

An Efficient Approach to an 8-Bit Digital Multiplier Architecture based on Ancient Indian Mathematics

Shardul P. Telharkar (B.E.)
Dept. of Electronics and Telecommunications
K J Somaiya Autonomous College of Engineering
Mumbai – 400077, India.

Shantanu P. Telharkar (B.E.)
Dept. of Electronics
K J Somaiya Autonomous College of Engineering
Mumbai – 400077, India.

Raj D. Pednekar (B.E.)
Dept. of Electronics and Telecommunications
K J Somaiya Autonomous College of Engineering
Mumbai – 400077, India.

Abstract—This paper encompasses the implementation of a novel approach to an 8-bit digital multiplier based on Ancient Indian mathematical algorithm (also known as the Vedic sutra) and its comparison with 2 other conventional multipliers. Since multiplication has become a fundamental function in applications such as Fourier transform, Arithmetic Logic Unit (ALU) architecture and in various sequential clocked circuits, an efficient multiplication technique is a major concern for computer architects. The three multipliers were programmed in Verilog, simulated on Xilinx 14.5 ISE and implemented on 2 FPGA devices, Spartan 3E and Spartan 6. A comparison based on empirical values of delays of each multiplier throws a light on their potential in making the digital circuits substantially efficient.

Keywords— Barrel shifter, Base computing module, Booth's algorithm, Nikhilam Navatashcaramam Dashatah, Power computing module, Propagation delay, Urdhwa-Tiryakbham.

I. INTRODUCTION

To begin with, we are going to compare three multipliers in this paper, out of which two multipliers are based on algorithms of Vedic mathematics while the third one is based on Booth's algorithm.

The system of Vedic mathematics was rediscovered from Ancient Indian Sanskrit texts known as Vedas between 1911 and 1918. According to this system complete mathematics is based on sixteen formulas known as "Sutras".

The sixteen sutras and their literal meanings are as follows [4]:

1. Ekadhikina Purvena -By one more than the previous one
2. Nikhilam Navatashcaramam Dashatah -All from 9 and the last from 10
3. Urdhva-Tiryakbham-Vertically and crosswise
4. Paraavartya Yojayet-Transpose and adjust
5. Shunyam Saamyasamuccaye-When the sum is the same, that sum is zero.

6. (Anurupye) Shunyamanyat-If one is in ratio, the other is zero
7. Sankalana-vyavakalanabhyam-By addition and by subtraction
8. Puranapuranyam-By the completion or non-completion
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 second and third sutras are going to be utilized here, Urdhva-Tiryakbham and Nikhilam Navatashcaramam Dashatah respectively.

A. Nikhilam Navatashcaramam Dashatah:

Nikhilam Sutra stipulates subtraction of a number from the nearest power of 10 i.e. 10, 100, 1000, etc. The powers of 10 from which the difference is calculated is called the Base. These numbers are considered to be references to find out whether given number is less or more than the Base. The difference between the base and the number is called NIKHILAM. The value of Nikhilam may be reference base, the Nikhilam of 87 is -13 and that of 113 is +13 respectively [1]. This algorithm has been discussed in detail in the next section.

B. Urdhwa-Tiryakbham:

Multiplication by this technique of a 2 figure integer by another 2 figures integer is done as follows [4], [6], [8]:

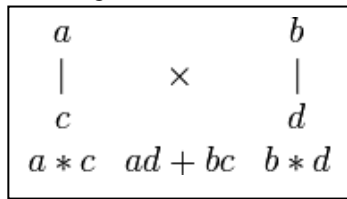


Fig. 1. Illustration of a 2-bit multiplication using ‘Urdhwa-Tiryakbham’ method.

From Fig. 1, if two numbers are given as $(10a + b)$ and $(10c + d)$, their multiplication can be calculated as:

$$(10a + b)(10c + d) = 100(c * d) + 10(ad + bc) + (b * d) \tag{1}$$

Equation (1) forms the basis of the ‘Urdhwa-Tiryakbham’ method, and likewise for 8-bit multiplier that we have designed, the algorithm expands on similar lines. However, the main focus of this paper remains on the second Vedic sutra, since our proposed architecture has been derived from this algorithm- ‘Nikhilam Navatashcaramam Dashatah’.

II. MATHEMATICAL BACKGROUND

A. Nikhilam-Vedic algorithm:

The Vedic mathematical sutra of ‘Nikhilam Navatashcaramam Dashatah’ calculates the product as follows:

Let A and B be two numbers to be multiplied (where the greater number is always taken as A).

