

# DESIGN AND IMPLEMENTATION OF 64-BIT CARRY SELECT ADDER IN FPGA

Shaik Magbul Basha<sup>1</sup>

[magbul1000@gmail.com](mailto:magbul1000@gmail.com)<sup>1</sup>

L. Srinivas Reddy<sup>2</sup>

[lsr.ngi@gmail.com](mailto:lsr.ngi@gmail.com)<sup>2</sup>

<sup>1</sup>UG Scholar, Dept of ECE, Nalanda Group of Institutions, Kantepudi, Sattenapalli, Guntur, A.P.

<sup>2</sup>Associate Professor, Dept of ECE, Nalanda Group of Institutions, Kantepudi, Sattenapalli, Guntur, A.P.

**Abstract:** In several data-processing processors Carry Select Adder (CSLA) is one quickest adders used to perform arithmetic functions. The approaching technologies depicts that there is a scope for reducing the space and power consumption in the CSLA. This work uses a straightforward gate level modification to considerably cut back the space and power of the CSLA. Based mostly on this modification CSLA designs have been developed and will be compared with the regular CSLA design. The projected style has reduced space and power as compared with the regular CSLA with solely a slight increase in the delay. This work evaluates the performance of the projected styles in terms of delay, area, power, and their product by hand with logical effort and through custom style and layout in zero.18-m CMOS method technology. The results analysis shows that the projected CSLA structure is healthier than the regular CSLA.

Keywords- Delay; Area; Array Multiplier, low power, VHDL Modeling & Simulation.

## I. Introduction

Area and power reduction in data path logic systems are the main area 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. The major speed limitation in any adder is in the production of carries and many authors have considered the addition problem. The CSLA is used in many computational systems to moderate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and

carry by considering carry input and then the final sum and carry are selected by the multiplexers (mux). To overcome above problem, the basic idea of the proposed work is by using n-bit binary to excess-1 code converters (BEC) to improve the speed of addition. This logic can be implemented with any type of adder to further improve the speed. Using Binary to Excess -1 Converter (BEC) instead of RCA 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 Full Adder (FA) structure.

## II. Literature Survey

On the basis of necessities such as space, delay and power consumption a number of the advanced adders square measure Ripple Carry Adder, Carry look-Ahead Adder and Carry choose Adder. Ripple Carry Adder (RCA) shows the compact style however their computation time is longer. Time essential applications build use of Carry Look-Ahead Adder (CLA) to derive quick results however it leads to increase in space. However the carry choose adder provides a compromise between tiny the tiny the little spaces however longer delay of RCA and giant area with small delay of Carry Look Ahead adder. Ripple Carry Adder consists of cascaded "N" single bit full adders. Output carry of previous adder becomes the input carry of next full adder. Therefore, the carry of this adder traverses longest path known as worst case delay path through N stages. Fig. one shows the diagram of ripple carry adder. Currently as the price of N will increase, delay of adder can additionally increase in a linear manner. Therefore, RCA has the lowest speed amongst all the adders because of large propagation delay but it occupies the least area. Now CSLA provides a way to get around this linear dependency is to anticipate all possible values of input carry i.e. 0 and 1 and evaluate the result in advance. Once the original value of carry is known, result can be selected using the multiplexer stage. Therefore the conventional CSLA makes use of Dual RCA's to generate the partial sum and carry by

considering input carry  $C_{in}=0$  and  $C_{in}=1$ , then the final sum and carry are selected by multiplexers.

