

## **An Efficient 64-Bit Carry Select Adder With Less Delay And Reduced Area Application**

**K Allipeera, M.Tech Student & S Ahmed Basha, Assitant Professor**

Department of Electronics & Communication Engineering  
St.Johns College Of Engineering & Technological  
Yemmiganur, Andhra Pradesh, India

### **Abstract**

Design of area, high speed and power-efficient data path logic systems forms the largest areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to transmit a carry through the adder. Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. From the structure of the CSLA, it is clear that there is scope for reducing the area and delay in the CSLA. This work uses a simple and an efficient gate-level modification (in regular structure) which drastically reduces the area and delay of the CSLA. Based on this modification 16, 32, 64 and 128-bit square-root Carry Select Adder (SQRT CSLA) architectures have been developed and compared with the regular SQRT CSLA architecture. The proposed design has reduced area and delay to a great extent when compared with the regular SQRT CSLA. This work estimates the performance of the proposed designs with the regular designs in terms of delay, area and synthesis are implemented in Xilinx FPGA. The results analysis shows that the proposed CSLA structure is better than the regular SQRT CSLA.

**Index Terms**—Application-specific integrated circuit (ASIC), area-efficient, CSLA, low delay.

### **1. INTRODUCTION**

Reduced area and high speed data path logic systems are the main areas of research in VLSI system design. High-speed addition and multiplication has always been a fundamental requirement of high-performance processors and systems. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position. There are many types of adder designs available (Ripple Carry Adder, Carry Look Ahead Adder, Carry Save Adder, Carry Skip Adder) which have its own advantages and disadvantages. The major speed limitation in any adder is in the production of carries and many authors considered the addition problem. To solve

The carry propagation delay CSLA is developed which drastically reduces the area and delay to a great extent.

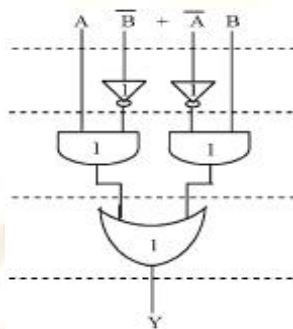
The CSLA is used in many computational systems design to moderate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. It uses independent ripple carry adders (for  $C_{in}=0$  and  $C_{in}=1$ ) to generate the resultant sum. However, the Regular CSLA is not area and speed efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input. The final sum and carry are selected by the multiplexers (mux). Due to the use of two independent RCA the area will increase which leads an increase in delay. To overcome the above problem, the basic idea of the proposed work is to use n-bit binary to excess-1 code converters (BEC) to improve the speed of addition. This logic can be replaced in RCA for  $C_{in}=1$  to further improves the speed and thus reduces the delay. Using Binary to Excess-1 Converter (BEC) instead of RCA in the regular CSLA will achieve lower area, delay which speeds up the addition operation. The main advantage of this BEC logic comes from the lesser number of logic gates than the Full Adder (FA) structure because the number of gates used will be decreased.

This work in brief is structured as follows. Section II deals with the delay and area evaluation methodology of the basic adder blocks and its corresponding delay and area values. Section III deals with the structure and function of BEC logic and its corresponding function table and logic equations. Section IV presents the architecture of the Regular CSLA of 128-bits. This SQRT CSLA has been developed using ripple carry adders and multiplexers. The architecture of the Modified SQRT CSLA is presented in Sections V. In section VI implementation methodologies and corresponding design tools are explained and finally the paper is concluded in section VIII.

### **II. BASIC ADDER BLOCK**

The adder block using a Ripple carry adder, BEC and Mux is explained in this section. In this we

calculate and explain the delay & area using the theoretical approach and show how the delay and area effect the total implementation. The AND, OR, and Inverter (AOI) implementation of an XOR gate is shown in Fig. 1. 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. Based on this approach, the blocks of 2:1 mux, Half Adder (HA), and FA are evaluated and listed in Table I.



