International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

Design of A Vedic Multiplier Using Area Efficient Bec Adder

Pulakandla Sushma & M.VS Prasad
sushmareddy0558@gmail.com[1] & prasadmadduri54@gmail.com[2]

[1]pg Scholar, Dept Of Ece, Siddhartha Institute Of Technology And Sciences Narapally, Ghatkesar, Ranga Reddy, Telangana.

[2]assistant Professor, Dept Of Ece , Siddhartha Institute Of Technology And Sciences Narapally, Ghatkesar, Ranga Reddy, Telangana.

Abstract: *This paper presents a new design methodology for less delay and area efficient Vedic Multiplier based up on ancient Vedic Mathematic techniques. This paper presents a technique for N×N multiplication is implemented and gives very less delay and area efficient for calculating multiplication results for 16×16 Vedic multiplier. In this paper the efficiency of Urdhva Tiryagbhyam (vertical and crosswise) Vedic method for multiplication which is different from the process of normal multiplication is presented. Urdhva -Tiryagbhyam is the most efficient algorithm that gives minimum delay for multiplication for all types of numbers irrespective of their size. Vedic multiplier is coded in Verilog HDL and stimulated and synthesized by using XILINX ISE software 14.7. Further the design of array multiplier is compared with the proposed multiplier in terms of l delay and area.*

Keywords— Vedic mathematics, Vedic multiplier, UrdhvaTiryagbhyam, Array multiplier, Ripple Carry Adder (RCA), Binary to Excess Code Converter (BEC), Half Adder (HA), Full Adder(FA), Carry Select Adder (CSLA).

I. INTRODUCTION

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. 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.

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.

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. 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. 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.

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.

II. VEDIC MULTIPLIER ARCHITECTURE

The architectures for 2×2, 4×4,. . .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.

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.

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

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 Figure1 .

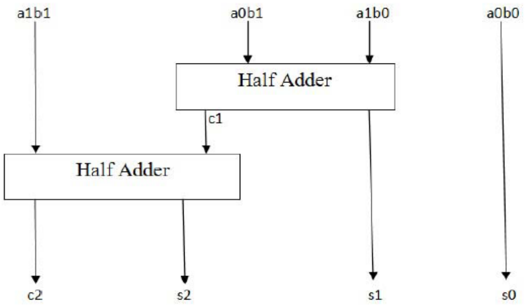


Figure1.. Block Diagram of 2×2 Vedic Multiplier

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.

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

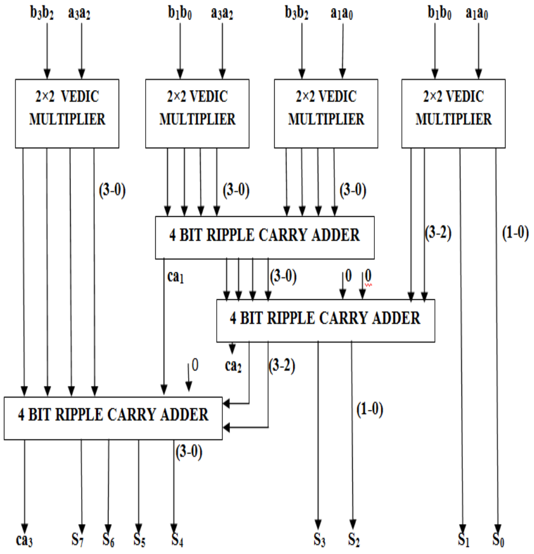


Figure 2. . Block Diagram of 4x4 bit Vedic Multiplier

The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA with Cin = 1 in the regular CSLA to achieve lower area and power consumption. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure. The SQRT CSLA has been chosen for comparison with the proposed design as it has a more balanced delay, and requires lower power and area. The delay and area evaluation methodology of the regular and modified SQRT CSLA are presented.

The AND, OR, and Inverter (AOI) implementation of an XOR gate is shown in Fig.3. The gates between the dotted lines are performing the operations in parallel and the numeric representation of each gate indicates the delay contributed by that gate. The delay and area evaluation methodology considers all gates to be made up of AND, OR, and Inverter, each having delay equal to 1 unit and area equal to 1 unit. We then add up the number of gates in the longest path of a logic block that contributes to the maximum delay. The area evaluation is done by counting the total number of AOI gates required for each logic block..

