Circuits and Systems, 2016, 7, 2332-2338
Published Online July 2016 in SciRes. http://www.scirp.org/journal/cs
http://dx.doi.org/10.4236/cs.2016.79203

Scientific Research Publishing

Implementation of *N*-Bit Binary Multiplication Using *N* – 1 Bit Multiplication Based on Nikhilam Sutra and Karatsuba Principles Using Complement Method

M. Nisha Angeline[1*], S. Valarmathy[2]

[1]Department of ECE, Velalar College of Engineering and Technology, Erode, India
[2]Department of ECE, Bannari Amman Institute of Technology, Sathyamangalam, India
Email: *nishavlsidesign@gmail.com

Received 18 April 2016; accepted 10 May 2016; published 19 July 2016

Abstract

This paper is designed to introduce new hybrid Vedic algorithm to increase the speed of the multiplier. This work combines the principles of Nikhilam sutra and Karatsuba algorithm. Vedic Mathematics is the mathematical system to solve the complex computations in an easier manner. There are specific sutras to perform multiplication. Nikhilam sutra is one of the sutra. But this has some limitations. To overcome the limitations, this sutra is combined with Karatsuba algorithm. High speed devices are required for high speed applications with compact size. Normally multipliers require more power for its computation. In this paper, new multiplication algorithm for the multiplication of binary numbers is proposed based on Vedic Mathematics. The novel portion in the algorithm is found to be in the calculation of remainder using complement method. The size of the remainder is always set as $N - 1$ bit for any combination of input. The multiplier structure is designed based on Karatsuba algorithm. Therefore, $N \times N$ bit multiplication is done by $(N - 1)$ bit multiplication. Numerical strength reduction is done through Karatsuba algorithm. The results show that the reduction in hardware leads to reduction in the delay.

Keywords

Nikhilam Sutra, Numerical Strength Reduction, Karatsuba, Vedic Multiplier, Weight Reduction

*Corresponding author.

How to cite this paper: Nisha Angeline, M. and Valarmathy, S. (2016) Implementation of *N*-Bit Binary Multiplication Using *N* – 1 Bit Multiplication Based on Nikhilam Sutra and Karatsuba Principles Using Complement Method. *Circuits and Systems*, **7**, 2332-2338. http://dx.doi.org/10.4236/cs.2016.79203

1. Introduction

Researchers are trying to design devices which require minimum space and power with high speed. The multipliers are the important unit in many high speed applications. But it needs more components and consumes more power. From the conventional multipliers, Bough-Wooley consumes less power but the bit length is restricted to 16 bits. For high speed devices, Wallace with Booth encoding produces good result. But Wallace will occupy more space due to the usage of more components [1]. To overcome these issues, multiplier based on Vedic Mathematics is designed. In [1], the Urdhva Tiryakbhyam Vedic multiplier is designed with adiabatic logic. The logic cells used in the half adder, full adder and AND gate are replaced with 2P-2N logic. They focused to reduce the power.

In [2], vertical and cross wise algorithm is implemented using various compressor adders like 5-3 adders, 10-4 adders, 20-4 adders and 20-5 adders. The percentage improvement between the traditional adders and these compressor adders is much less. The compressor adders are used in the conventional multipliers to validate their proposed work. The result shows the significant improvement. The Vedic multiplier using McCMOS (Multi Channel CMOS) with 65 nm and 45 nm technology is proposed in [3] [4]. The power delay product is reduced from 48% to 70% using this technology.

The MAS (Multiplier Adder Subtractor) unit is incorporated [5] in the design of conventional ALU using Vedic Mathematics. The conventional ALU consists of Arithmetic Unit, Logic Unit and shifter module. MAS unit comprises all the necessary arithmetic modules to build arithmetic unit. In [6], 16 × 16 bit multiplier block is built, the functionality is verified in XC3STQ144 Xilinx kit, and the delay is compared with conventional multipliers. The final GDSII format is derived using Cadence tool.

