Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 4, April -2017

High Speed Vedic Multiplier Based Review with its Proposed Implementation for Parallel FIR Filter

[1]Vaibhav V. Manusmare, [2]Devendra S. Chaudhari

[1,2] Electronic and Telecommunication Engineering Department, Government College of Engineering, Amravati.

Abstract: *The performance of any processor depends upon its speed, area and delay. Multiplier play important role in any digital design. In Digital Signal Processor (DSP) application multiplier is main and essential functional block. Conventional multiplication process requires more iteration and hence more time for operation. To improve the performance of the processor its speed, area and power consumption need to be improved. Many researchers have been trying to design multipliers offering high speed, low power consumption, less area in one multiplier thus making it suitable for various high speed, low power application. For many DSP applications processors speed is the major concerned parameter compared to other useful parameters like area and power. Vedic mathematics is an ancient mathematics that has a unique method of computation based on simple rule. Urdhva Tiryagbhyam algorithm of Vedic mathematics is applicable for all case of multiplication. This paper reviews Vedic mathematics based high speed multiplier for Finite Impulse Response (FIR) filter design. The Digital filter design using Vedic mathematics in Field Programmable Logic Array (FPGA) facilitates high speed as compared to conventional method and other algorithm***.**

Keywords: *Vedic mathematics, array multiplier, Urdhva Triyagbhyam, Nikhilam algorithm, FIR filter.*

I. INTRODUCTION

In the modern era of technology, the most crucial parameters that require ultimate attention are speed and power consumption [1]. In Digital Signal Processing (DSP) application such as fast Fourier transform (FFT), discreet Fourier transform (DFT) etc, high speed and low power multipliers are of great importance. The multiplier is an essential hardware block of any computing system. Multipliers are the commonly used architectures inside the processor. Multiplier is also applicable in digital filtering [2] and various image processing application [8]. Since multiplication dominates the execution time of most DSP algorithms, use of high speed multiplier is very desirable. The conventional multiplication method is performed by successive summing and shifting operations. After each step of calculation partial product is generated. This requires more number of iteration. The number of gates requires for the operation also increases which in turns increase power consumption. This is the main factor that determines the performance of the multiplier. To reduce the iteration, time and power consumption, Vedic mathematics is better alternative. Vedic Multiplier is one of the fastest and low power consumption multiplier. It was found to perform better as compared to the other algorithm [3][4].

Vedic mathematics is based on the algorithms which are simple, powerful and logical. The techniques in Vedic mathematics are mainly based on sixteen sutras. Among this sixteen sutra's, Urdhva Tiryagbhyam and Nikhilam are used for multiplication [3]. Vedic multipliers provide best results compared to conventional ones and Urdhva Tiryagbhyam sutra architecture is more efficient compared to Nikhilam Sutra. Nikhilam Sutra based multipliers is more efficient for larger inputs, that is inputs near to base. In Urdhva Tiryagbhyam algorithm all partial products required for multiplication are generate in parallel fashion. These partial products are added to obtain final product leading to a very high speed approach. The speed of multiplier is determined by the speed of adders used for partial product addition. Regularity and simplicity of Vedic mathematics facilitate easy implementation in FPGA. This paper is organized into 7 sections. An introduction in first section is followed by background of Vedic mathematics. Third and fourth section describes a brief idea about conventional and Vedic multiplier respectively. Fifth section provides previous work in Vedic mathematics related to VLSI and DSP. Sixth section briefly describes outline of the proposed system. Conclusion is stated in the seventh section.