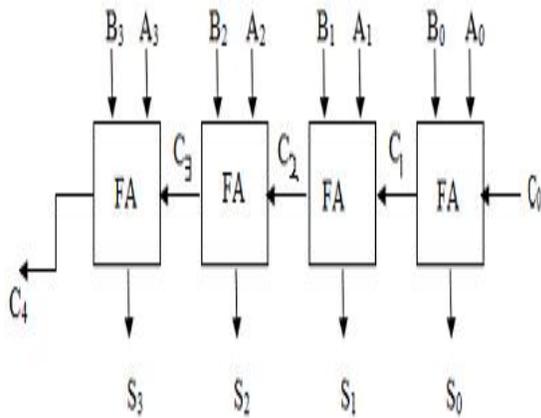


Fig. 1 4-bit Ripple Carry Adder

### III. BEC

The basic idea of this work is to use Binary to Excess-1 converter (BEC) instead of RCA with  $C_{in}=1$  in conventional CSLA in order to reduce the area and power. BEC uses less number of logic gates than N-bit full adder structure. To replace N-bit RCA, an N+1 bit BEC is required. Therefore, Modified CSLA has low power and less area than conventional CSLA. SQRT CSLA has been chosen for comparison with modified design using BEC as it has more balanced delay, less area and low power. Regular SQRT CSLA also uses dual RCAs. In order to reduce the delay, area and power, the design is modified by using BEC instead of RCA with  $C_{in}=1$ . Therefore, the modified SQRT CSLA occupies less area, delay and low power. Further also, the parameters like delay, area and power can be reduced.

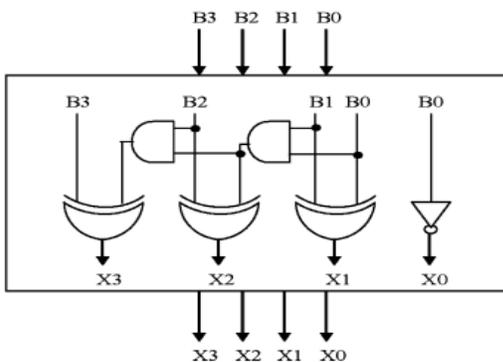


Fig. 2. 4-b BEC

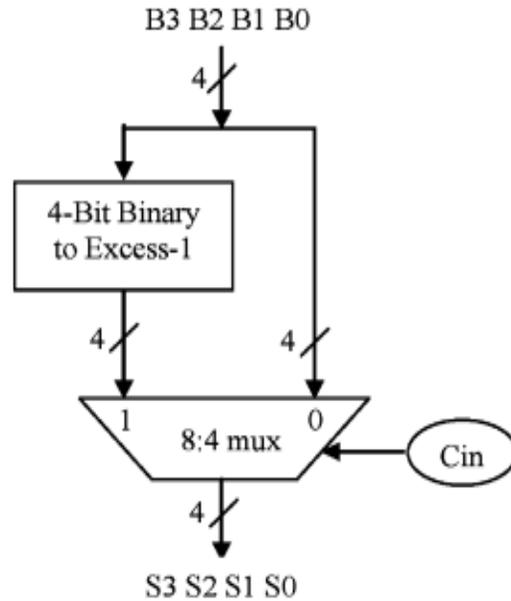


Fig. 3. 4-b BEC with 8:4 mux.

TABLE I  
TRUTH TABLE OF 4-BIT BINARY TO EXCESS-1  
CONVERTER

Binary Logic $B_0 B_1 B_2 B_3$	Excess-1 Logic $X_0 X_1 X_2 X_3$
0000	0001
0001	0010
0010	0011
0011	0100
0100	0101
0101	0110
0110	0111
0111	1000
1000	1001
1001	1010
1010	1011
1011	1100
1100	1101
1101	1110
1110	1111
1111	0000

Fig. 4 shows the 16-bit Conventional CSLA. The conventional CSLA is area consuming due to the use of dual RCA's.

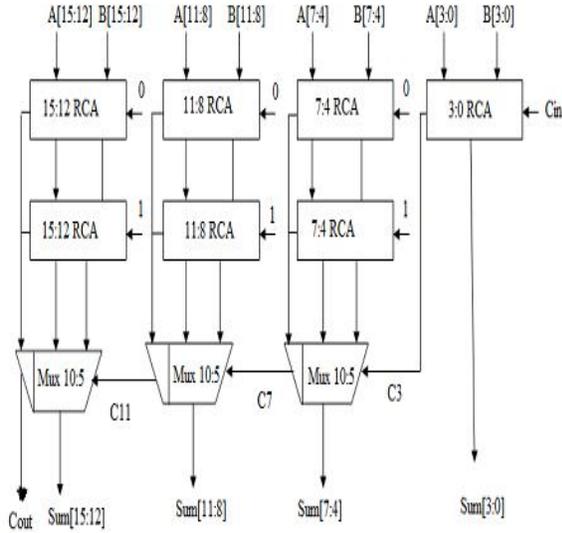


Fig. 4 16-bit conventional carry select adder

Modified Sqrt CSLA is similar to that of regular Sqrt CSLA, the only difference is we replace RCA with  $C_{in}=1$  with BEC. This replaced BEC performs the same operation as that of the replaced RCA with  $C_{in}=1$ . Fig. 5 shows the block diagram of modified Sqrt CSLA. This structure consumes less area; delay and power than regular Sqrt CSLA because of less number of transistors are used.