The problem solving techniques using Vedic Mathematics not only reduce computational time but also give the way for effective learning. In [7], the arithmetic operations addition, subtraction, multiplication and division are performed using Nikhilam sutra. In [7], vinculum operations are explained and the method to find 10's complement if the number contains n zeros at the right side is well explained. In [8], Ekanyunena Purvena is explained and its architecture for binary numbers is given. Actually this is the sub sutra for Nikhilam. The important condition is one multiplicand which should contain array of 9 (*i.e*. 9 or 99 or 9999…). The multiplication is done through subtraction here. In [9], various conventional multipliers are compared with Vedic multiplier in terms of area, speed and power.

In [10], the Dadda multiplier is designed with pipelining. They modified the structure of D-Flipflop which is used for pipelining. And also sp-D3Lsum-based HA is used for tree reduction of Dadda algorithm. The design is implemented using 1P-9M Low-K UMC 90 nm CMOS process technology in Cadence Virtuoso. DRC and LVS check for the proposed design is done by Cadence Assura. In [11], the implementation of linear convolution and circular convolution is done using the Vedic multiplier. In [12], the Vedic multiplier is designed using Nikhilam sutra and Karatsuba algorithm. In that, the remainder calculation is done through the subtraction operation. The modification of the multiplier structure is done in [13]. Here the remainder is calculated by computing 2's complement of the input numbers. But in [13] [14], the inputs are swapped if the multiplier is greater than multiplicand. In this work, without swapping, the multiplication is done through the calculation of remainder using 2's complement method.

2. Proposed Architectures

The architecture is designed based on the combination of Karatsuba and Nikhilam sutra. in the conventional Karatsuba algorithm, the remainder is determined by taking Least Significant Half of the number without alteration. In the proposed work, the remainder is computed using Nikhilam Sutra. The detailed algorithm is given in [13] [14]. In this section, the proposed algorithms are presented and their architectures for three different modes are given. The results are proven theoretically in this section. Three modes are discussed in detail below.

Mode I—Positive Remainders
Mode II—Negative Remainders
Mode III—Mixed Remainders

2.1. Algorithm for Mode I

Input: *A*, *B*

Output: P

Step 1: Considering A and B are N bit numbers and having positive remainders (*i.e*. both are greater than 2^{N-1}). The positive remainders are derived by considering the numbers without MSB having $N-1$ bits. (Considering A_r and B_r)

Step 2: Multiplying the remainders A_r and B_r. *i.e*. $m_1 = r_1 * r_2$.

Step 3: Shifting the input A left side by $N-1$ times. ($m_2 = A \ll N-1$).

Step 4: Shifting the remainder of B, B_r by $N-1$ times ($m_3 = B_r \ll N-1$).

Step 5: Adding all the components to derive the final product $P = m_1 + m_2 + m_3$.

The proposed architecture for positive remainders is shown in **Figure 1**. As per the algorithm, the product is derived as follows

$$\begin{aligned} P = AB &= 2^{N-1} * A + A_r * B_r + 2^{N-1} * B_r \\ &= \left(A \ll N-1\right) + \left(A_r * B_r\right) + \left(B_r \ll N-1\right) \end{aligned} \tag{1}$$

In [13], the architecture of the algorithm is derived based on remainder. The remainder is derived by subtracting the highest weight by the numbers A or B.

2.2. Algorithm for Mode II

Input: A, B

Output: P

Step 1: Considering A and B are having negative remainders. (*i.e*. both are less than 2^{N-1}). The remainder is computed by complementing A & B with $N-1$ bits. (Consider A_r and B_r).

Step 2: Multiplying the remainders A_r and B_r. *i.e*. $m_1 = r_1 * r_2$.

Step 3: Shifting the input A left side by $N-1$ times ($m_2 = A \ll N-1$).

Step 4: Shifting the remainder of B, B_r by $N-1$ times ($m_3 = B_r \ll N-1$).

Step 5: Adding all the components to derive the final product $P = m_1 + m_2 - m_3$.

