ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design of Low Power and High Speed Modified Carry Select Adder for 16 bit Vedic Multiplier

Sameena Begum
M.Tech,
JNIT, Hyderabad.

K.Venkateshwarlu, M.Tech
Associate professor,
JNIT, Hyderabad.

Abstract:

A high speed and low power 16×16 Vedic Multiplieris designed by using low power andhigh speed modified carry select adder.This work proposes an high speed Vedic Multiplier based on area, delay and power efficient Carry Select Adder. In this paper a fast method of multiplication based on ancient Indian Vedic mathematics is proposed. The whole of Vedic mathematics is based on 16 sutras and manifests a unified structure of mathematics. Among the various methods of multiplication in Vedic mathematics, Urdhava tiryakbhyam is discussed in detail. All the redundant logic operations present in the conventional CSLA are eliminated and proposed a new logic formulation for CSLA. The proposed CSLA design involves significantly less area and delay than the recently proposed BEC-based CSLA. The multiplier discussed here is compared with other multiplier to highlight the speed and power superiority of the vedic multiplier.

Index Terms:

Carry Select Adder, Logic formulation, Vedic Multiplier.

I.NTRODUCTION:

Multiplication is one of the fundamental block in almost all the arithmetic logic units. This Vedic multiplication is mainly used in the fields of the Digital Signal Processing (DSP) and also in so many applications like Fast Fourier Transform, convolution, filtering and microprocessor applications[2,3,9]. In most of the DSP algorithms multiplier is one of the key component and hence a high speed and area efficient multiplier is needed and multiplication time is also one of the predominant factor for DSP algorithms. The ancient mathematical techniques like Vedic mathematics used to reduce the computational time such that it can increases speed and also requires less hardware.

There are sixteen sutras and sixteen sutras (sub formulae) constructed by swahiji. Vedic is a word obtained from the word "Veda" and its meaning is "store house of all knowledge". Vedic mathematics mainly consists of the 16 sutras which it can be related to the different branches of mathematics like algebra, arithmetic geometry.

II. ANCIENT VEDIC MATHEMATICAL ALGORITHMS:

The Vedic mathematics mainly reduces the complex typical calculations in to simpler by applying sutras as stated above. These Vedic mathematic techniques are very efficient and take very less hardware to implement. These sutras are mainly used for multiplication of two decimal numbers and we extend these sutras for binary multiplications. Some of the techniques are discussed below.

A. Urdhva -Tiryagbhyam Sutra (Vertically and Crosswise):

Booth multipliers are generally used for multiplication purposes. Booth Encoder, Wallace Tree, Binary Adders and Partial Product Generator are the main components used for Booth multiplier architecture. Booth multiplier is mainly used for 2 applications are to increase the speed by reduction of the partial products and also by the way that the partial products to be added. In this section we propose a Vedic multiplication technique called "Urdhva-Tiryakbhyam – Vertically and crosswise." Which can be used not only for decimal multiplication but also used for binary multiplication? This technique mainly consists of generation of partial products parallel and then we have to perform the addition operation simultaneously[3]. This algorithm can be used for 2x2, 4x4, 8x8....N×N bit multiplications. Since the sums and their partial products are calculated in parallel the Vedic multiplier does not depends upon the processor clock frequency.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Hence there is no need of increasing the clock frequency and if the clock frequency increases it will automatically leads to the increase in the power dissipation. Hence by using this Vedic multiplier technique we can reduces the power dissipation. The main advantage of this Vedic multiplier is that it can reduces delay as well as area when compared with the other multipliers.