If 10^R is their common base, the numbers can be expressed as [8]:

$$A = 10^R - s_1 \tag{2}$$

$$B = 10^R - s_2 \tag{3}$$

(where s_1 and s_2 are termed as their ‘distance from base’)

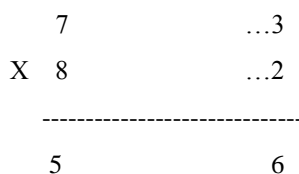
The product P can be calculated as:

$$P = (10^R)(A - s_2) + (s_1 \times s_2) \tag{4}$$

OR

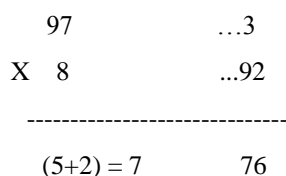
$$P = (10^R)(B - s_1) + (s_1 \times s_2) \tag{5}$$

Example 1 (base 10):



Hence, $7 \times 8 = (5 \times 10) + 6$.

Example 2 (base 100):



Hence, $97 \times 8 = ((5 + 2) \times 100) + 76 = 776$.

B. Novel Approach to the Vedic algorithm:

One can apparently observe from example 2, that keeping a common base increases the multiplication complexity manifolds. This paper harnesses a novel approach of multiplication by taking into account two different bases and implementing an architecture that provides a reduced delay for arriving at an accurate answer.

Consider two numbers A and B again, A being the larger one.

Now, if 10^M and 10^N are their nearest bases respectively, the numbers can be expressed as:

$$A = 10^M - d_1 \tag{6}$$

$$B = 10^N - d_2 \tag{7}$$

(where d_1 and d_2 are termed as their ‘distance from base’)

The product P is calculated as follows [3]:

$$P = (10^N)(A - d_2(10^{M-N})) + (d_1 \times d_2) \tag{8}$$

OR

$$P = (10^N)(B(10^{M-N}) - d_1) + (d_1 \times d_2) \tag{9}$$

From equations (8) and (9) it is clear that $d_2(10^{M-N})$ or $B(10^{M-N})$ is merely a left shift. However, one needs to understand that this approach takes into consideration the difference in bases and hence computational complexity reduces drastically. Should M be equal to N, equation (8) takes the form of equation (4) while equation (9) takes the form of equation (5).

To arrive at the perfect output, either we scale up the value of d_2 (that is distance of smaller number from its base) to make it comparable to the larger number or we scale up the value of the smaller number B itself to make it comparable to d_1 (that is distance of the larger number from its base). This is the fundamental difference between equations (8) and (9). To implement this algorithm, we have followed equation 8. However, one should note that both the ways are correct and none of the ways affects the answer in anyway.

C. Theory of Bases:

In order to implement the ‘sutra’ as a digital multiplier, we need to know the exact number of digits that the product will contain so that we can carry forward the exceeding digits. After a thorough observation and study, we have come to generalize a rule for the number of digits contained in the product of ‘Nikhilam’ algorithm based multiplication.

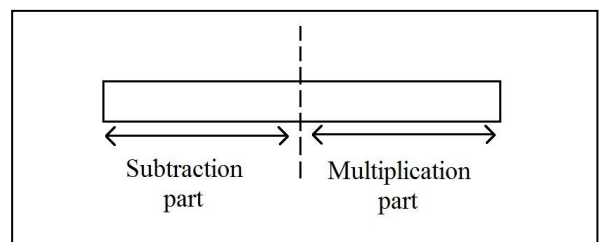


Fig. 2. Illustration of sectional parts of the final product As shown in Fig. 2, the whole product is divided into 2 parts, namely,

1) *Subtraction part.*

2) *Multiplication part.*

The parts are so named, because that part of the product is obtained by subtraction of d_2 from A and by multiplication of terms d_1 and d_2 respectively.

Considering A and B as input numbers and 10^M and 10^N as the bases respectively, we define 2 rules,

(1) The number of digits in the subtraction part is equal to the power index of the base of the larger number (or A).

(2) The number of digits in the multiplication part is equal to the power index of the base of the smaller number (or B).

Thus we must right shift the subtracted answer by that number which is the power index of the base of the smaller number. This rightly justifies the first term in equations 8 and 9.

III. PROPOSED ARCHITECTURE (NIKHILAM VEDIC ALGORITHM BASED MULTIPLIER)

A. Elementary Modules:

Architecturally, a digital multiplier by our approach has 2 elementary modules:

1) Base computing module (BCM):

This module simply accepts an 8-bit number [1] that is input to the multiplier and generates its base number. If a base is itself input to this module, the same number will appear at the output.