The architecture for negative remainders is shown in **Figure 2**. The product is determined as follows

$$\begin{aligned} P = AB &= 2^{N-1} * A + A_r * B_r - 2^{N-1} * B_r \\ &= \left(A \ll N-1\right) + \left(A_r * B_r\right) - \left(B_r \ll N-1\right) \end{aligned} \tag{2}$$

2.3. Algorithm for Mode III

Input: A, B

Output: P

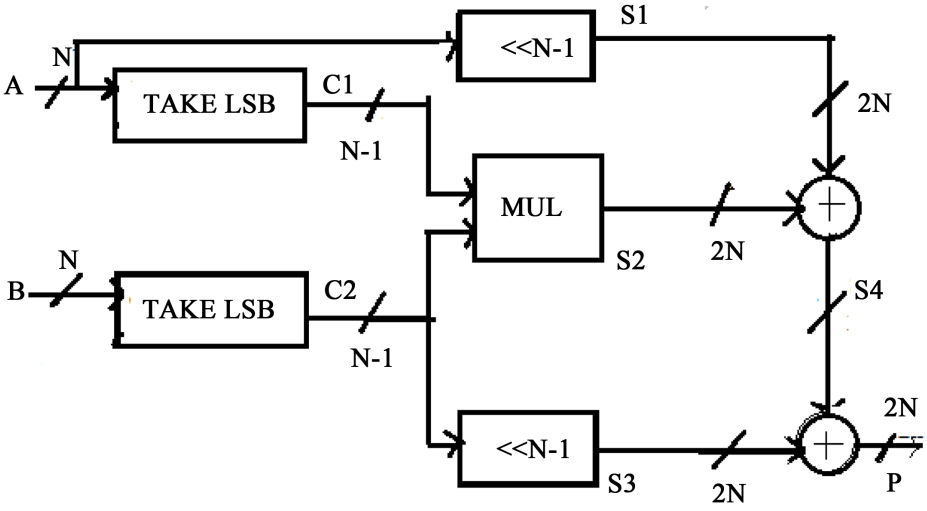

Figure 1. Proposed architecture for positive remainders.

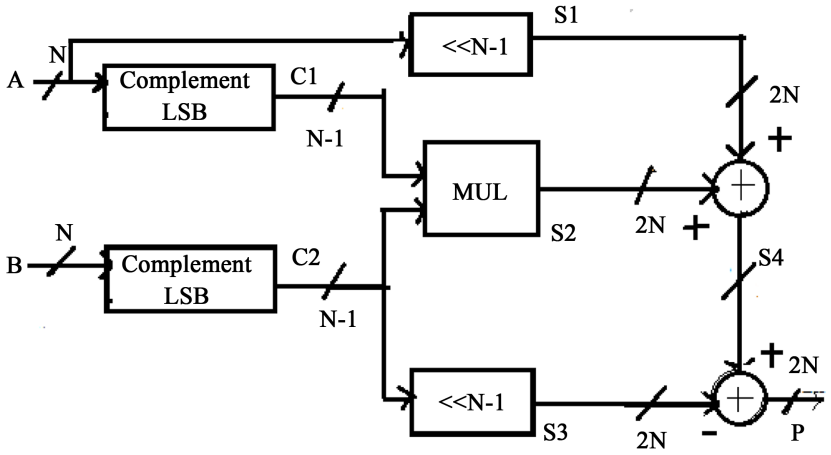

Figure 2. Proposed architecture for negative remainders.

Step 1: Considering A and B are having mixed remainders (*i.e.* one is positive remainder and the other is negative remainder). The positive remainder is derived as per Mode I and the negative remainder is calculated as per Mode II (consider A_r and B_r).

Step 2: Multiplying the remainders A_r and B_r. *i.e.* $m_1 = r_1 * r_2$. The product will be negative due to mixed remainders.

Step 3: Shifting the input A left side by $N-1$ times ($m_2 = A \ll N-1$).

Step 4: Shifting the remainder of B, B_r by $N-1$ times ($m_3 = B_r \ll N-1$). The sign of m_3 depends on the type of remainder B_r.

Step 5: Adding all the components to derive the product $P = m_1 - m_2 \pm m_3$.

