International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.3, 2015

RESEARCH ARTICLE

ISSN: 2321-7758

AN EFFICIENT AND FAST VLSI ARCHITECTURE FOR CARRY SELECT ADDER

RAKSHA CHOUKSEY[1], NEHA VERMA[2]
[1,2]Department Of Electronics & Communication, Trinity Institute of Technology & Research, Bhopal[INDIA]

Article Received:06/05/2015 | Article Revised on:12/05/2015 | Article Accepted on:16/05/2015

RAKSHA CHOUKSEY

ABSTRACT
Carry Select Adder (CSA) is one of the fastest adder used in many processors to perform fast arithmetic operations. Speed of processor greatly depends on its multiplier as well as adder performance. Due to which high speed adder architecture become important. Several adder architecture designs have been developed to increase the efficiency of the adder. In this paper, we present an innovative CSA architecture. Verification of our proposed design is done through design and implementation of 16, 32 and 64 bit adder circuits. Comparison is done with existing structure of adder and proves the efficiency of our proposed design. In this paper, we introduce an architecture that performs high speed addition in carry select adder using binary to excess-1 converter (BEC). These designs are implemented on Xilinx device family.

Keywords: - Carry Select Adder (CSA), Ripple Carry Adder (RCA), Area – Delay Product, Binary to Excess-1 converter (BEC).

I. INTRODUCTION

Adders are commonly found in the critical path of many building blocks of microprocessors, microcontroller, digital image processing and digital signal processing chips. Adders are essential not only for addition, but also for subtraction, multiplication, and division. Addition is one of the fundamental arithmetic operations. A fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. Speed is important constraints for designing of any digital circuits. In digital adders, speed of addition is limited by the time required for a carry to propagate through the adder. In conventional adder the sum for each bit position is generated sequentially only after the previous bit position has been summed and carry propagated into the next position [1]. Many approaches are there to improve the performance of the adder. Carry select adder (CSA) is one among them. CSA is used to solve the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the final sum. However, CSA consumes more area because it uses multiple pairs of ripple carry adders(RCA) to generate the partial sum and carry by considering carry input '0' and '1' respectively, then the final sum and carry are chosen by the use of multiplexers [2]. The fundamental approach is in this paper is used modified binary to excess-1 converter to reduce the area and delay of adder and increase the performance [1].

II. BINARY TO EXCESS-1 CONVERTER (BEC)

Binary to excess-1 converter is used to reduce the area and power consumption in Carry Select Adder [2]. Figure 1 shows the basic structure of 4-bit BEC. The Boolean expressions of the 4-bit BEC is as

X0 = ~ B0 (1)

X1 = B0^B1 (2)

X2 = B2^ (B0 & B1) (3)

X3 = B3^ (B0 & B1 & B2) (4)

Table 1: Function Table of 4-bit Excess-1 converter (BEC)

Binary[2:0]	Excess- 1[2:0]
0000	0001
0001	0010
0010	0011
|	|
|	|
1111	0000

The main idea of Binary to excess-1 converter is used instead of the RCA (ripple carry adder) with Cin =1 in order to reduce the area and power consumption of the 16-B Carry Select Adder.

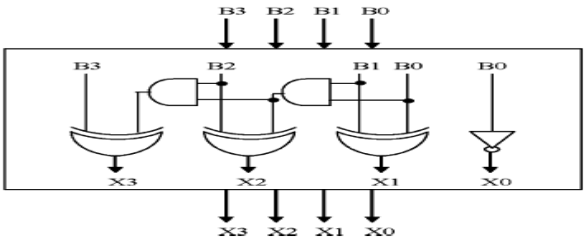

Figure 1: 4-bit Binary to Excess-1 converter (BEC).