B. Example for Decimal Multiplication Using Vedic Mathematics:

To illustrate this technique, let us consider two decimal numbers 252 and 846 and the multiplication of two decimal numbers 252×846 is explained by using the line diagram shown in below figure1. First multiply the both numbers present on the two sides of the line and then first digit is stored as the first digit of the result and remaining digit is stored as pre carry for the next coming step and the process goes on and when there is more than one line then calculate the product of end digits of first line and add the result to the product obtained from the other line and finally store it as a result and carry. The obtained carry can be used a carry for the further steps and finally we will get the required result which is the final product of two decimal numbers 252x846. Take the initial carry value as the zero. For clear understanding purpose we explained the complete algorithm in the below line diagram such that each bit represents a circle and number of bits equal to the number of circles present.

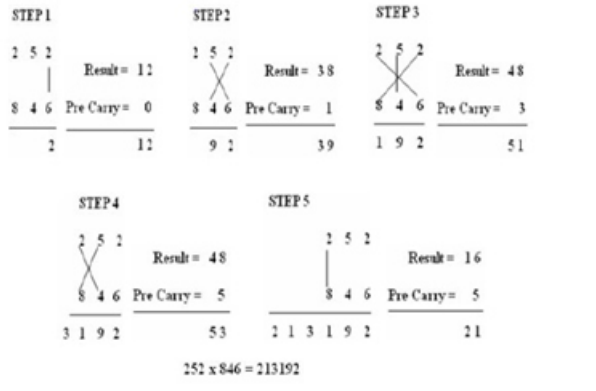


Figure.1. Multiplication of two decimal numbers

III. MODIFIED VEDIC MULTIPLIER ARCHITECTURE:

The architectures for 2×2, 4×4, 8×8, 16×16. . .N×N bit modules are discussed in this section.

In this section, the technique used is 'Urdhva-Tiryakbhyam' (Vertically and Crosswise) sutra which is a simple technique for multiplication with lesser number of steps and also in very less computational time. The main advantage of this Vedic multiplier is that we can calculate the partial products and summation to be done concurrently. Hence we are using this Vedic multiplier in almost all the ALU's.

A. 2×2 Vedic Multiplier Block:

To explain this method let us consider 2 numbers with 2 bits each and the numbers are A and B where A=a0a1 and B=b0b1 as shown in the below line diagram. First the least significant bit (LSB) bit of final product (vertical) is obtained by taking the product of two least significant bit (LSB) bits of A and B is a0b0. Second step is to take the products in a crosswise manner such as the least significant bit (LSB) of the first number A (multiplicand) is multiplied with the next higher bit of the multiplicand B in a crosswise manner. The output generated is 1-Carry bit and 1bit used in the result as shown below. Next step is to take product of 2 most significant bits (MSB) and for the obtained result previously obtained carry should be added. The result obtained is used as the fourth bit of the final result and final carry is the other bit.

$$s0 = a0b0 \quad (1)$$
$$c1s1 = a1b0 + a0b1 \quad (2)$$
$$c2\ s2 = c1 + a1b1 \quad (3)$$

The obtained final result is given as c2s2s1s0. A 2×2 Vedic multiplier block is implemented by using two half adders and four two input and gates as shown in below Figure 2.

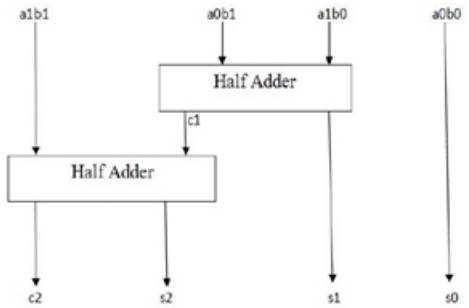


Figure .2. Block Diagram of 2×2 Vedic Multiplier

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

B. 4x4 Vedic Multiplier Block:

In this section, now we will discuss about 4x4 bit Vedic multiplier. For explaining this multiplier let us consider two four bit numbers are A and B such that the individual bits can be represented as the A3A2A1A0 and B3B2B1B0. The procedure for multiplication can be explained in terms of line diagram shown in below figure. The final output can be obtained as the C6S6S5S4S3S2S1S0.

