **(Nearest base)**
Step 3	$\mathbf{B = b -}$ **(Nearest base)**
Step 4	$\boldsymbol{C = A \times B}$
Step 5	$\boldsymbol{D = a + B = b + A}$
Step 6	$\boldsymbol{Result = 100 \times D + C}$

Table 4:Steps for Nikhilam sutra (number greater than nearest base) [4]

Below is the example of 2 digit number which are greater than the nearest base using Nikhilam algorithm. We will multiply 108 x 109 using Nikhilam algorithm.

Step 1	108×109
Step 2	$A = 108 - 100 = 8$
Step 3	$B = 109 - 100 = 9$
Step 4	$C = 8 \times 9 = 72$
Step 5	$D = 108 + 9 = 109 + 8 = 117$
Step 6	$Result = 100 \times 117 + 72 = 11772$

Table 5:Steps for Nikhilam sutra (number greater than nearest base)

Below is the example of 2 digit number multiplication which has base other than 100 and this base need to be modified. Consider example of multiplying 46 x 45. Here base is 50.

Step 1	46×45
Step 2	$A = 50 - 46 = 4$
Step 3	$B = 50 - 45 = 5$
Step 4	$C = 4 \times 5 = 20$
Step 5	$D = 46 - 5 = 45 - 4 = 41$
Step 6	$Result = 50 \times 41 + 20 = 2070$

Table 6:Steps for multiplying 2- digit number with modified base

II.APPROACHES FOR VEDIC MULTIPLIER USING NIKHILAM SUTRA

A. KARATSUBA ALGORITHM

Karatsuba algorithm is the simplest algorithm which is based on divide and conquer paradigm. Karatsuba algorithm can multiply 2-digit multiplication in 3 multiplication steps instead of 4 steps[3]. If we need to multiply two, 2 – digit decimal numbers $a_1 a_2 \times b_1 b_2$ then,

1. **A =** $\boldsymbol{a_1 \times b_1}$
2. **B =** $\boldsymbol{a_2 \times b_2}$
3. **C =** $\boldsymbol{(a_1 + a_2) \times (b_1 + b_2)}$
4. **D = C –A – B , here D =** $\boldsymbol{(a_1 \times b_2) + (a_1 \times b_1)}$
5. **Result = 100 x A + 10 x D + b;**

If the number has more digits, then this method can be applied by splitting the multiplicand and multiplier in two parts. $O\left(n^{\log\frac{3}{\log 2}}\right)$[3].Some shift operation and addition operation can be ignored as multiplication operation is costly.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

For Example, let multiply 95 x 96. Using Karatsuba algorithm we can compute it as follows.

1. A = 9 x 9 = 81
2. B = 5 x 6 = 30
3. C = (9 + 5) x (9 + 6) = 14 x 15 = 210
4. D = C – A – B = 210 – 81 – 30 = 99
5. Result = 100 x A + 10 x D + B = 100 x 81+10 x 99 + 30 = 9120

Binary Multiplication

Binary multiplication can be performed by converting n-bit multiplication to (n-1) bit multiplication using some additional add, subtract and shift operation. This conversion is applied repeatedly until we get 1-bit multiplication.

2-bit multiplication can be performed using single bit. For Example, multiplication of 11 x 11 can be performed as shown below. Here a = 11 and b = 11

1. A = 11 – 10 = 01; Subtract the multiplicand from the nearest base.
2. B = 11 – 10 = 01; Subtract the multiplier from the same base.
3. C = B x A = 1 x 1 = 1;
4. D = a + B = b + A = 11 + 1 = 100;
5. Result = 10 x D + C = 10 x 100 + 1 = 1001

For 3 bit multiplication consider the example of 101 x 110. Here we are having a = 101 and b = 110.

1. A = 101–100 = 01; Subtract the multiplicand from the nearest base
2. B = 110 – 100 = 10; Subtract the multiplier from the nearest base.
3. C = 10 x 01 = 10;
4. D = a + B = b + A = 101 +10 = 110 + 01 = 111;
5. Result = 100 x D + C = 100 x 111 + 10 = 11110

For 4 bit multiplication, let us consider an example of 1111 x 1111. Here a = b = 1111.

1. A = 1111 – 1000 = 111
2. B = 1111 – 1000 = 111;
3. C = A – 100 = 111 – 100 = 11;
4. D = B – 100 = 111 – 100 = 11;
5. E = C -10 = 11 – 10 = 1;
6. F = D – 10 = 11 – 10 = 1;
7. G = E x F = 1 x 1 = 1;
8. H = (C + F) x 10 + G = 1001;
9. I = (A + D) x 100 + H = 110001;
10. Result = (a + B) x 1000 + I = 11100001;

From the above example, we can see that multiplication is involved only in step 7 apart from that all other steps are having either shift, add or subtract operations. From this we can recognise that computation is simple if multiplicand and multiplier both are same.

Squaring using Nikhilam Algorithm

Nikhilam sutras has special advantage when both multiplicand and multiplier are same. If both are same then such operation is known as squaring. Nikhilam algorithm to calculate square of the binary integer is given below.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

$$x \times y = \frac{(x+y)^2 - (x-y)^2}{4}$$

Algorithm below describes the Nikhilam squaring. It takes input as a binary number of n–bits and produces square of as its output which can be up to 2n-bits [3]. First while-loop is used for the computation of the forward direction subtraction operations. Two counters and are used to keep track of processed input and proper base subtraction respectively. If loop is used to check whether the corresponding bit is 0 or 1. If base power to be subtracted is more than the number itself then else part of if-loop is executed. This happens to be onlywhen the first bit of the number is 0. Least significant bits multiplication is assigned in. Second while-loop is used for the computation of reverse direction for shifting and addition operations.