III. BASIC STRUCTURE OF REGULAR 16-B CSA USING RCA

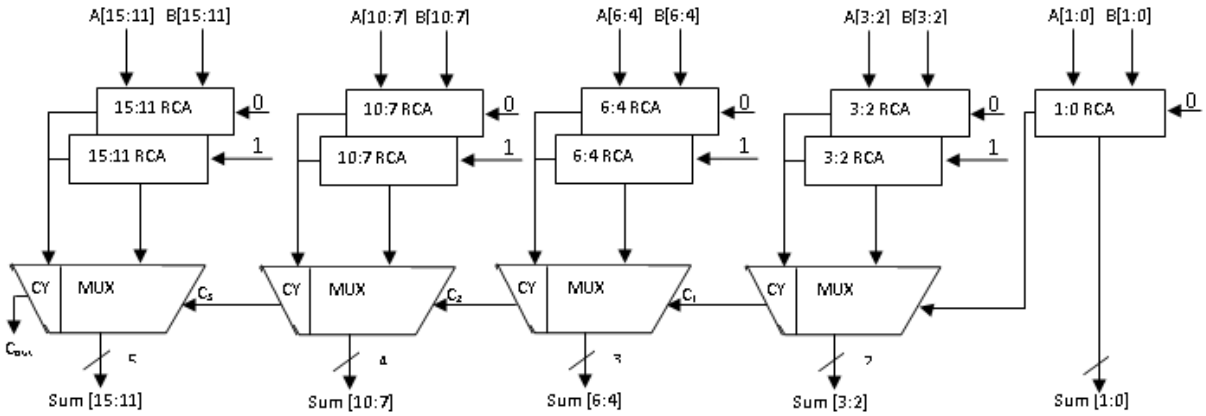

Figure 2: 16-Bit Carry Select Adder using Ripple Carry Adder

IV. MODIFIED STRUCTURE OF 16-B CARRY SELECT ADDER USING BEC

The structure of the modified 16-Bit CSA using BEC instead of the RCA with $C_i = 1$ is shown in Figure 3.

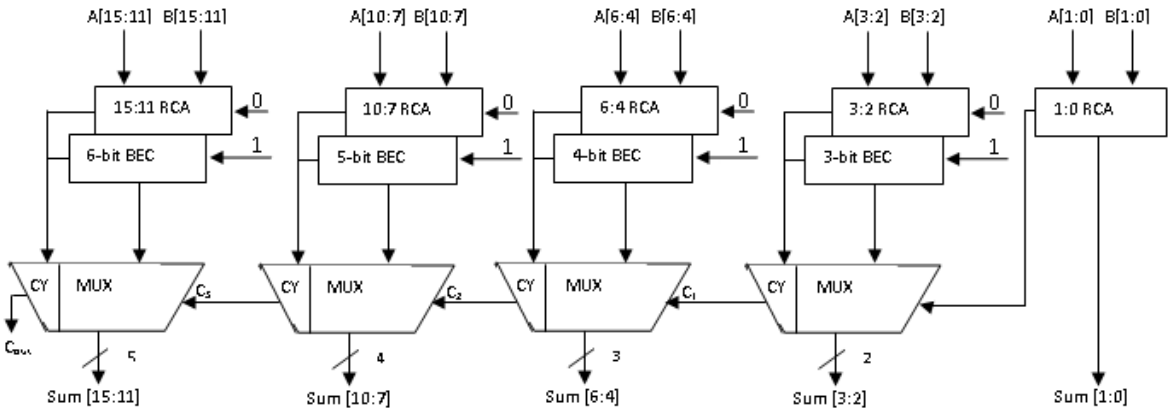

Figure 3: Modified 16-B Carry Select Adder using BEC

Figure 2 shows schematic of the 16-Bit CSA using RCA. The structure consists of five groups with different bit size RCA. RCA is simplest adder but their performance is limited by a carry that propagate from the least-significant bit to the most-significant bit. Group 1 contains only one 2-bit RCA which adds the input bits and the carry input and results to sum [1:0] and the carry out. The carry out of the Group 1 which acts as the selection input to mux of group 2. If the carry-in is 0, the sum and carry-out of the upper RCA is selected, and if the carry-in is 1, the sum and carry-out of the lower RCA is selected [2]. Similarly the remaining groups will be selected depending on the Cout from the previous groups.

The structure is again divided into five groups with different bit size RCA and BEC. The Binary to excess one Converter (BEC) replaces the ripple carry adder with Cin=1, in order to reduce the power consumption and area of the regular CSA. In this structure also group 1 contains one 2- bit RCA and output of this RCA goes to the next group. From the next group it contains the combination of one 2-bit RCA and one 3-bit BEC. One input to the mux goes from the RCA with Cin=0 and other input from the BEC. Then the final sum and carry are selected by the multiplexers (mux) [1].