The partial products are calculated in parallel and hence delay obtained is decreased enormously for the increase in the number of bits. The Least Significant Bit (LSB) S0 is obtained easily by multiplying the LSBs of the multiplier and the multiplicand. Here the multiplication is followed according to the steps shown in the line diagram in figure 3. After performing all the steps the result (Sn) and Carry(Cn) is obtained and in the same way at each step the previous stage carry is forwarded to the next stage and the process goes on.

$$S0 = A0B0 \quad (4)$$
$$C1S1 = A1B0 + A0B1 \quad (5)$$
$$C2S2 = C1 + A0B2 + A2B0 + A1B1 \quad (6)$$
$$C3S3 = C2 + A0B3 + A3B0 + A1B2 + A2B1 \quad (7)$$
$$C4S4 = C3 + A1B3 + A3B1 + A2B2 \quad (8)$$
$$C5S5 = C4 + A3B2 + A2B3 \quad (9)$$
$$C6S6 = C5 + A3B3 \quad (10)$$

For clear understanding, observe the block diagrams for 4x4 as shown below figure 3 and within the block diagram 4x4 totally there are four 2x2 Vedic multiplier modules, and three ripple carry adders which are of four bit size are used. The four bit ripple carry adders are used for addition of two four bits and likewise totally four are use at intermediate stages 3 of multiplier.

The carry generated from the first ripple carry adder is passed on to the next ripple carry adder and there are two zero inputs for second ripple carry adder. The arrangement of the ripple carry adders are shown in below block diagram which can reduces the computational time such that the delay can be decrease.

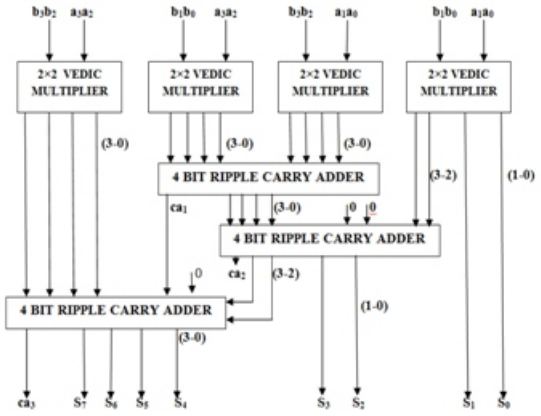


Figure .3. Block Diagram of 4x4 bit Vedic Multiplier

C. 8x8 Vedic Multiplier Block:

In this section, now we will discuss about 4x4 bit Vedic multiplier. For explaining this multiplier let us consider two 8 bit numbers are A and B such that the individual bits can be represented as the A7A6A5A4A3A2A1A0 and B7B6B5B4B3B2B1B0. The procedure for multiplication can be explained in terms of line diagram shown in below figure 4. The final output can be obtained as the 16S15S14S13S12S11S10S9S8S7S6S5S4S3S2S1S0. The partial products are calculated in parallel and hence delay obtained is decreased enormously for the increase in the number of bits. The Least Significant Bit (LSB) S0 is obtained easily by multiplying the LSBs of the multiplier and the multiplicand. Here the multiplication is followed according to the steps shown in the line diagram in figure 4. After performing all the steps the result (Sn) and Carry (Cn) is obtained and in the same way at each step the previous stage carry is forwarded to the next stage and the process goes on.