The architecture for mixed remainder is shown in **Figure 3**. The product of the numbers is calculated as follows

$$\begin{aligned} P = AB &= \left(2^{N-1} * A\right) - \left(A_r * B_r\right) + \left(2^{N-1} * B_r\right) \\ &= \left(A \ll N-1\right) - \left(A_r * B_r\right) + \left(B_r \ll N-1\right) \end{aligned} \tag{3}$$

The input multiplexer is used here to derive the remainder. Based on MSB value of A and B, the remainder is calculated. For negative remainder, the complement of A is taken. For the positive remainder, the number is taken directly considering $N-1$ bits. The multiplier unit is used to multiply the remainder terms $A_r B_r$. m_1 is derived by shifting the value of A by $N-1$ bits. (*i.e.* $2^{N-1}A$). m_4 is the term that represents the multiplication of $2^{N-1}B_r$. The adder/subtractor is designed to select the operation based on the type of remainder (r_2). If $B_{N-1} = 1$, the remainder will be negative, adder/subtractor will perform subtraction operation. If $B_{N-1} = 0$, it will perform addition operation.

The combined structure is shown in **Figure 4**. Unlike [13] [14], no need to swap the inputs when one number is larger than the other. Using the combined structure, the number in any mode can be calculated. This structure is similar to the structure shown in **Figure 3**. A simple Ex-or gate is used as control signal to select addition/subtraction operation.

3. Results

The various conventional multipliers are considered and compared with proposed multiplier. The computational delay for various multipliers is listed in **Table 1**. The simulation result is shown in **Figure 5**. While comparing delay with other methods, Vedic multiplier has minimum delay among all methods and hence combination of proposed with conventional Vedic has been used for high speed applications. While comparing the area, Wallace will occupy more space because it requires more number of components. The comparison table for power analysis is shown in **Table 2**. By seeing both results, proposed Vedic multiplier is efficient in area and speed. Therefore, instead of using other methods in the proposed algorithm, the proposed algorithm is called in successive manner. The result for successive approximation of proposed algorithm is shown in **Table 3**.

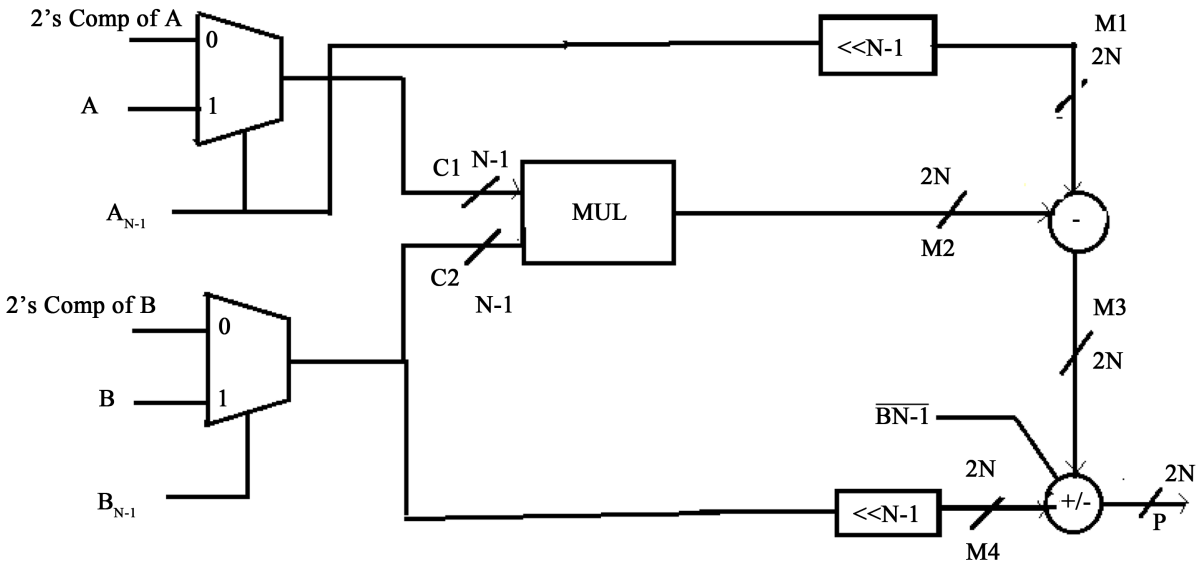

Figure 3. Proposed architecture for mixed remainders.