The structure is again divided into five groups with different bit size RCA and BEC. The Binary to excess one Converter (BEC) replaces the ripple carry adder with Cin=1, in order to reduce the power consumption and area of the regular CSA. In this structure also group 1 contains one 2- bit RCA and output of this RCA goes to the next group. From the next group it contains the combination of one 2-bit RCA and one 3-bit BEC. One input to the mux goes from the RCA with Cin=0 and other input from the BEC. Then the final sum and carry are selected by the multiplexers (mux) [1].

V. PROPOSED 16-B CARRY SELECT ADDER

In our proposed work Carry Select Adder using modified binary to excess-1 converter (BEC) is designed. The proposed structure is shown in figure 4. This architecture is used to improve the overall system performance in terms of delay and area, when compare to other existing design. Half adder is used to generate the partial sum for Cin= 0 and modified BEC is used for calculating the partial sum for Cin=1.

The structure is again divided into five groups with different bit size RCA and modified BEC. After 1st group of our proposed deign, we replaced all the RCA block form our modified BEC block. So from block diagram itself it is clear that it is less area consuming architecture.

Group 1 contains 2- bit RCA which contains one ripple carry adder which adds the input bits and input carry and results to sum[1:0] and carry out. The carry out of the group 1 will acts as the selection input to the next block. In group 2 modified BEC will give output for carry input Cin=0 and for Cin=1 and the correct output is selected with the help of multiplexer. The internal architecture of group 2 is shown in figure 5. This modified BEC circuit will provide output for either the carry input Cin=0 or cin=1. For Cin=0 the output is selected with the help of half adder and for doing this operation we have define one intermediate signal Su1 and the sum output is taken at Su1 and carry output at C2. For Cin=1 the whole circuitry will provide the sum and carry output. The other groups operation can be designed in the same way. In same way we can also design for 32 bit and 64 bit adder.

VI. SIMULATION RESULTS

All the designing and experiment regarding algorithm that we have mentioned in this paper is being developed on Xilinx 6.1i Spartan 3A, Vertex 2P and Vertex E updated version. Xilinx 6.1i has a couple of the striking features such as low memory requirement, fast debugging, and low cost. The latest release of ISETM (Integrated Software Environment) design tool provides the low memory requirement approximate 27 percentage low. ISE 6.1i provides advanced tools like smart Compile technology with better usage of their computing hardware provides faster timing closure and higher quality of results for a better time to designing solution.

Figure 6, 7, 8 represents the simulation result of Regular 16-Bit CSA with RCA, Modified 16-Bit CSA with BEC and Proposed 16-Bit CSA with modified BEC respectively.

In order to perform comparison, various adder – regular 16-bit CSA using RCA, 16-bit CSA using RCA and BEC, proposed carry select adder implemented on a Xilinx 6.1i.