**Fig 1: Delay and area evolution of xor**

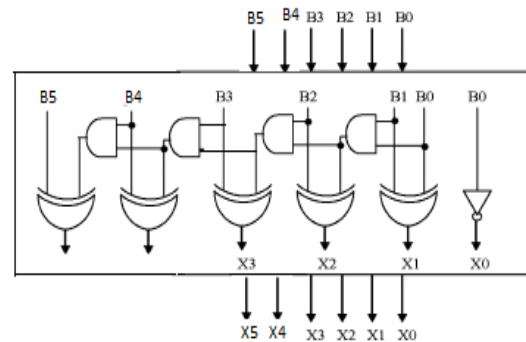
Design	Delay	Area
XOR	3	5
2:1 MUX	3	4
Half Adder	3	6
Full Adder	6	13

**Table 1: Delay and area evaluation of CSLA**

addition is performed using ripple carry adder and for  $C_{in}=1$  the operation is performed using 6-bit BEC (replacing the RCA for  $C_{in}=1$ ). The resultant is selected based on Carry in signal from the previous group. The total delay depends on mux delay and  $C_{in}$  signal from previous group.

**III. BEC**

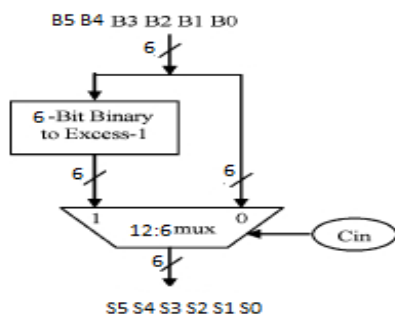
The basic work is to use Binary to Excess-1 Converter (BEC) in the regular CSLA to achieve lower area and increased speed of operation. This logic is replaced in RCA with  $C_{in}=1$ . This logic can be implemented for different bits which are used in the modified design. The main advantage of this BEC logic comes from the fact that it uses lesser number of logic gates than the n-bit Full Adder (FA) structure. As stated above the main idea of this work is to use BEC instead of the RCA with  $C_{in}=1$  in order to reduce the area and increase the speed of operation in the regular CSLA to obtain modified CSLA. To replace the n-bit RCA, an n+1 bit BEC logic is required. The structure and the function table of a 6-bit BEC are shown in Figure.2 and Table .2, respectively.



**Figure 2: 6-binary to excess-1 converter**

**Table 2: Function table of the 6-bit BEC**

B[5:0]	X[5:0]
000000	000001
000001	000010
000010	000011
111111	000000



**Fig 2: 6-bit BEC with 12:6 mux**

Fig 2 shows the basic 6-bit addition operation which includes 6-bit data, a 6-bit BEC logic and 12:6 mux. The addition operation is performed for  $C_{in}=0$  and for  $C_{in}=1$ . For  $C_{in}=0$  the