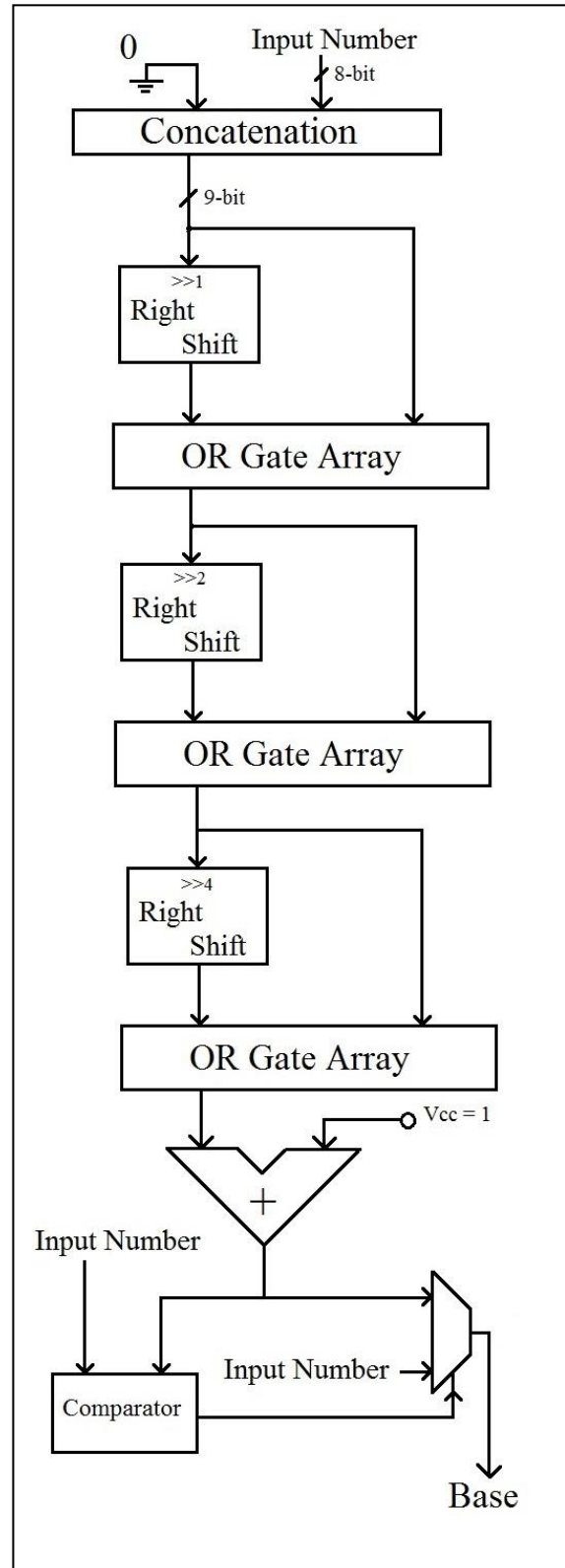


Fig. 3. Base Computing Module (BCM) architecture using Bit-twiddling algorithm

As seen from Fig. 3, the module accepts an 8-bit input number and computes a 9-bit base number. It has been programmed by using the 'bit-twiddling' algorithm. Successive approximation by arithmetic right shift operation, yields either the same input number (in cases when the input number is a base itself) or its base number (in cases when the input number is not a base). The final multiplexer decides between these two inputs.

2) Power-index computing module (PCM):