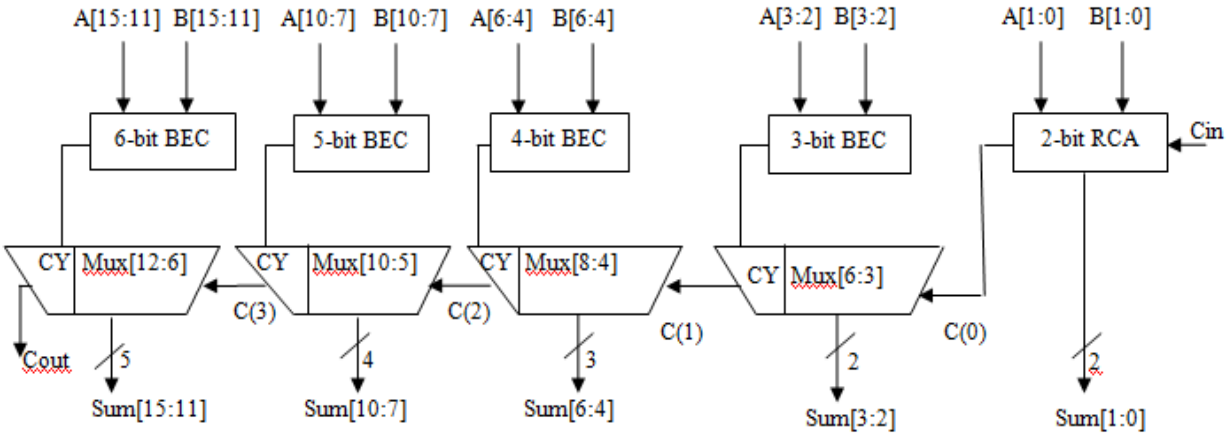

Figure 4: Proposed 16-B Carry Select Adder using Modified BEC

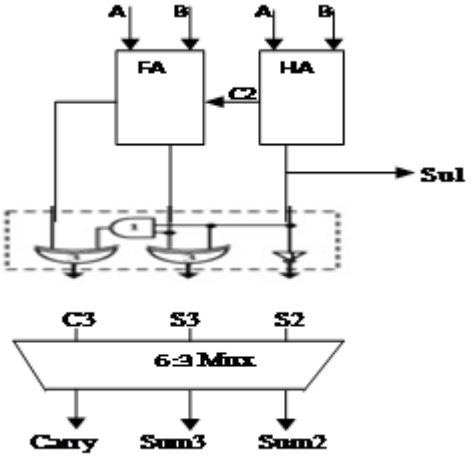

Figure 5: Group 2 of Modified Binary to Excess-1 Converter

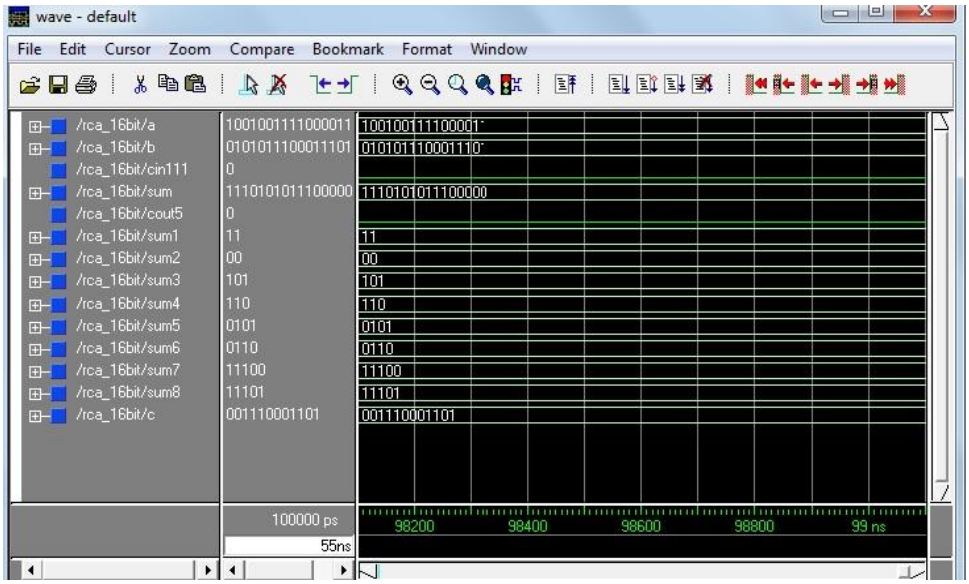

Figure 6: Simulation result of Regular 16-Bit CSA

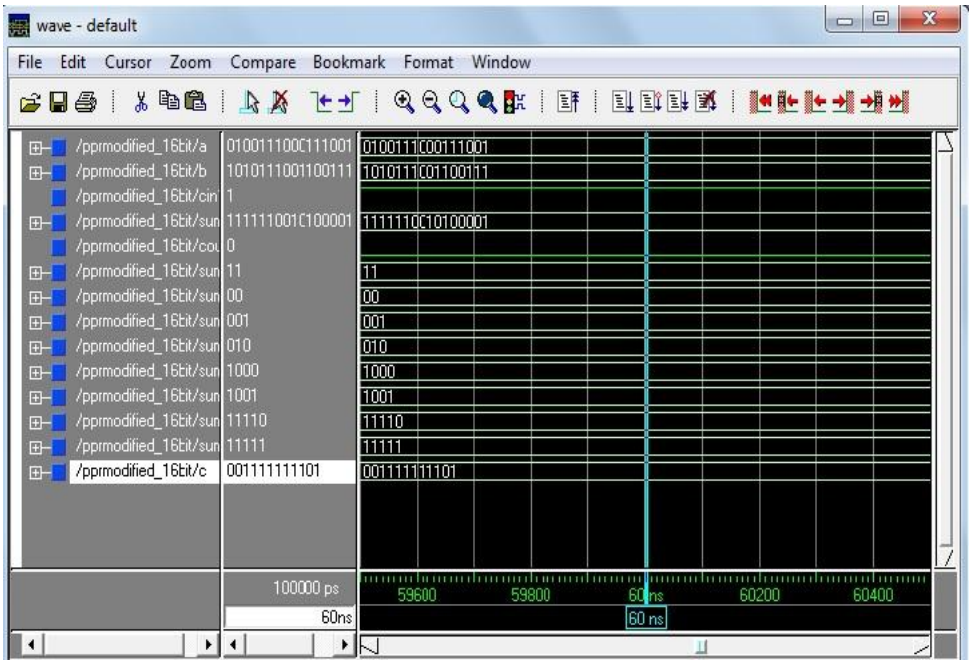

Figure 7: Simulation result of Modified 16-Bit CSA using BEC

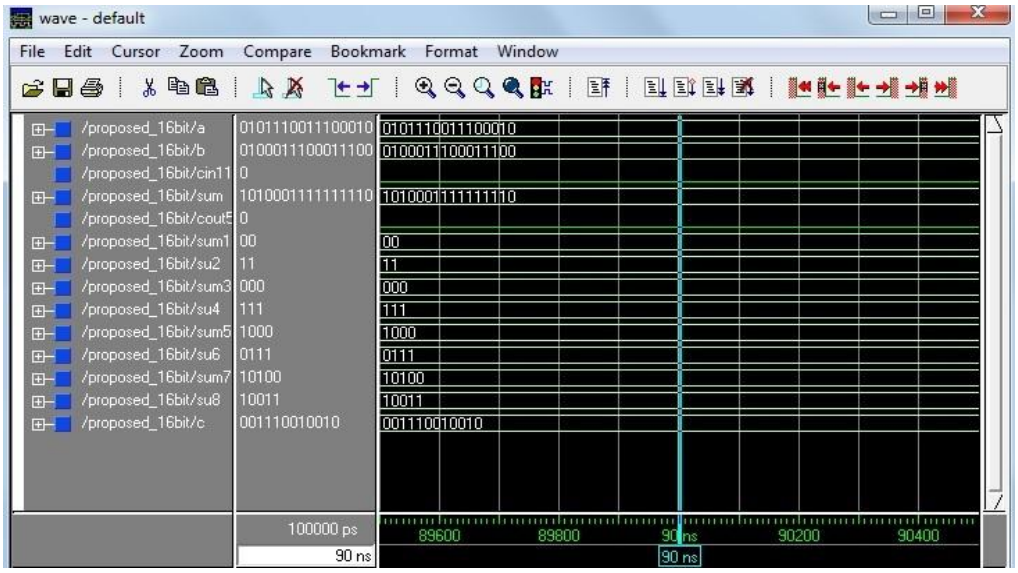

Figure 8: Simulation result of Proposed 16-Bit CSA using modified BEC

Table 2: Comparison Table of Different Adder For Different Device Family

Size	Adder	No. of Slices	Delay (ns) (Spartan 3)	Delay (ns) (Vertex E)	Delay (ns) (vertex 2p)
16-bit CSA	Regular CSA (dual RCA)	30 out of 768	21.298	24.269	13.110
	Regular CSA (RCA with BEC)	25 out of 768	18.786	21.354	11.600

	Proposed CSA (Modified BEC)	25 out of 768	17.826	20.108	10.965
32-bit CSA	Regular CSA (dual RCA)	59 out of 768	35.932	43.046	22.040
	Regular CSA (RCA with BEC)	49 out of 768	30.428	36.685	18.770
	Proposed CSA (Modified BEC)	51 out of 768	28.508	34.193	17.500
64-bit CSA	Regular CSA (dual RCA)	118 out of 768	64.660	79.835	39.630
	Regular CSA (RCA with BEC)	99 out of 768	53.712	67.347	33.110
	Proposed CSA (Modified BEC)	101 out of 768	49.872	62.363	30.570