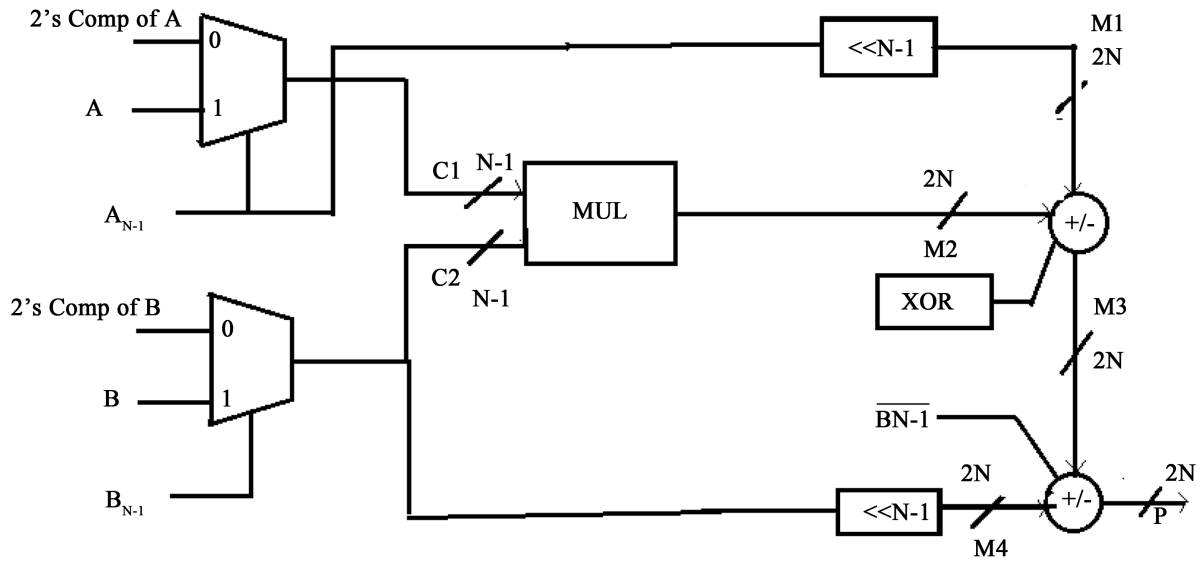

Figure 4. Proposed combined architecture of multiplier.

4. Conclusion and Future Work

In this paper, a new multiplication algorithm using Nikhilam sutra is presented. The modification of binary Vedic multiplier with the previous papers is presented here. In the calculation of remainder, a single bit is reduced, and hence usage of components will be reduced. Therefore, the interconnection delay and computation time are reduced. The speed and the area are optimized using this modified Vedic multiplier. The performance of the modified multiplier varies on the type of multiplier used. Finally successive approximation of proposed algorithm is also done here. Comparing with conventional methods, the combination of multiplier with Wallace multiplier gives reduced stage delay. But this combination consumes more power. Normally, Vedic multiplier is used to perform the operation with minimum delay. Therefore, in combination with conventional Vedic multiplier the proposed method gives better result. For high speed applications, proposed method with Wallace multiplier can be used. For low power and low area applications, proposed multiplier with Vedic (Urdhava) or Braun multiplier can be used. From the results it is clear that the proposed algorithm is best suited for high speed