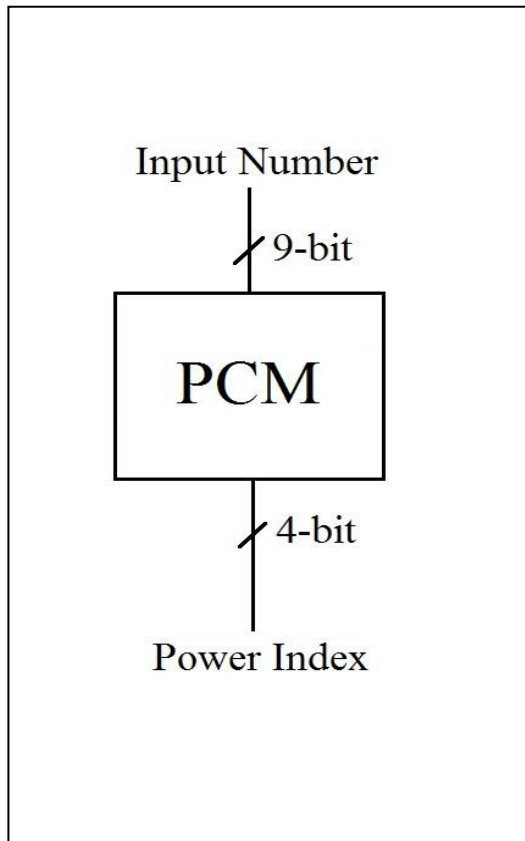


Fig. 4. Power index Computing Module (PCM)

As illustrated in Fig. 4, this module is primarily used for [1] acquiring the power index of the bases of the 2 numbers. Since a base can attain a highest value of $(9)_{10}$ or $(1001)_2$, output of PCM is a 4-bit value.

B. Main Architecture:

As explained in Section 2.3, [1] the subtraction part and multiplication part are computed separately by the multiplier and later added together.

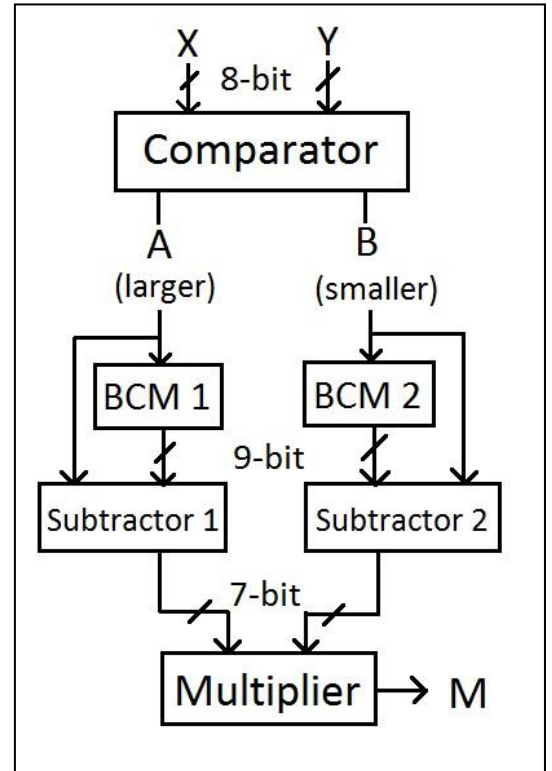


Fig. 5. Architecture for calculation of 'Multiplication part' of the final Product P.

As seen from the Fig. 5, the multiplication of d_1 and d_2 from equation (8) is done in this stage. In a conventional way, this part is termed as M. (which will be added later to part S). It is a 14-bit result.