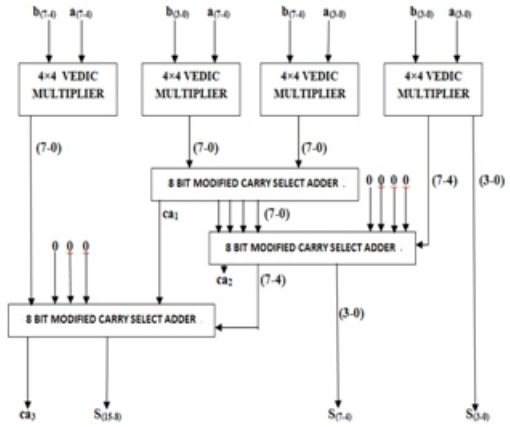


Figure .4. Block Diagram of 8x8 bit Vedic Multiplier

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

For clear understanding, observe the block diagrams for 8x8 as shown below and within the block diagram 8x8 totally there are four 4x4 Vedic multiplier modules, and three modified carry select adders which are of 8 bit size are used. The 8 bit modified carry select adders are used for addition oftwo 8 bits and likewise totally four are use at intermediate stages of multiplier. The carry generated from the first modified carry select adder is passed on to the next modified carry select adder and there are four zero inputs for second modified carry select adders. The arrangement of the modified carry select adders are shown in below block diagram which can reduces the computational time such that the delay can be decrease.

D. 16x16 Vedic Multiplier Block:

In this section, now we will discuss about 4x4 bit Vedic multiplier[10]. For explaining this multiplier let us consider two 8 bit numbers are A and B such that the individual bits can be represented as the A [15:0] and B [15:0]. The procedure for multiplication can be explained in terms of line diagram shown in below figure 5. The final output can be obtained as the C16S[31:0]. The partial products are calculated in parallel and hence delay obtained is decreased enormously for the increase in the number of bits. The Least Significant Bit (LSB) So is obtained easily by multiplying the LSBs of the multiplier and the multiplicand. Here the multiplication is followed according to the steps shown in the line diagram in figure. After performing all the steps the result (Sn) and carry (Cn) is obtained and in the same way at eachstep the previous stage carry is forwarded to the next stage and the process goes on. 16 Bit Modified Carry Select Adder.

The CSLA has two units: 1) the sum and carry generator unit (SCG) and 2) the sum and carry selection unit as shown in the figure 5 . The SCG unit consumes most of the logic resources of CSLA and significantly contributes to the critical path. Different logic designs have been suggested for efficient implementation of the SCG unit. We made a study of the logic designs suggested for the SCG unit of conventional and BEC-based CSLAs of by suitable logic expressions. The main objective of this study is to identify redundant logic operations and data dependence. Accordingly, we remove all redundant logic operations and sequence logic operations based on their data dependence.

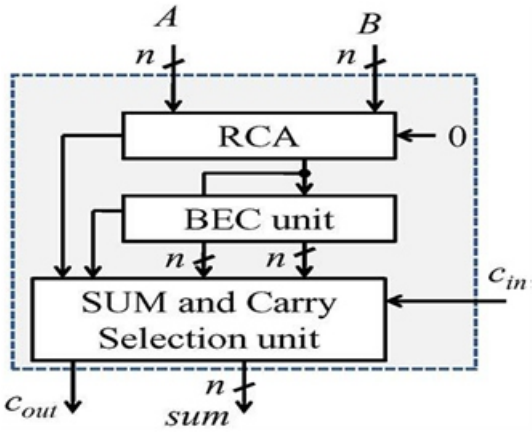
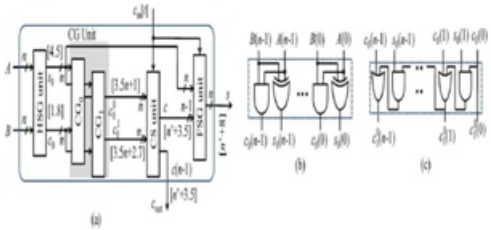


Figure .5.: Structure of the BEC-based CSLA; n is the input operand bit-width

The proposed CSLA is based on the logic formulation given in (4a)–(4g), and its structure is shown in Figure . 6(a). It consists of one HSG unit, one FSG unit, one CG unit, and one CS unit. The CG unit is composed of two CGs (CG0 and CG1) corresponding to input-carry ,,0 and ,,1. The HSG receives two n-bit operands (A and B) and generate half-sum word so and half-carry word co of width n bits each. Both CG0 and CG1 receive so and co from the HSG unit and generate two n-bit full-carry words c11 and c10 corresponding to input-carry ,,0 and ,,1, respectively. The logic diagram of the HSG unit is shown in Figure. 6(b). The logic circuits of CG0 and CG1 are optimized to take advantage of the fixed input-carry bits. The optimized designs of CG0 and CG1 are shown in Figure. 6(c) and (d), respectively.