INPUT: $A = \sum_0^{n-1} a_i x^i$
OUTPUT: $B = A * A = \sum_0^{2n-1} b_k x^k$
$A_1 \leftarrow A$
$i \leftarrow 2, j \leftarrow n-1$
while$(i \le n$ **and** $j \ge 1)$
do
  if $(A_i > 2^j)$ **then**
    $A_i \leftarrow A_{i-1} - 2^j$
  else
    $A_i \leftarrow A_{i-1}$
  end if
$i \leftarrow i+1, j \leftarrow j-1$
end while
$B_1 = A_n * A_n$
$i \leftarrow 2, j \leftarrow n-1$
while$(i \le n$ **and** $j \ge 1)$
do
  if $(A_j \neq A_{j+1})$ **then**
    $B_i \leftarrow B_{i-1} + (A_j + A_{j+1})2^{i-1}$
  else
    $B_i \leftarrow B_{i-1}$
  end if
$i \leftarrow i+1, j \leftarrow j-1$
end while
return $B \leftarrow B_n$

Figure 1: Nikhilam Squaring Algorithm [3]

B. RECURSIVE ALGORITHM

This method completely avoids use of modified base method by selecting appropriate base and hence it avoids additional small multiplication. The bases in this method are in geometric progression as opposed to arithmetic progression.

The differences from the base may not be small as expected by Nikhilam sutra due to such base selection. Therefore in such case, the product of differences may itself be performed by recursively applying the proposed algorithm. Until the final product of difference is small as to be accomplished with multiple addition the process continues.

In this manner any two numbers, irrespective of their magnitude can be decomposed successively by the algorithm. Below is the Recursive algorithm which depends on the ratio of the number of 1s and the number of 0s in the binary representation of the higher of the two multiplicands.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Algorithm [2]

1. Of the two multiplicands, say X and Y, select the smaller multiplicand.
2. If either X or Y is 0 or 1 then go to step 6.
3. If the second MSB of this smaller multiplicand is 1 then select ‘Base’ B as 2^b else the select B as 2^ (b-1).
4. Subtract this base from both the multiplicands to get differences X' and Y'.
5. Apply steps 1 to 3 to X' and Y' recursively till X' = 1 or Y' = 1 OR X' = 0 or Y' = 0.
6. At recursions, get the partial result at each step by either X + Y' OR Y + X'.
7. Multiply X' and Y' in the final stage (where one of X' or Y' is either 0 or 1.
8. At each step the respective b no. of bits of the partial result are considered while other MSBs are carries generated.
9. Using step 8, merge the partial results to get the final product.

10111	111	11
10101	101	01
101100 11100	1000	11
	00	11
11110		
111100011		
23 x 21 = 483		

Table 7:Example of Recursive algorithm

III.CONCLUSION

In this paper possibility of applying the Nikhilam sutra of Vedic Mathematics has been explored. This sutra has an advantage of converting large digit multiplication to corresponding small digit multiplication. This sutra is basically more effective when both the multiplier and multiplicand are near to same base power.

REFERENCES

[1] Nikhil Mistri, Prof. S. B. Somani, “Design and Deployment of Vedic Multiplier using Vedic Mathematics- A review”, NCRTETIT 2016

[2] Ajinkya Kale, Shaunak Vaidya, Ashish Joglekar,” A Generalized Recursive Algorithm for Binary Multiplicationbased on Vedic Mathematics”

[3] Shri Prakash Dwivedi,” An Efficient Multiplication Algorithm Using Nikhilam Method”, arXiv:1307.2735v1 [cs.DS] 10 Jul 2013

[4] Can Eyupoglu,” Investigation of the Performance of Nikhilam MultiplicationAlgorithm”, Procedia - Social and Behavioral Sciences 195 (2015) 1959 – 1965

[5] Tirthaji, B.K.M.; “Vedic mathematics”, MotilalBanarsidass Publication, 1992.

[6] Yogita Bansal CharuMadhuPardeep Kaur,” HIGH SPEED VEDIC MULTIPLIER DESIGNSA REVIEW”, Proceedings of 2014 RAECS UIET Panjab University Chandigarh, 06 – 08 March, 2014

[7] Sumod Abraham Sukhmeet Kaur Shivani Singh,” Study of Various High Speed Multipliers”, 2015 International Conference on Computer Communication and Informatics (*ICCCI* -2015), Jan. 08 – 10, 2015, Coimbatore, INDIA.

[8] *S.ChinthanaiSelvi, Ms.R.Vigneshwari,Dr.G.Maryamirthasagayee,”* PERFORMANCE ANALYSIS OF MULTIPLIERUSING VARIOUS TECHNIQUES*”*, IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication SystemsICIIECS’15

[9] R. Sridevi, AnirudhPalakurthi, AkhilaSadhula, HafsaMahreen,” Design of a High Speed Multiplier (Ancient Vedic Mathematics Approach)”, International Journal of Engineering Research(ISSN: 2319-6890) Volume No.2, Issue No.3, pp : 183-186.

[10] GarimaRawat, KhyatiRathore, Siddharth Goyal, Shefali Kala and Poornima Mittal,” Design and Analysis of ALU: Vedic MathematicsApproach”, ICCCA2015