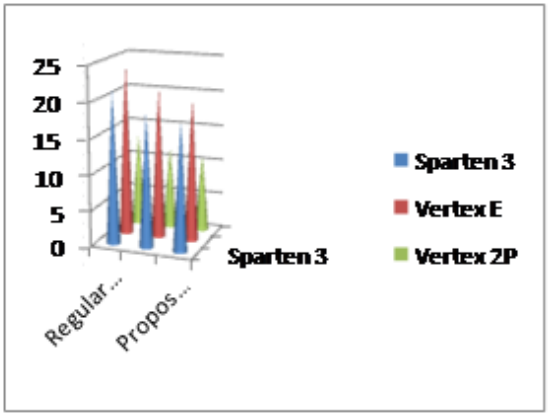

Figure 9: comparison of Adder for Delay (Word Size =16)

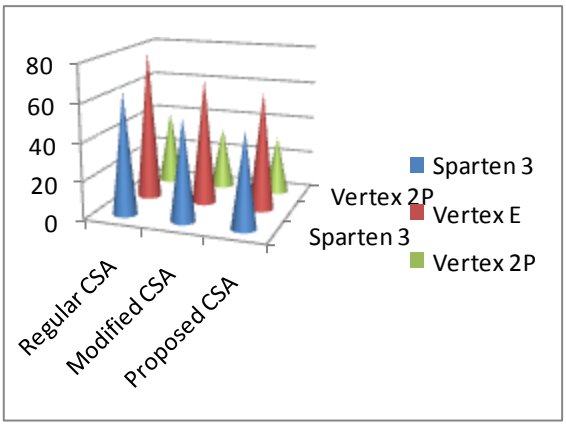

Figure 11: comparison of Adder for Delay (Word Size =64)

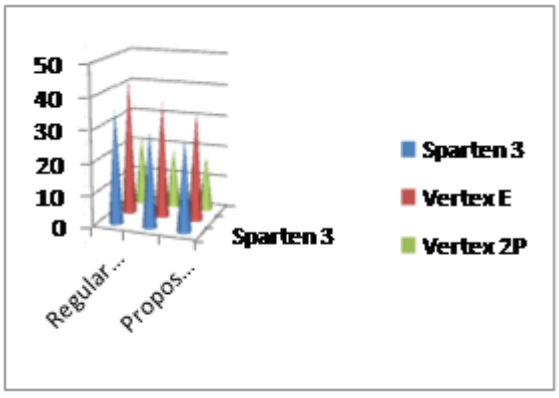

Figure 10: comparison of Adder for Delay (Word Size =32)

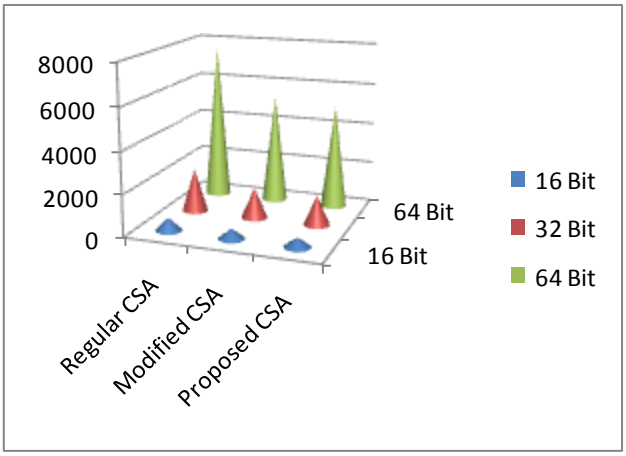

Figure 12: comparison of Adder for Area-Delay Product For Spartan 3 Device Family

Table 2 shows the comparison of different adders in different device family and found that our proposed result is best among all the families. Figure 9,10,11 shows the comparison of adders in terms of delays for word size 16,32 and 64 respectively. Figure 12 shows the comparison of area-delay product for all the Carry Select Adder in spartan 3 device family for 16-bit, 32-bit and 64-bit and the product is decreasing for our proposed design as shown in figure. All the designing and experiment regarding algorithms have been captured by VHDL and the functionality is verified by RTL and gate level simulation.