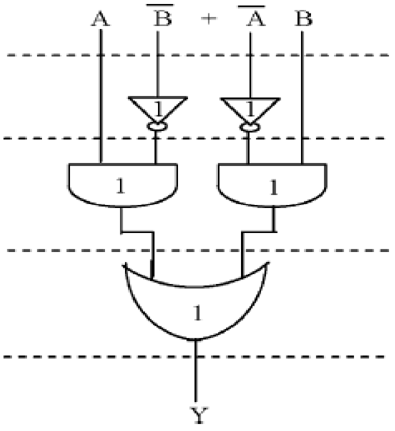


Fig.3. (AOI) implementation of an XOR gate

Binary to Excess Converters

As stated above the main idea of this work is to use BEC instead of the RCA with Cin = 1 in order to reduce the area and power consumption of the regular CSLA. To replace the n-bit RCA, an n + 1-bit BECis required. A structure and the function table of a 4-b BEC are shown in Fig.4 and Table 4.II, respectively.

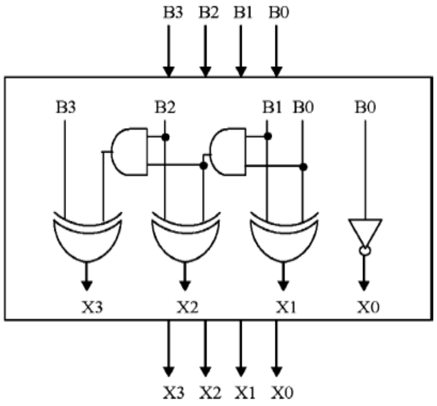


Fig.4. 4-b BEC.

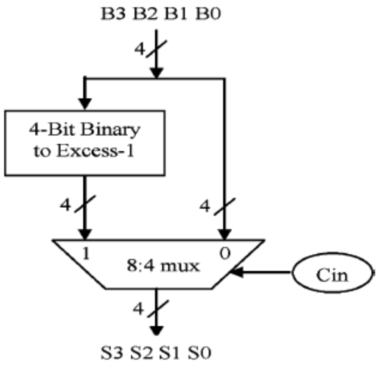


Fig.5. 4-b BEC with 8:4 mux.

TABLE.I

FUNCTION TABLE OF THE 4-b BEC

B[3:0]	X[3:0]
0000	0001
0001	0010
⋮	⋮
1110	1111
1111	0000

Fig.5 illustrates how the basic function of the CSLA is obtained by using the 4-bit BEC together with the mux. One input of the 8:4 mux gets as it input (B3, B2, B1, and B0) and another input of the mux is the BEC output. This produces the two possible partial results in parallel and the mux is used to select either the BEC output or the direct inputs according to the control signal Cin. The importance of the BEC logic stems from the large silicon area reduction when the CSLA with large number of bits are designed. The Boolean expressions of the 4-bit BEC is listed as (note the functional symbols NOT, & AND;^XOR)

X0 = B0
X1 = B0^B1
X2 = B2^(B0 & B1)
X3 = B3^(B0 & B1 & B2)

The structure of the 16-b regular SQRT CSLA is shown in Fig.6. It has five groups of different size RCA., in which the numerals within [] specifythe delay values, e.g., sum2 requires 10 gate delays. The steps leading to the evaluation are as follows.

1) The group2 has two sets of 2-b RCA. Based on the consideration of delay values of Table I, the arrival time of selection input c1[time(t) = 7] of 6:3 mux is earlier than s3[t =8] and later than s2[t = 6]. Thus, sum3[t = 11] is summation ofs3 and mux[t = 3] and sum2[t = 10] is summation of c1 and mux.

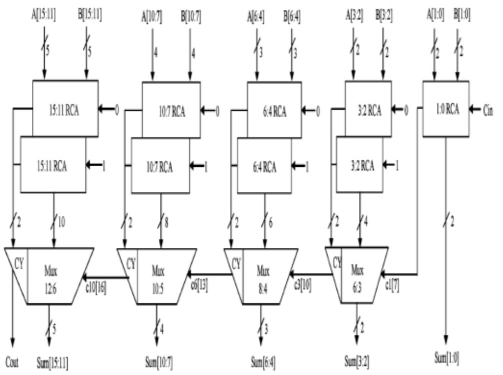


Fig.6 Regular 16-b SQRT CSLA.