So(i) = A(i) (+) B(i),Co(i) = A(i) . B(i) 4(a)
C10(i) = C10(i-1) . So(i) + Co(i) for (10(0) = 0) 4(b)
C11(i) = C11(i-1) . So(i) + Co(i) for (11(0) = 1) 4(c)
C(i) = C10(i) if (Cin = 0) C(i) = C11(i) if (Cin = 1) 4(d)

carry words c10 and c11 follow a specific bit pattern. If co1(i)=,,1, then c11(i)=1,irrespective of so(i) and c(i), foro = i = n - 1. This feature is used for logic optimization of the CS unit.



ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

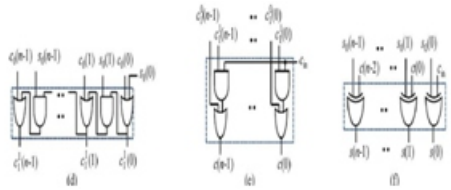


Figure.6:(a) Proposed CS adder design, where n is the input operand bit-width..(b) Gate-level design of the HSG. (c) Gate-level optimized design for input-carry =0. (d) Gate-level optimized design for input-carry =1. (e) Gate-level design of the CS unit. (f) Gate-level design of the final-sum generation (FSG) unit.

Optimized design of the CS unit is shown in Figure. 6(e), which is composed of n AND–OR gates. The final carry word c is obtained from the CS unit. The MSB of c is sent to output as cout and (n-1) LSBs are XORed with (n - 1) MSBs of half-sum (s0) In the FSG [shown in Figure. 6(f)] to obtain (n - 1) MSBs of final-sum(s) .The LSB of s0 is XORed with cin to obtain the LSB of s.The architecture of 16x16 Vedic multiplier using "Urdhva Tiryagbhyam" Sutra is shown in Figure.7. The 16x16 Vedic multiplier architecture is implemented using four 8x8. Vedic multiplier modules, three 16 bit modified carry select adder shown in figure 6 (a) .The first step in the design of 16×16 block will be grouping the 8 bit of each 16 bit input. These pair terms will form vertical and crosswise product terms. Each input bit-pair is handled by a separate 8×8 Vedic the schematic of a 16×16 block designed using 8×8 blocks. The partial products represent the Urdhva vertical and cross product terms.

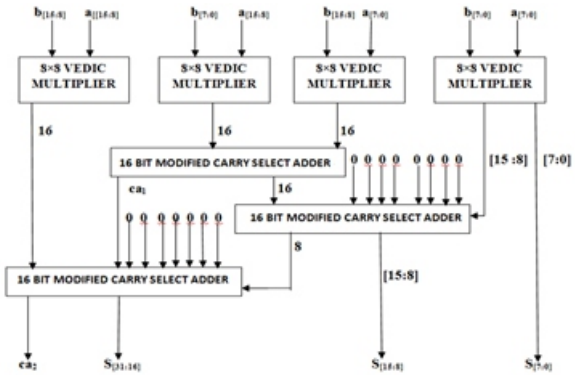


Figure.7. Block Diagram of 16x16 bit Vedic Multiplier.

IV. RESULTS AND COMPARISON:

In 2×2, 4×4, 8×8, 16×16 multiplication operations were designed using Verilog HDL and simulation is performed in Xilinx,RTL compilation is performed in RC and physical design is carried out using Encounter RTL Compiler under Cadence environment.