II. VEDIC MATHEMATICS

Vedic mathematics being an ancient mathematics provides unique technique of mental calculation based on simple rules and principles. 'Vedic' is term derived from the Sanskrit word 'Veda', which means collection of all knowledge. Veda rediscovered by the holiness Jagadguru Shree Bharti Krishna Tirtha Ji Maharaj (1884−1960) between 1911−1918. He discovered a set of 16 Sutras (aphorisms) and 13 Sub-Sutras (corollaries) from the Atharva Veda. He developed methods and techniques for amplifying the principles contained in the aphorisms and their corollaries, and called it Vedic mathematics [2]. The Sutras apply to almost every branch of mathematics. They apply even to complex problems involving a large number of mathematical operations. Application of the Sutras saves a lot of time and effort in solving the problems, compared to the formal methods. Though the solutions appear like magic, the application of the Sutras is perfectly logical and rational. The Sutras provide not only methods of calculation, but also ways of thinking for their application. It provides ease and quick computation of all the basic as well as complex mathematical operation.

III. CONVENTIONAL MULTIPLIER

1. Array Multiplier

Array multiplication is the traditional way of multiplication. It is based on add and shift algorithm. In this multiplication operation, number of partial products is added. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product is then shifted after the multiplication of one bit of multiplier with multiplicand. After the multiplication process of all the multiplicand bits with all the multiplier bits, finally all the partial products are added [5][6]. Addition is performed with normal carry propagate adder. Array multiplier consume more time for operation due to utilization of more numbers of gates.

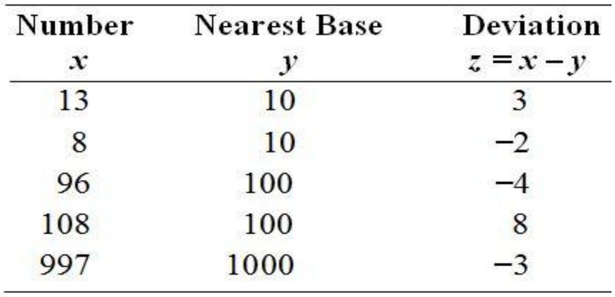
IV. VEDIC MULTIPLIER

1. Nikhilam Navatascaramam Dasatah

The formula simply means: “all from 9 and the last from 10.”

The formula can be very effectively applied in multiplication of larger numbers [9], which are nearer to bases like 10, 100, 1000 i.e., to the powers of 10. The procedure of multiplication using the Nikhilam involves minimum number of steps, space, time saving and only mental calculation. The numbers taken can be either less or more than the base considered. The difference between the number and the base is termed as deviation. Deviation may be positive or negative. Positive deviation is written without positive sign and the negative deviation is written with negative sign (or a bar on the number) as shown in Table I.

Table I. Deviation for Nikhilam Method



Number x	Nearest Base y	Deviation $z = x - y$
13	10	3
8	10	−2
96	100	−4
108	100	8
997	1000	−3

Example: Multiply 97 × 88

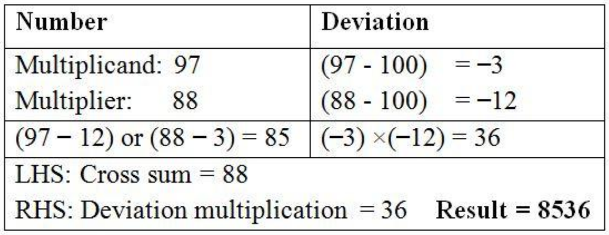
In this case, choose 100 as a base as this two numbers are nearest to 100. The multiplicand 97 is three less than base 100 and multiplier 88 is twelve less than 100. The deviation for 97 and 88 are 3 and 12 respectively. The product or answer will have two parts, one on the left side and the other on the right. The Right Hand Side (R.H.S.) of the answer is the product of the deviations of the numbers. Left Hand Side (L.H.S.) of the answer is the sum of one number with the deviation of the other. It can be arrived at in any one of the two ways [3].

i) Cross sum of deviation –12 in the second row with the original number 97 in the first row i.e. 97–12=85.

ii) Cross sum of deviation –3 in the first row with the original number 88 in the second row i.e. 88–3=85.

R.H.S. must contain number of digits equals to number of zeroes in the base. In case R.H.S. contains less number of digits than the number of zeroes in the base, the remaining digits are filled up by giving zero or zeroes on the left side of the R.H.S. otherwise the excess digit or digits are to be added to L.H.S of the answer if the number of digits is more than the number of zeroes in the base [1].