Implementation Of Vedic 16x16 Multiplier Using BEC

The design of 4x4 Vedic multiplier is used as a basic building block diagram for design of 8x8 Vedic multiplier. Further design of 16x16 is implemented by using 8x8 Vedic multiplier as basic building block. The aim of using BEC is to reduce the usage of gates compared to normal Vedic multiplier which in turn reduces the power consumption. The structure of proposed Vedic multiplier is shown in figure 7. It has 4 groups of same size i.e each group consists of 8*8 Vedic multiplier whose inputs are partitioned according to Urdhva-Tiryagbhyam sutra. Outputs from Vedic multiplier are given as inputs to BEC adders of different sizes.

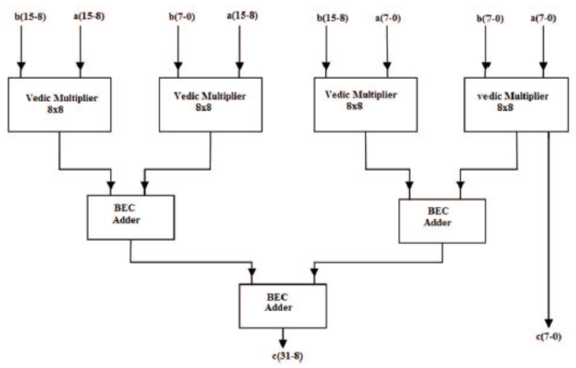
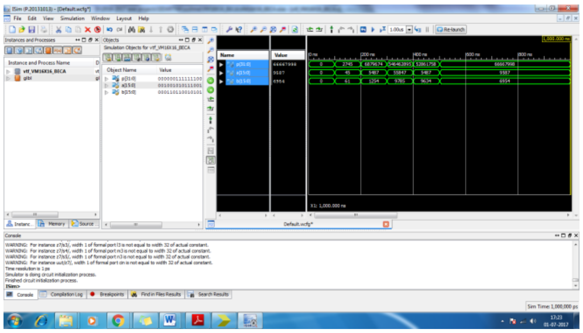


Fig. 7.Block diagram of 16x16 Vedic multiplier using BEC

III. RESULTS

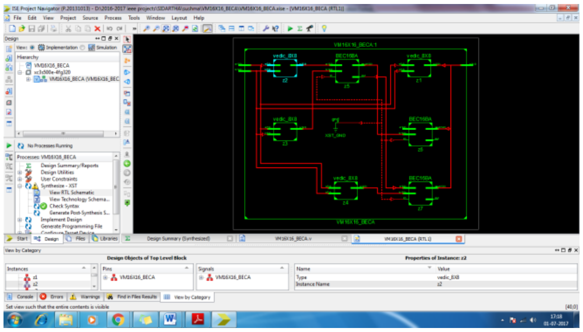
Proposed 16x16 Vedic Multiplier using BEC adder

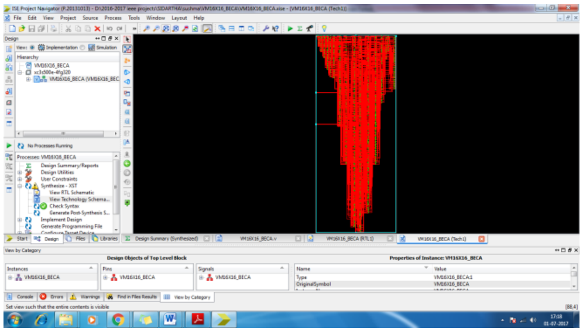
SIMULATION



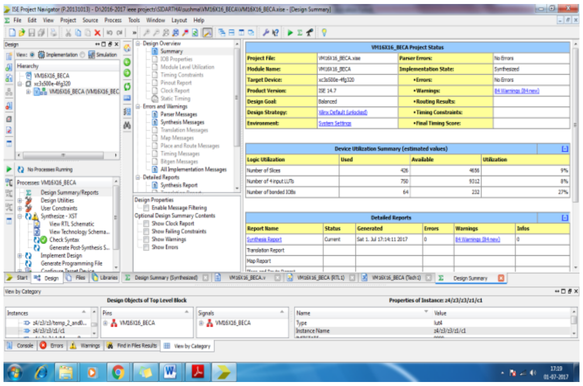
RTL SCHEMATIC

TECHNOLOGY





DESIGN SUMMARY



VM16X16_BECA Project Status			
Project File:	VM16X16_BECA.xise	Parser Errors:	No Errors
Module Name:	VM16X16_BECA	Implementation State:	Synthesized
Target Device:	xc3s500e-4fg320	•Errors:	No Errors
Product Version:	ISE 14.7	•Warnings:	84 Warnings (84 new)
Design Goal:	Balanced	•Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:	
Environment:	System Settings	•Final Timing Score:	