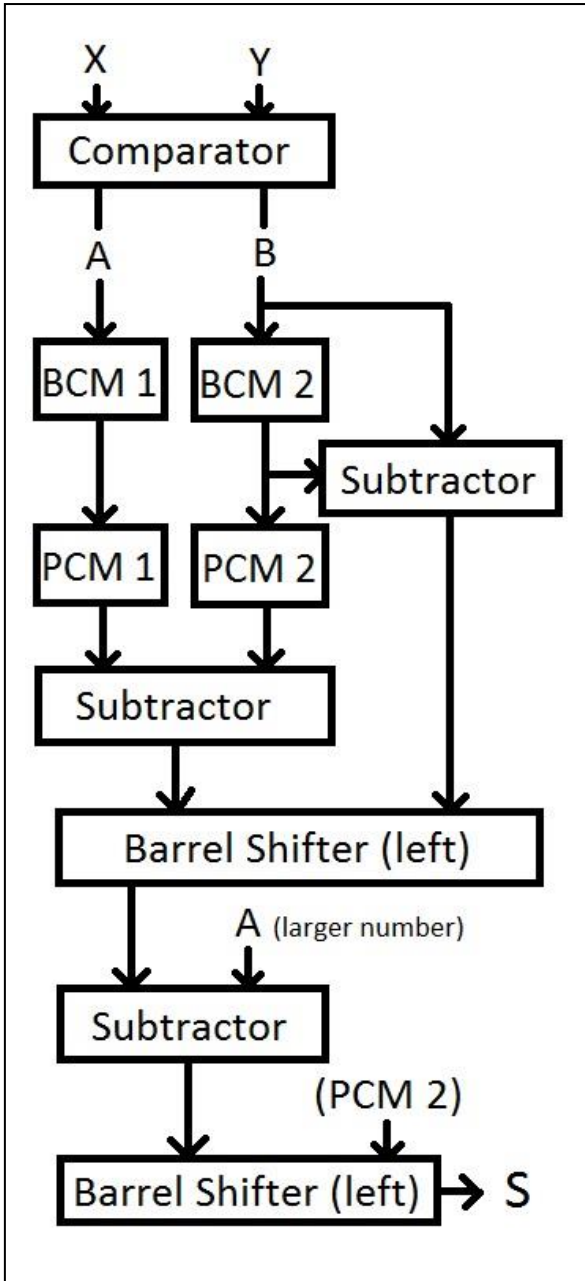


Fig. 6. Architecture for calculation of 'Subtraction part' of the final Product P.

The Fig. 6 above illustrates how calculation of $(A - d_2 (10^{M-N}))$ is implemented. M and N are first acquired from the PCM modules. The subtracted result is forwarded to a barrel shifter that 'scales up' the value of d_2 , in order to make it comparable with A. Value of $(A - d_2)$ is then barrel shifted that number of times which is the power index of the base of smaller number. This provides us the second input to the final adder. The result is termed as 'Subtraction Part' or simply 'S'. Note that, although Fig. 5 and Fig. 6 both show a Comparator and 2 BCM modules, in practice, the hardware is common and least amount of hardware has been used by our architecture.

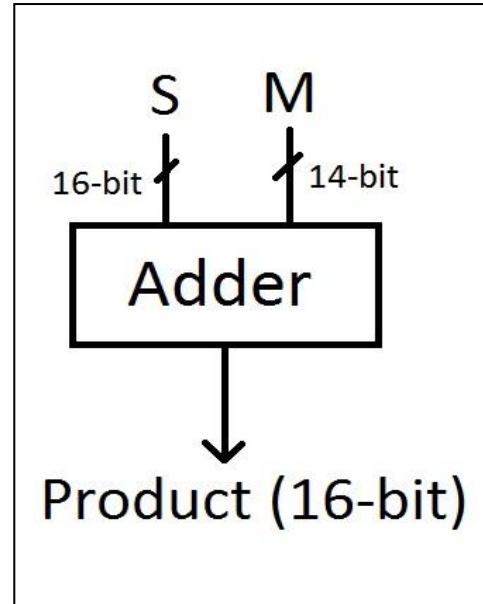


Fig. 7. Final Adder performing $S + M = P$

The adder produces the ultimate 16-bit product of the two input numbers 'S' and 'M'.

C. Output Simulation on Xilinx 14.5 ISE iSim:

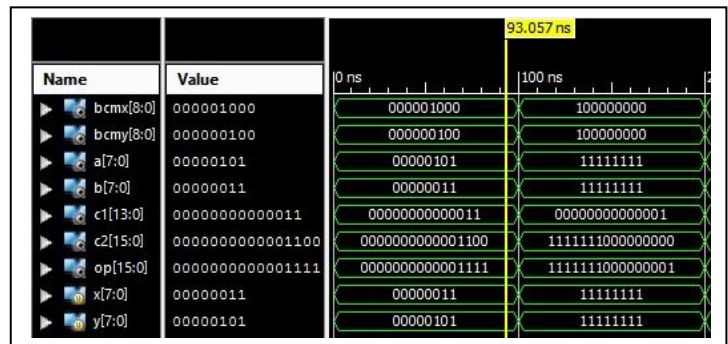


Fig. 8. Output Waveforms of the Proposed Nikhilam algorithm based multiplier.

Fig. 8, shows the multiplication of 2 numbers $(5)_{10}$ and $(3)_{10}$ which are $(101)_2$ and $(11)_2$ respectively. The 16-bit answer that the simulator yields is thus $(000000000001111)_2$.

It can be seen that exactly like Fig. 5 and Fig. 6, the input numbers are X and Y. But since we want the former number to be greater, we pass them through comparator and name the greater input number as A and the later as B. Further from Fig. 8, 'bcmx' and 'bcmy' are the bases of the numbers A and B respectively. 'c1' and 'c2' are the multiplication and the subtraction parts calculated as per Fig. 5 and Fig. 6. Output 'op' is calculated simply by adding 'c1' and 'c2'.

IV. CONCLUSION

Three multiplier designs were programmed in Verilog for comparing them on the basis of delay or speed. To get appropriate results, 2 Xilinx FPGA Devices were used- Spartan 3E and Spartan 6.