Table II. Multiplication of two Numbers near to base 100 using
Nikhilam method



Number	Deviation
Multiplicand: 97 Multiplier: 88	(97 - 100) = −3 (88 - 100) = −12
(97 − 12) or (88 − 3) = 85	(−3) ×(−12) = 36
LHS: Cross sum = 88 RHS: Deviation multiplication = 36 **Result = 8536**	

2. Urdhva Triyagbhyam

Urdhva Triyagbhyam is a multiplication algorithm in Vedic mathematics for the multiplication of decimal numbers. This is the general formula applicable to all cases of multiplication and also in the division of a large number by another large number. The meaning of this algorithm is "Vertical and Crosswise". The individual digits of both the operands are subjected to vertical and cross wise multiplication. Vedic multiplier is proved to be more efficient in terms of area and time delay.

For multiplication of two digits numbers the algorithm works as follows

Step 1: First digit of the multiplicand is multiplied by first digit of the multiplier vertically, forms the right hand most part of the answer.

Step 2: Diagonal multiplication of first digit of the multiplicand with second digit of the multiplier followed by multiplication the second digit of the multiplicand with first digit of the multiplier. Addition of these two numbers gives the next, i.e., second digit of the answer.

Step 3: Multiplication of second digit of the multiplicand with second digit of the multiplier vertically gives the left hand most part of the answer.

Schematically the process can be represented as shown in Fig.1

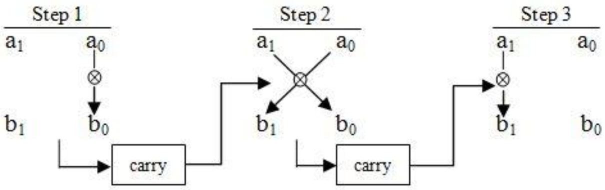


Fig. 1 Schematic representation of Urdhva Tiryagbhyam Method

Similarly multiplication can be done for any digital or binary numbers using Urdhva Triyagbhyam algorithm [5]. The block diagram for 4 bit multiplier using Urdhva Triyagbhyam algorithm is shown in Fig. 2