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slices	426	4656	9%
Number of 4 input LUTs	750	9312	8%
Number of bonded IOBs	64	232	27%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Sat 1. Jul 17:14:11 2017	0	84 Warnings (84 new)	0
Translation Report					
Map Report					

Properties of Instance: z4/z3/z3/z1/c1

Name	Value
Type	lut4
Instance Name	z4/z3/z3/z1/c1

IV. CONCLUSION

The main focus of this paper is to introduce a method for designing of Vedic multiplier using Binary to excess converter (BEC) adder circuit. The execution time and area of the proposed method for convolution using vedic multiplication algorithm is compared with that of convolution with the simple multiplication is less. The proposed Vedic multiplier gives less power consumption when compared to other multiplier techniques because the number of additions gets reduced by applying Urdhva-Tiryakbhyam which is a short approach form of multiplication. This multiplier has very less delay because of addition new BEC adder. By comparing the values of both Array and Vedic multiplier it is clear that the delay for Vedic multiplier is much less when compared with Array multiplier. As we increase number of bits delay can be reduced by using Vedic multiplier than Array multiplier.

REFERENCES

[1] G. Ganesh Kumar, V. Charishma , "Design of High Speed Vedic Multiplier using Vedic Mathematics Techniques", International Journal of Scientific and Research Publications, Volume 2, Issue 3, March 2012.

[2] S.P.Pohokar, R.S.Sisal, K.M.Gaikwad, M.M.Patil, Rushikesh Borse," Design and Implementation of 16 x 16 Multiplier Using Vedic Mathematics", Department of Electronics and Telecommunication Engineering, Sinhgad Academy of Engineering, Kondhwa Pune, India.

[3] Poornima M, Shivaraj Kumar Patil, Shivukumar , Shridhar K P , Sanjay H," Implementation of Multiplier using Vedic Algorithm", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-6, May 2013.

[4] Sudeep. M. C, Sharath Bimba. M, Mahendra Vucha, “Design and FPGA Implementation of High Speed Vedic Multiplier”, International Journal of Computer Applications, Volume 90-No. 16, March 2014.

[5] G.Vaithiyanathan, K.Venkatesan, S.Sivaramakrishnan, S.Siva, and S. Jayakumar, “Simulation and Implementation of Vedic Multiplier Using VHDL Code” International Journal of Scientific & Engineering Research Volume 4, Issue 1, January-2013.

[6] Jagadguru Swami Sri Bharti Krishna Tirthaji Maharaja,”Vedic Mathematics or Sixteen Simple Mathematicle Formulae from the Veda, Delhi(1965)”, Motilal Banarsidass, Varanasi,India.

[7] Chilton Fernandes, Samarth Borkar, “Application of Vedic Mathematics in Computer Architecture”, International Journal of Research in Engineering and Science (IJRES), Volume 1, Issue 5, September 2013.

[8] C. Sheshavali, K. Niranjan Kumar, “Design and Implementation of Vedic Multiplier”, International Journal of Engineering Research and Development, Volume-8, PP.23-28, Issue 6,September 2013.

[9] Premananda B.S. , Samarth S.Pai, Shashank S.Bhat.,”Design and Implementation of 8-Bit Vedic Multiplier”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Volume 2, Issue 12, December 2013.

[10] A Debasish Subudhi, Kanhu Charan Gauda, Abinash Kumar Pala, Jagamohan Das, " Design and Implementation of High Speed 4x4 Multiplier", International Journal of Advanced Research in computer Science and Software Engineering, Volume 4, Issue 11, November 2014.

[11] B. Tapasvi, K. Bala Sindhuri, I. Chaitanya Varma, N. Udaya Kumar, "Implementation of 64 Bit KoggeStone Carry Select Adder With BEC For Efficient Area", IJRECE, Volume 3, Issue 1, Jan - March 2013

[12] B. Ramkumar and Harish M Kittur, “Low–Power and Area-Efficient Carry Select Adder”,IEEE transactions on very large scale integration(VLSI) systems, VOL.20, NO.2, february 2012. Reducing the delay is very much advantage because it can increases speed and this Vedic multiplier is very useful for low power and high speed applications.