We have coded the Vedic Multiplier in VHDL using the proposed CSLA design and the existing CSLA designs for bit-widths 8, 16, and 32. All the designs are synthesized in the Cadence Design Suit. The net list file obtained from the DC are processed in the IC Compiler (ICC). After placement and route, the area, Delay, and power reported by the ICC are listed Table 1 for comparison. As shown in Table 1, the proposed SQRT-CSLA involves significantly less area and less delay and consumes less power than the existing designs. We can find from Figure 9 that the proposed Multiplier design offers a saving of 39% ADP and 37% energy than the conventional Vedic Multiplier; The proposed CSLA saves 32% ADP and 33% energy than the BEC-based SQRT-CSLA; on average, for different bit-widths.

The proposed system was analysed for performance and overhead with existing multiplication techniques. The result of performance analysis is visualized in the form of graph to provide a clear insight on the improvements achieved. This analysis was carried out on the following metrics: throughput, Area, ADP, and speed. The resultant values is tabulated.

Table 1: Parameter measurements

Design	Throughput	Area	Delay ns	ADP
Conv CSLA	.552	1438.1	3.45	.692
CSLA With BEC	27.2	1228.2	3.78	21.7
CSLA Using D-Latch	.58	906	4.08	.72
Propsed CSLA	.542	951	3.42	.668

The tabulated values are graphicaly plotted below for comparison of the existing systems and the proposed system. Series
1) Existing Vedic Multiplier
2) Proposed Multiplier.

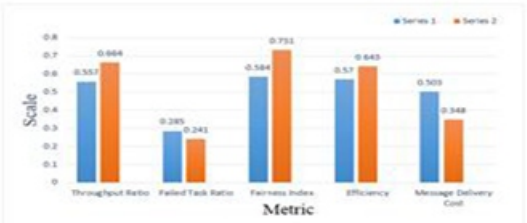


Figure.8.: Performance analysis of proposed system vs existing systems.

The overhead reduction achieved for computing Different Multiplier on analysis result, is visualized using a pie chart.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

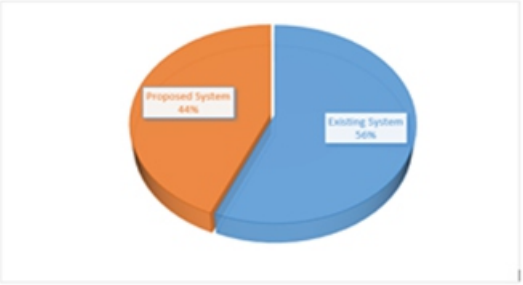


Figure.9: Computational Overhead

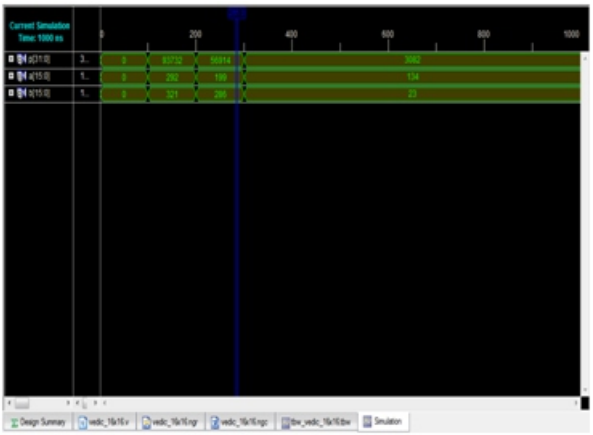


Figure.10. Simulated output for 16x16 bit Vedic Multiplier

The values and the graph proves that this system is more efficient than any existing location calibration techniques based on the ground of fairness, adaptability, scalability and minimal cost for communication and calibration and has minimal computational overhead.