V. CONCLUSION

A simple technique is proposed in this paper to reduce the delay of Carry Select Adder. The area-delay product of our proposed design show a decrease for 16-bit, 32-bit and 64-bit sizes which shows the success of the proposed design. The regular CSA has a disadvantage of large chip area and larger delay. The modified CSA with BEC reduces the area and delay, when compare to conventional CSA. The technique which is discussed in this paper reduces area and delay when compared to both previous CSA. It would be interesting to test the design for 128-bit.

REFERENCE

[1] B. Ramkumar and Harish M Kittur, “Low-Power and Area-Efficient Carry Select Adder”, IEEE Transsactions on Very Large Scale Integration (VLSI) Systems, VOL. 20, No. 2 Feb 2012.

[2] Ms. S.Manju, Mr. V. Sornagopal “An Efficient SQRT Architecture of Carry Select Adder Design by Common Boolean Logic”, 978-1-4673-5301-4/13/$31.00 ©2013 IEEE.

[3] Sajesh Kumar U., Mohamed Salih K. K. Sajith K.,“Design and Implementation of Carry Select Adder without Using Multiplexers”, 2012 1st International Conference on Emerging Technology Trends in Electronics, Communication and Networking 978-1-4673-1627-9/12/$31.00 ©2012 IEEE.

[4] Samiappa Sakthikumaran, S. Salivahanan, V. S. Kanchana Bhaaskaran, V. Kavinilavu, B. Brindha and C. Vinoth, “A Very Fast and Low Power Carry Select Adder Circuit”, 978-1-4244 -8679-3 /11/$26.00 ©2011 IEEE.

[5] B. Ramkumar, H.M. Kittur, and P. M. Kannan, "ASIC implementation of modified faster carry save adder," Eur. J Sci. Res., vol. 42, no. 1, pp. S3-S8, 2010.

[6] Padma Devi, Ashima Girdher, Balwinder Singh, “Improved Carry Select Adder with Reduced Area and Low Power Consumption”, International Journal of Computer Applications (0975 – 8887) Volume 3 – No.4, June 2010.

[7] T. Y. Ceiang and M. J. Hsiao, “Carry-select adder using single ripple carry adder,” *Electron. Lett.*, vol. 34, no. 22, pp. 2101–2103, Oct. 1998.

[8] Z. Abid, H. El-Razouk and D.A. El-Dib, “Low power multipliers based on new hybrid full adders”, Microelectronics Journal, Volume 39, Issue 12, Pages 1509-1515, 2008.

[9] I-Chyn Wey, Cheng-Chen Ho, Yi-Sheng Lin, and Chien Chang Peng 'An Area-Efficient Can), Select Adder Design by Sharing the Common Boolean Logic Term' Proceeding on the international Multiconference of eng. and computer scientist 2012 , IMECS 2012.

[10] J. M. Rabaey, Digtal Integrated Circuits-A Design Perspective. Upper Saddle River, NJ: Prentice-Hall, 2001.

[11] Hasan Krad and Aws Yousif Al-Taie, “Performance Analysis of a 32-Bit Multiplier with a Carry-Look-Ahead Adder and a 32-bit Multiplier with a Ripple Adder using VHDL”, Journal of Computer Science 4 (4): 305-308, 2008.

[12] Wang, Y. Pai, C.Song, X., “The design of hybrid carry look ahead/ carry-select adders”, Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on Volume 49, pp.16-24, 2002.

[13] Yotmgjoon Kim and Lee-Sup Kim, “A Low Power Carry Select Adder with Reduced Area”, 0-7803-6685-9/01/$10.0002001 TEEE.

[14] W. Jeong and K. Roy, “Robust high-performance low power adder”, Proc. of the Asia and South Pacific Design Automation Conference, pp. 503-506, 2003.

[15] V.G. Oklobdzija, “High-Speed VLSI Arithmetic Units: Addersand Multipliers”, in “Design of High-Performance Microprocessor Circuits”, Book edited by A. Chandrakasan, IEEE press, 2000.