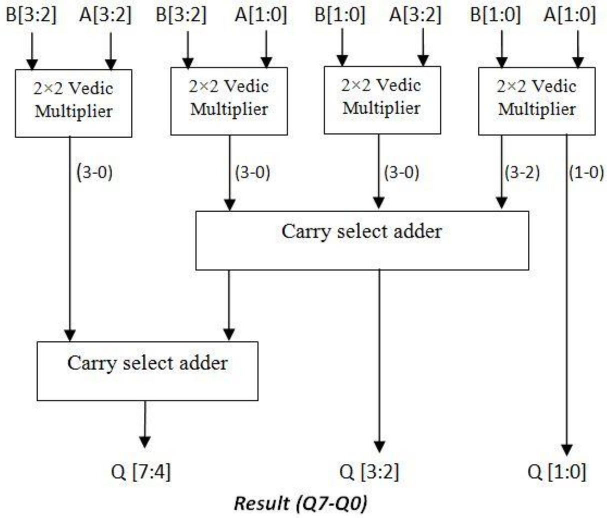


Fig.2 Block diagram of 4×4 bit Vedic multiplier

V. A REVIEW OF VEDIC MATHEMATICS WORK RELATED TO VLSI AND DSP

The requirement of high speed is on demand for various digital processing and image processing applications. Many researchers are trying to design multipliers with high speed, low power consumption, less area thus increasing the efficiency of the processor. The Vedic mathematics algorithm Urdhva Tiryagbhyam and Nikhiliam has found to be fast and simple compared to Booth and Array multiplier. Hence Vedic mathematics has high priority in many digital signal processing and image processing application as seen in most of the literature. Chopade and Mehta [3] presented performance analysis of both Nikhiliam and Urdhva Tiryagbhyam algorithm using VHDL language in Xilinx FPGA. It was observed that for 8 bit and 16 bit Urdhva Tiryagbhyam algorithm provides 50% improvement in delay than that of Nikhilam, whereas 100% better than that of binary multiplier. Also Nikhilam multiplier provides better performance than Urdhva and Array as number of bits increases. [1] have proposed the comparative study of two different algorithms, conventional Booth Multiplier and Vedic method Nikhilam using reversible logic gates. Both the multipliers are designed and implemented by using Xilinx 13.2 ISE simulator. They analyzed the results in terms of delay, area and power consumption, Booth algorithm requires more delay than the Nikhilam algorithm. The Nikhilam multiplier requires less number of adder cell hence less area of hardware than the Booth multiplier. Reversible logic gates reduce the power dissipation and make the system more efficient. 16x16 Multiplier using Urdhva Tiryagbhyam algorithm is presented in [5]. Later implementation of Vedic 16x16 multiplier using binary to excess converter (BEC) is designed [6]. The aim of using BEC is to increase the speed of operation and to reduce the usage of gates compared to Ripple Carry Adder (RCA).

Gavali and Kadam [10] introduced 11x8 bit multiplier using Vedic multiplier and implemented architecture using Xilinx ISE Design Suite 13.2 with Spartan 3E FPGA. They provided comparison of different Vedic multipliers in terms of speed and area. It has been observed that the proposed multiplier achieves a maximum clock frequency of 203.938 MHz which is an improvement over existing techniques. Sharma *et al*, [7] presented design of high speed multiplier and squaring architectures based upon Vedic mathematics sutras and realized using Xilinx Spartan-3E FPGA. They observed that these architectures offer great improvement in delay as the number of bit size increases. In [4] single precision and double precision floating point multipliers using Booth, *Karatsuba* and Vedic algorithm has been proposed. Vedic multiplier has shorter time delay as it utilizes less number of gates compared to Booth and *Karatsuba* algorithm.

A technique of Vedic mathematics has been used to reduce power in image processing application [8]. The Vedic algorithm is implemented in image filtering to enhance the image quality. This is useful in applications where the operating speed is of essence. In [9] Finite Impulse Response (FIR) low pass filter has been designed for order 20,34,68,128 using Nikhiliam, Urdhva Tiryagbhyam and conventional method with different window tequniues. It was observed that Nikhilam multiplier has better speed of response compared to Urdhva multiplier and conventional methods.

IV. PROPOSED IMPLEMENTATION

The proposed system describes an approach to the implementation of parallel FIR filter on FPGA using Vedic mathematics algorithm. The FPGA approach to digital filter implementation includes higher sampling rates than are available from traditional DSP chips, lower costs and more flexibility than the alternate approaches.

A large percentage of filters implemented in DSP are FIR filters. FIR filters are generally preferred in applications where there is requirement of linear phase over the whole frequency range. FIR filters are all zero filters and relation between its input and output is obtained by using the equation of linear convolution.

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \qquad (1)$$

In an FIR filter, since both input sequences *x* (n) and impulse response *h (n)* are finite duration sequences, their linear convolution *y (n)* is also finite in duration. Two parallel FIR filter consists of two filter inputs *(X0, X1)*, two filter coefficients *(H0, H1)* and two filter outputs *(Y0, Y1).*

Based on Fast FIR Algorithm, output of two parallel FIR filter written as

$$Y0 = \{\tfrac{1}{2}[(H0 + H1)(X0 + X1) + (H0 - H1)(X0 - X1)] - H1\,X1\} + z^{-2}\,H1\,X1$$

$$Y1 = \tfrac{1}{2}[(H0 + H1)(X0 - X1) - (H0 - H1)(X0 - X1)]$$

Fig. 3 shows implementation of two-parallel FIR filter structure based on symmetric convolution.