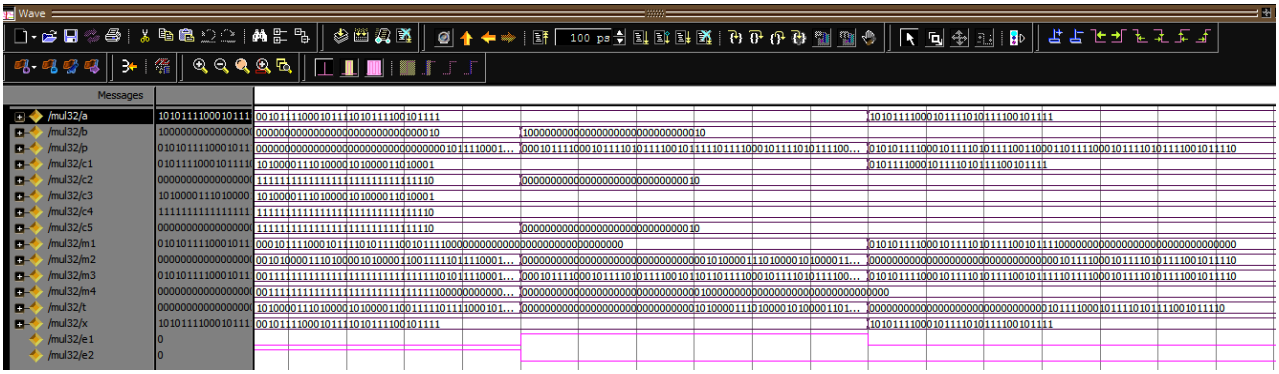

Figure 5. Simulation waveform for the multiplier with bit size 32.

Table 1. Comparison of delay with various methods.

Methods	Delay in nS			
	4 Bit	8 Bit	16 Bit	32 Bit
Array Multiplier	15.269	31.111	62.437	123.387
Vedic-Array	9.02	29.542	54.871	99.587
Shift and Add Multiplier	15.677	33.840	63.089	124.112
Vedic-SAA	15.92	33.324	62.254	122.567
Braun Multiplier	13.088	23.331	62.437	127.776
Vedic-Braun	9.325	18.198	34.562	88.547
Wallace Tree Multiplier	12.756	22.863	44.258	87.776
Vedic-Wallace	8.57	18.67	37.478	94.249
Vedic Multiplier	11.752	21.564	41.684	82.289
Vedic-Vedic	8.015	15.234	21.452	32.584

Table 2. Comparison of power with various methods.

Methods	Power in mW			
	4 Bit	8 Bit	16 Bit	32 Bit
Array Multiplier	298	312	368	440
Vedic-Array	122	259	485	621
Shift and Add Multiplier	310	368	501	636
Vedic-SAA	172	356	503	694
Braun Multiplier	113	149	406	706
Vedic-Braun	128	138	232	238
Wallace Tree Multiplier	113	154	375	706
Vedic-Wallace	111	145	347	674
Vedic Multiplier	123	142	250	489
Vedic-Vedic	119	130	212	443

Table 3. Comparison of delay using successive approximation method.

Methods	Delay in nS			
	4 Bit	8 Bit	16 Bit	32 Bit
Array Multiplier	15.269	31.111	62.437	123.387
Shift and Add Multiplier	15.677	33.840	63.089	124.112
Braun Multiplier	13.088	23.331	62.437	127.776
Wallace Tree Multiplier	12.756	22.863	44.258	87.776
Proposed Multiplier	8.021	14.654	21.475	30.542

and compact applications.

References

[1] Appasaheb, B.R. and Kanchana Bhaaskaran, V.S. (2013) Design and Implementation of an Efficient Multiplier Using Vedic Mathematics and Charge Recovery Logic. *Proceedings of International Conference on VLSI, Communication, Advanced Devices, Signals & Systems and Networking* (*VCASAN*-2013), *Volume 258 of the Series Lecture Notes in Electrical Engineering*, 101-108. http://dx.doi.org/10.1007/978-81-322-1524-0_15

[2] Bensal, Y. and Madhu, C. (2016) A Novel High-Speed Approach for 16 × 16 Vedic Multiplication with Compressor Adders. *Computers and Electrical Engineering*, **49**, 39-49. http://dx.doi.org/10.1016/j.compeleceng.2015.11.006