The three designs are:

A. *Multiplier based on Vedic sutra of 'Urdhwa Tiryakbham'.*

B. *Multiplier based on 'Booth's Algorithm'.*

C. *Proposed multiplier based on a novel approach using Vedic sutra of 'Nikhilam Navatashcaramam Dashatah'.*

The empirical results provided by Xilinx 14.5 ISE:

TABLE I COMPARISON BASED ON MAXIMUM COMBINATIONAL DELAY IN NANoseconds.

| FPGA Family | Multipliers Implemented | | |
|-------------|---|---|---|
| | <i>8-bit Multiplier (Urdhwa-Tiryakbham) (Delay)</i> | <i>8-bit Multiplier (Booth's Algorithm) (Delay)</i> | <i>8-bit Proposed Multiplier (Nikhilam Vedic Sutra) (Delay)</i> |
| Spartan 3E | 27.978 ns | 34.142 ns | 26.562 ns |
| Spartan 6 | 19.914 ns | 21.669 ns | 16.647 ns |

From the results, we conclude the delay of the proposed multiplier is lesser than both the other multipliers considered for comparison. Booth's algorithm was referred from [7]. On Xilinx Spartan 3E the improvement in speed from 'Urdhwa Tiryakbham' based multiplier is 5.06 % while on Xilinx Spartan 6 the improvement is 16.4 %. Since speed of a digital circuit is the principal concern to enhance performance, we conclude that the proposed architecture is an efficient approach for implementing 8-bit digital multiplication than a conventional multiplier harnessing 'Urdhwa Tiryakbham' algorithm or a multiplier harnessing commonly followed

'Booth's algorithm'. Further, looking at the results of both the FPGA devices, it can be deduced that better is the hardware on which the architectures are implemented, better is the improvement achieved in terms of speed.

REFERENCES

- [1] Pavan Kumar, Saiprasad Goud, A.Radhika, "FPGA Implementation of high speed 8-bit Vedic multiplier using barrel shifter". Kumar, U.C.S.P. ;Goud, A.S. ; Radhika, A. Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference. DOI: 10.1109/ICEETS.2013.6533349, Publication Year: 2013 , Page(s): 14 - 17
- [2] "Cmos Vlsi Design: A Circuits And Systems Perspective, 3/E" Authors – Neil H.E. Weste, David Harris and Ayan Banerjee. Publications - Pearson Education.
- [3] Arindam Banerjee, Debesh Kumar Das, "The Design of Reversible Multiplier Using Ancient Indian Mathematics". ISED '13 Proceedings of the 2013 International Symposium on Electronic System Design, 2013-12-10. IEEE Computer Society Washington, DC, USA, 2013. ISBN-978-1-4799-2301-4. Pages 31-35
- [4] Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho – "Multiplier design based on ancient Indian Vedic Mathematics". SoC Design Conference, 2008. ISOC '08. International Volume: 02 DOI: 10.1109/SOCD.2008.4815685, Publication Year: 2008 , Page(s): II-65 - II-68 Cited by: Papers (14)
- [5] Patterson, David and John Hennessy. "Computer Organization and Design: The Hardware/Software Interface", Second Edition. ISBN 1-55860-428-6. San Francisco, California: Morgan Kaufmann Publishers. 1998.
- [6] "Implementation of high speed low power combinational and sequential circuits using reversible logic." Shah, H.; Rao, A.; Deshpande, M.; Rane, A.; Nagvekar, S.. Advances in Electrical Engineering (ICAE), 2014 International Conference on Year: 2014. Pages: 1 - 4, DOI: 10.1109/ICAE.2014.6838457.
- [7] Andrew D. Booth. "A signed binary multiplication technique." The Quarterly Journal of Mechanics and Applied Mathematics, Volume IV, Pt. 2, 1951 [1]
- [8] Prabir Saha, Arindam Banerjee, Partha Bhattacharyya, Anup Dandapat, "High speed ASIC design of complex multiplier using vedic mathematics" , Proceeding of the 2011 IEEE Students' Technology Symposium 14-16 January, 2011, IIT Kharagpur, pp. 237-241.
- [9] Jagadeshwar Rao M, Sanjay Dubey," A High Speed Wallace Tree Multiplier Using Modified Booth Algorithm for Fast Arithmetic Circuits" IOSR Journal of Electronics and Communication Engineering (IOSRJECE), Volume 3, PP 07-11, Issue 1 (Sep-Oct 2012).