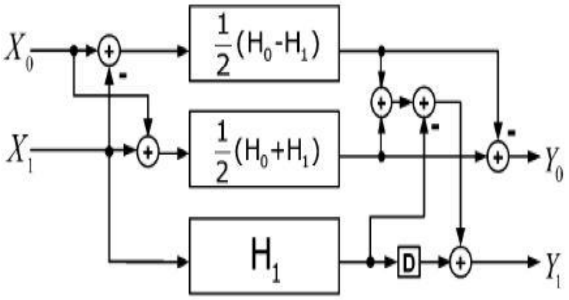


Fig.3 Two parallel symmetric FIR filter structure

Two parallel FIR filter structure has three sub-filter blocks. Out of which 2 sub-filter blocks *(H0 - H1)* and *(H0 + H1)* are equipped with symmetric coefficients. Similarly the structure can be implemented for three parallel FIR filter. Multipliers within each sub filter block will be utilized using Vedic mathematics.

An FIR filter design using Vedic mathematics containing carry select adder can also be modified using binary to excess converter which might give better performance in terms of speed of operation.

VII. CONCLUSION

The previous work on Vedic mathematics observed in VLSI and DSP suggests that Vedic multipliers are faster than the conventional multipliers. Urdhva Triyagbhyam algorithm increase speed and ultimately reduces the time delay. Nikhiliam algorithm is more efficient for larger numbers near to base 10 whereas Urdhva Triyagbhyam algorithm is suitable for all types of numbers. This effective algorithm can be applied to design of digital filters. As FIR filter mostly deals with multiplication, Vedic mathematics based algorithm can improve efficiency in terms of processing speed. It also facilitates reduced area, decreased power consumption.

REFERENCES

[1] H. P. Patil and S. D. Sawant, “Design and Implementation of Energy Efficient Vedic Multiplier using FPGA”, IEEE International Conference on Information Processing (ICIP) Vishwakarma Institute of Technology, pp. 206 − 210, 2015.

[2] T. Shukla, P. K. Shukla and H. Prabhakar, “High Speed Multiplier for FIR Filter Design using Window”, IEEE International Conference on Signal Processing and Integrated Networks (SPIN), pp. 486– 491, 2014.

[3] S. S. Chopade and R. Mehta, “Performance Analysis of Vedic Multiplication Technique using FPGA”, IEEE Bombay Section Symposium (IBSS), pp. 1–6, 2015.

[4] R. K. Kodali, L. Boppana and S. S.Yenamachintala, “FPGA Implementation of Vedic Floating Point Multiplier”, Informatics, communication and Energy Systems (SPICES), 2015 IEEE International Conference, pp.1– 4 , 2015.

[5] S. P. Pohokar, R. S. Sisal, K. M. Gaikwad, M. M. Patil and R. Borse, “Design and Implementation of 16x16 Multiplier Using Vedic Mathematics”, International Conference on Industrial Instrumentation and Control (ICIC), pp. 1174–1177, 2015.

[6] G. Ram, D. Rani, Y. Lakshmanna, K. Sindhuri, “Area Efficient Modified Vedic Multiplier”, IEEE International Conference on Circuit, Power and Computing Technologies [ICCPCT], 2016.

[7] R. Sharma, M. Kaur and G. Singh, “Design and FPGA Implementation of Optimized 32-Bit Vedic Multiplier and Square Architectures”, IEEE International Conference on Industrial Instrumentation and Control (ICIC), pp. 960–965, 2015.

[8] S. Jayakumar and S. Sumarthi, “High speed Vedic Multiplier for Image Processing using FPGA”, International conference on Intelligent System and Control (ISCO), pp. 1 – 4, 2016.

[9] S. Srivastava and D. Nagaria, “Design of High Performance FIR Filter Using Vedic Mathematics in MATLAB”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Issue 10, October 2014.

[10] K. R. Gavali and P. Kadam, “VLSI Design of High Speed Vedic Multiplier for FPGA Implementation”, IEEE International Conference on Engineering and Technology (ICETECH), pp. 936 – 939, 2016.