V.CONCLUSION :

This entire work is based on creating a area delay and power efficient Vedic multiplier based on carry select adder. This paper presents a novel way of realizing a high speed multiplier using Urdhva Tiryagbhyam sutra and carry select addition technique. A 16-bit modified multiplier is designed. The 16-bit multiplier is realized using four 8-bit Vedic multipliers and proposed carry select adders. Carry select adders(CSLA) are modified such as all the redundant logic operations present in the conventional CSLA are eliminated and proposed a new logic formulation for CSLA. In the proposed scheme, the carry select (CS) operation is scheduled before the calculation of final-sum, which is different from the conventional approach.

The proposed 16- bit multiplier gives a total delay of 15.050 ns which is less when compared to the total delay of any other renowned multiplier architecture. Results also indicate a 13.65% increase in the speed when compared to normal Vedic multiplier without carry select adder technique.

REFERENCES :

[1] K.K.Parhi, VLSI Digital Signal Processing. New York, NY, USA: Wiley,1998.

[2] Mrs. M. Ramalatha, Prof. D.Sridharan, "VLSI Based High Speed Karatsuba Multiplier for Cryptographic Applications Using Vedic Mathematics", IJSCI, 2007

[3] Ramkumar B, Kittur H.M. (2012) ,"Low Power and Area Efficient Carry Select Adder," IEEE transactions on VLSI systems, Vol 20,No.2.

[4] Sreeni vasulu P, Srinivasa Rao K., Vinay Babu A., (2012) " Energy And Area Efficient Carry Select Adder on a Reconfigurable Hardware" International Journal of Engineering Research and Application, Vol 2, Issue 2.

[5] Jai Skand, Priya Keerthu, Deepthi Shakti, (2012) " An Efficient Design of Vedic Multiplier using New Encoding Scheme" International Journal on Computer Applications,Vol-53,No.11.

[6] Anvesh kumar, Ashish Raman and Sarin,Arun Khosla R.K, (2010)"Small Area Reconfigurable FFT Design by Vedic Mathematics " in Proc IEEE ICAAE10,Singapoure, vol 5,pp. 836-838,.

[7] Tam Anh Chu, (2002) "Booth Multiplier with Low Power High Performance Input Circuitary", US Patent, 6.393.454 B1.

[8] Gankhuyag G, Chan Mo Kim, Yong Beom Cho, (2008), "Multiplier Design based on ancient Indian Vedic Mathematics", in SOC Design Conference, Volume 2.

[9] Y. Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," Electron. Lett., vol. 37, no. 10, pp. 614–615, May 2001.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[10] Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry select adder for low power application," in Proc. IEEE Int. Symp. Circuits Syst., 2005, vol. 4, pp. 4082–4085.

[11]Swami Bharati Krishna Tirtha, Vedic Mathematics. Delhi: Motilal Banarsidass Publishers, 1965.

[12] D. Goldberg, "Computer Arithmetic", in Computer Architecture: A Quantitative Approach, J.L.Hennessy and D.A. Patterson ed., pp. A1-A66, San Mateo, CA: Morgan Kaufmann, 1990.

[13]A.D. Booth, "A Signed Binary MultiplicationTechniq ue",Qrt.J.Mech.App. Math.,, vol. 4, pp.236–240, 1951.

[14]A.P. Nicholas, K.R Williams, J. Pickles, "Application of Urdhava Sutra", Spiritual Study Group, Roorkee (India),1984.

[15]Ming-Chen Wen, Sying-Jyan Wang, and Yen-Nan-Lin, .Low PowerParallel Multiplier with Column Bypassing., Electronics letters, 10,12 May 2005 Volume 41, Issue Page(s): 581 -583

[16] Harpreet Singh Dhillon and Abhijit Mitra, "A Reduced- Bit Multiplication Algorithm for Digital Arithmetics", International Journal of Computational and Mathematical Sciences 2;2 © www.waset.org Spring 2008.

[17] Stephen Brown and Zvonko Vranesic, 2005. Fundamentals of Digital Logic with VHDL Design, 2nd Edition. McGraw-Hill Publishing Companies.

[18] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput., vol. EC-11, no. 3, pp. 340–344, Jun. 1962.