The Boolean expressions for the 6-bit BEC logic are expressed below

$X_0 = \sim B_0$

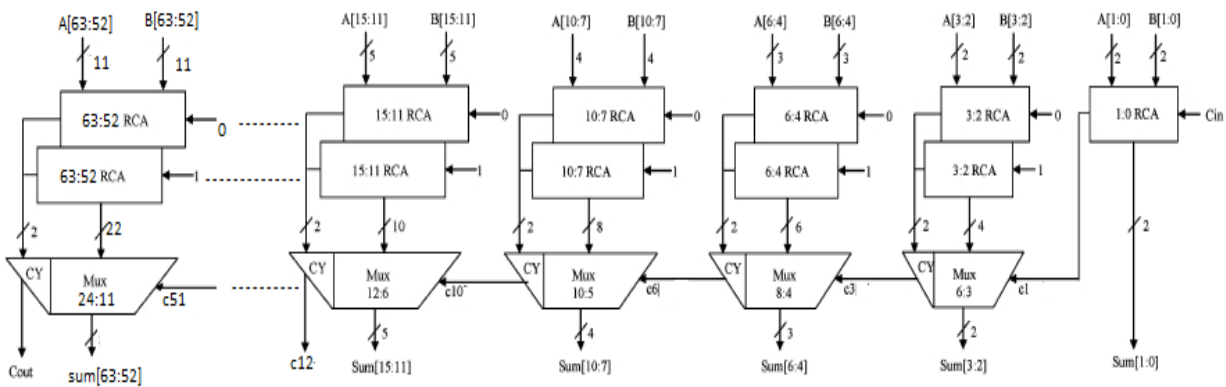
$X_1 = B_0 \wedge B_1$

$$X2 = B2 \wedge (B0 \& B1)$$

$$X3 = B3 \wedge (B0 \& B1 \& B2)$$

$$X4 = B4 \wedge (B0 \& B1 \& B2 \& B3)$$

$$X5 = B5 \wedge (B0 \& B1 \& B2 \& B3 \& B4)$$



**Figure 3: Architecture of Regular 64-bit Sqrt CSLA**

#### IV. ARCHITECTURE OF REGULAR 64-BIT Sqrt CSLA

A 16-bit carry select adder can be developed in two different sizes namely uniform block size and variable block size. Similarly a 32, 64 and 128-bit can also be developed in two modes of different block sizes. Ripple-carry adders are the simplest and most compact full adders, but their performance is limited by a carry that must propagate from the least-significant bit to the most-significant bit. The various 16, 32, 64 and 128-bit CSLA can also be developed by using ripple carry adders. The speed of a carry-select adder can be improved upto 40% to 90%, by performing the additions in parallel, and reducing the maximum carry delay.

Fig 3 shows the Regular structure of 64-bit Sqrt CSLA. It includes many ripple carry adders of variable sizes which are divided into groups. Group 0 contains 2-bit RCA which contains only one ripple carry adder which adds the input bits and the input carry and results to sum [1:0] and the carry out. The carry out of the Group 0 which acts as the selection input to mux which is in group 1, selects the result from the corresponding RCA (Cin=0) or RCA (Cin=1). Similarly the remaining groups will be selected depending on the Cout from the previous groups.

In Regular CSLA, there is only one RCA to perform the addition of the least significant bits [1:0]. The remaining bits (other than LSBs), the addition is performed by using two RCAs corresponding to the one assuming a carry-in of 0, the other a carry-in of 1 within a group. In a group, there are two RCAs that receives the same data inputs but different Cin. The upper adder has a carry-in of 0, the lower adder a carry-in of 1. The actual Cin from the preceding

sector selects one of the two RCAs. That is, as shown in the Fig.3, 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.

For this Regular CSLA architecture, the implementation code, for the Full Adders and Multiplexers of different sizes (6:3, 8:4, 10:5 up to 24:11) were designed initially. The regular 64-bit, 128-bit CSLA were implemented by calling the ripple carry adders and all multiplexers.

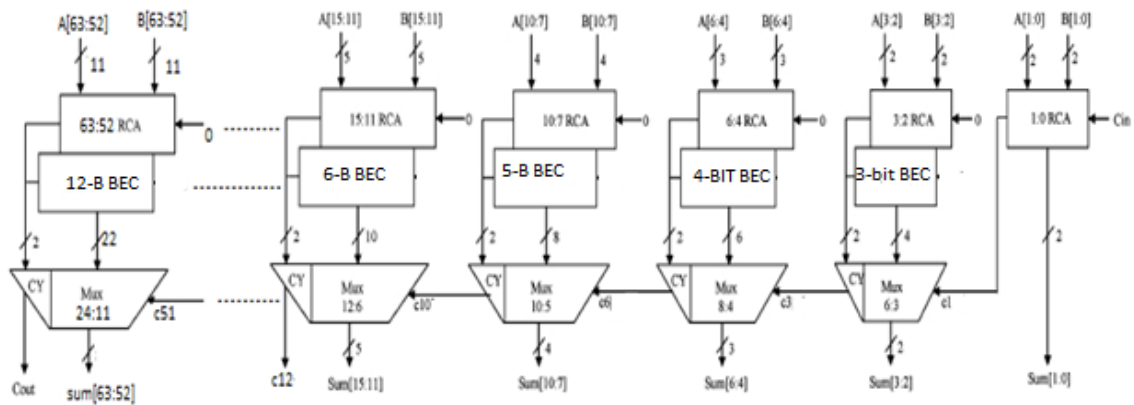
#### V. ARCHITECTURE OF MODIFIED 64-BIT Sqrt CSLA

This architecture is similar to regular 64-bit Sqrt CSLA, the only change is that, we replace RCA with Cin=1 among the two available RCAs in a group with a BEC. This BEC has a feature that it can perform the similar operation as that of the replaced RCA with Cin=1. Fig 4 shows the Modified block diagram of 64-bit Sqrt CSLA. The number of bits required for BEC logic is 1 bit more than the RCA bits. The modified block diagram is also divided into various groups of variable sizes of bits with each group having the ripple carry adders, BEC and corresponding mux. As shown in the Fig.4, Group 0 contain one RCA only which is having input of lower significant bit and carry in bit and produces result of sum[1:0] and carry out which is acting as mux selection line for the next group, similarly the procedure continues for higher groups but they includes BEC logic instead of RCA with Cin=1. Based on the consideration of delay values, the arrival time of selection input C1 of 8:3 mux is earlier than the sum of RCA and BEC. For remaining groups the selection input arrival is later than the RCA and BEC. Thus, the sum1 and c1