[3] Gupta, R., Dhar, R., Baishnab, K.L. and Mehedi, J. (2014) Design of High Performance 16 Bit Multiplier Using Vedic Multiplication Algorithm with McCMOS Technique. 2014 *International Conference on Green Computing Communication and Electrical Engineering* (*ICGCCEE*), Coimbatore, 6-8 March 2014, 1-6. http://dx.doi.org/10.1109/icgccee.2014.6922296

[4] Kayal, D., Mostafa, P., Dandapat, A. and Sarkar, C.K. (2013) Design of High Performance 8 Bit Multiplier Using Vedic Multiplication Algorithm with McCMOS Technique. *Journal of Signal Processing Systems*, **76**, 1-9. http://dx.doi.org/10.1007/s11265-013-0818-3

[5] Huddar, S.R., Rupanagudi, S.R., Janardhan, V., Mohan, S. and Sandya, S. (2013) Area and Speed Efficient Arithmetic Logic Unit Design Using Ancient Vedic Mathematics on FPGA. *Advances in Computing, Communication, and Control*, *Volume* 361 *of the Series Communications in Computer and Information Science*, 475-483. http://dx.doi.org/10.1007/978-3-642-36321-4_45

[6] Jagannatha, K.B., Lakshmisagar, H.S. and Bhaskar, G.R. (2013) FPGA and ASIC Implementation of 16-Bit Vedic Multiplier Using Urdhva Triyakbhyam Sutra. *Emerging Research in Electronics, Computer Science and Technology*, 248.

[7] Mehera, C. (2013) Computing Technique in Ancient India. *Bio-Inspired Computing and Applications*, *Volume* 6840 *of the Series Lecture Notes in Computer Science*, 282-289.

[8] Khan, A. and Das, R. (2015) Novel Approach of Multiplier Design Using Ancient Vedic Mathematics. *Information Systems Design and Intelligent Applications*, *Volume* 340 *of the Series Advances in Intelligent Systems and Computing*, 265-272. http://dx.doi.org/10.1007/978-81-322-2247-7_28

[9] Pratyusha, V.L.V., Narendra Babu, P.Y., Nivetha, S. and Jagadeesh, P. (2016) Comparison of Multipliers to Reduce Area and Speed. *Proceedings of the International Conference on Soft Computing Systems*, *Volume* 397 *of the Series Advances in Intelligent Systems and Computing*, 663-671. http://dx.doi.org/10.1007/978-81-322-2671-0_63

[10] Shabbir, Z., Ghumman, A.R. and Chaudhry, S.M. (2015) A Reduced-sp-D3Lsum Adder-Based High Frequency 4 × 4 Bit Multiplier Using Dadda Algorithm. *Circuits, Systems, and Signal Processing*, 1-22.

[11] Srimani, S., Kundu, D.K., Panda, S. and Maji, B. (2015) Implementation of High Performance Vedic Multiplier and Design of DSP Operations Using Vedic Sutra. *Computational Advancement in Communication Circuits and Systems*, *Volume* 335 *of the Series Lecture Notes in Electrical Engineering*, 443-449. http://dx.doi.org/10.1007/978-81-322-2274-3_49

[12] Thakare, L.P., Deshmukh, A.Y. and Khandale, G.D. (2014) VHDL Implementation of Complex Number Multiplier Using Vedic Mathematics. *Proceedings of International Conference on Soft Computing Techniques and Engineering Application*, *Volume* 250 *of the Series Advances in Intelligent Systems and Computing*, 403-410. http://dx.doi.org/10.1007/978-81-322-1695-7_46

[13] Nisha Angeline, M. and Valarmathy, S. (2016) Implementation of N-bit Binary Multiplication Using N-1 Bit Multiplication Based on Nikhilam Sutra Principles and Bit Reduction. *Transylvanian Review*, **XXIV**, 982-992.

[14] Nisha Angeline, M. and Valarmathy, S. (2016) Implementation of Hybrid Vedic Multiplier Nikhilam Sutra and Karatsuba Algorithm for N-Bit Multiplier Using Successive Approximation of N-1 Bit Multiplier. *Asian Journal of Information Technology*. (Accepted)