(output from mux) are depending on mux and results computed by RCA and BEC respectively. The sum2 depends on c1 and mux. For the remaining parts the arrival time of mux selection input is always greater than the arrival time of data inputs from the BEC's. Thus, the delay of the remaining MUX depends on the arrival time of mux

selection input and the mux delay. In this Modified CSLA architecture, the implementation code for Full Adder and Multiplexers of 6:3, 8:4, and 10:5 up to 24:11 were designed. The design code for the BEC was designed by using NOT, XOR and AND gates. Then 2, 3, 4, 5 up to 11-bit ripple carry adder was designed.



**Figure 4: Architecture of Modified 64-bit Sqrt CSLA**

**Table 3: Comparison values**

Sl. No.	Adders		Delay (ns)	Area
1.	16 – bit	Regular	16.27	43
		Modified	14.67	47
2.	32 – bit	Regular	20.96	90
		Modified	18.83	102
3.	64 – bit	Regular	33.85	189
		Modified	23.71	212
4.	128 – bit	Regular	42.23	439
		Modified	35.29	441

**VI.RESULTS**

The implemented design in this work has been simulated using Verilog-HDL (Modelsim). The adders (of various size 16, 32, 64 and 128) are designed and simulated using Modelsim. All the V files (Regular and Modified) are also simulated in Modelsim and corresponding results are compared. After simulation the different size codes are synthesized using Xilinx ISE 9.1i. The simulated V files are imported into the synthesized tool and

corresponding values of delay and area are noted. The synthesized reports contain area and delay values for different sized adders. The similar design flow is followed for both the regular and modified Sqrt CSLA of different sizes.

Table 3 shows the comparison of regular and modified CSLA of various bits which includes Delay and area comparisons. From the table it is

clear that the delay decreases for 16-bit modified method when compared with regular method. Similarly the table also shows the comparison for the various 32, 64, and 128 bits.

Midwest Symp. Circuits and Systems, 1995, pp. 1014–1017

The comparative values of areas shows that the number of LUT will be more for modified method for the 16, 32 and 64. This value decreases gradually for 128 bits. For 256 bits the value almost equal to regular method which will reduces more for still higher order bits. Thus the modified method decreases the delay and also area to a great extent.

## VII. ACKNOWLEDGMENT

K.Allipeera would like to thank Mr. S. Ahmed Basha, Assitant professor ECE Department who had been guiding throughout the project and supporting me in giving technical ideas about the paper and motivating me to complete the work effectively and successfully.

## VIII. CONCLUSION

An efficient approach is proposed in this paper to reduce the area and delay of SQRT CSLA architecture. The reduction in the number of gates is obtained by simply replacing the RCA with BEC in the structure. The compared results shows that the modified SQRT CSLA has a slightly larger area for lower order bits which further reduces for higher order bits. The delay is reduced to a great extent with the modified SQRT CSLA. Thus the results shows that using modified method the area and delay will decrease thus leads to good alternative for adder implementation for many processors. The modified CSLA architecture is therefore low area and high speed approaches for VLSI hardware implementation.

## REFERENCES

1. Bedrij, O. J., (1962), "Carry-select adder," IRE Trans. Electron. Comput. Pp.340–344.
2. Ramkumar,B. , Kittur, H.M. and Kannan ,P. M.,(2010 ),"ASIC implementation of modified faster carry save adder," Eur. J. Sci. Res., vol. 42, no. 1,pp.53–58.
3. Kim ,Y. and Kim ,L.-S.,(May2001), "64-bit carry-select adder with reduced area," Electron Lett., vol. 37, no. 10, pp. 614–615.
4. Ceiang, T. Y. and Hsiao. J., (Oct 1998),"Carry-select adder using single ripple carry adder," Electron. Lett., vol. 34, no. 22, pp. 2101– 2103
5. He, Y., Chang, C. H. and Gu, J., (2005), "An A rea efficient 64-bit square root carry-select adder for low power application," in Proc. IEEE Int. Symp.Circuits Syst. vol. 4, pp. 4082–4085.
6. E. Abu-Shama and M. Bayoumi, "A New cell for low power adders," in Proc.Int.