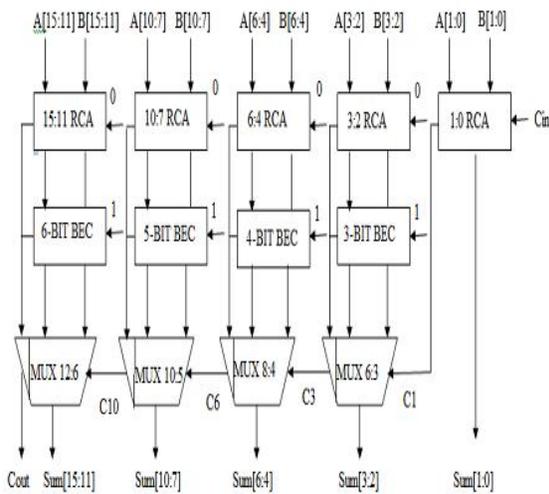


Fig. 5 Modified 16-bit Sqrt CSLA

The CSLA has two units: 1) the sum and carry generator unit (SCG) and 2) the sum and carry selection unit [9]. 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 [6] 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.

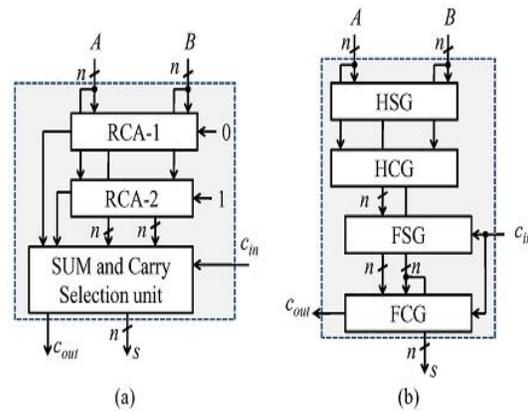


Fig. 6. (a) Conventional CSLA;  $n$  is the input operand bit-width. (b) The logic operations of the RCA is shown in split form,

Where HSG, HCG, FSG, and FCG represent half-sum generation, half-carry generation, full-sum generation, and full-carry generation, respectively.

#### IV. Proposed Adder Design

The proposed CSLA is based on the logic formulation given in (4a)–(4g), and its structure is shown in Fig. 3(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 words  $0$  and half-carry word  $c_0$  of width  $n$  bits each. Both CG0 and CG1 receive  $s_0$  and  $c_0$  from the HSG unit and generate two  $n$ -bit full-carry words  $c_{01}$  and  $c_{11}$  corresponding to input-carry '0' and '1', respectively. The logic diagram of the HSG unit is shown in Fig. 3(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 Fig. 3(c) and (d), respectively.

The CS unit selects one final carry word from the two carry words available at its input line using the control signal  $c_{in}$ . It selects  $c_0$  when  $c_{in} = 0$ ; otherwise, it selects  $c_1$ . The CS unit can be implemented using an  $n$ -bit 2-to-1 MUX. However, we find from the truth table of the CS unit that carry words  $c_0$  and  $c_1$  follow a specific bit pattern. If  $c_0(i) = '1'$ , then  $c_1(i) = 1$ , irrespective of  $s_0(i)$  and  $c_0(i)$ , for  $0 \leq i \leq n-1$ . This feature is used for logic optimization of the CS unit. The optimized design of the CS unit is shown in Fig. 3(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  $c_{out}$ , and  $(n-1)$  LSBs are XORed with  $(n-1)$  MSBs of half-sum ( $s_0$ ) in the FSG [shown in Fig. 3(f)] to obtain  $(n-1)$  MSBs of final-sum ( $s$ ). The LSB of  $s_0$  is XORed with  $c_{in}$  to obtain the LSB of  $s$ .

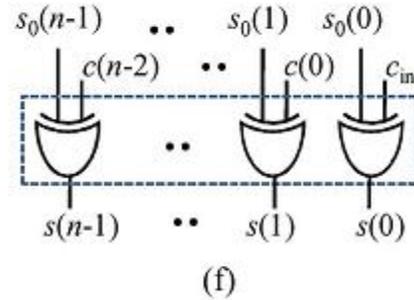
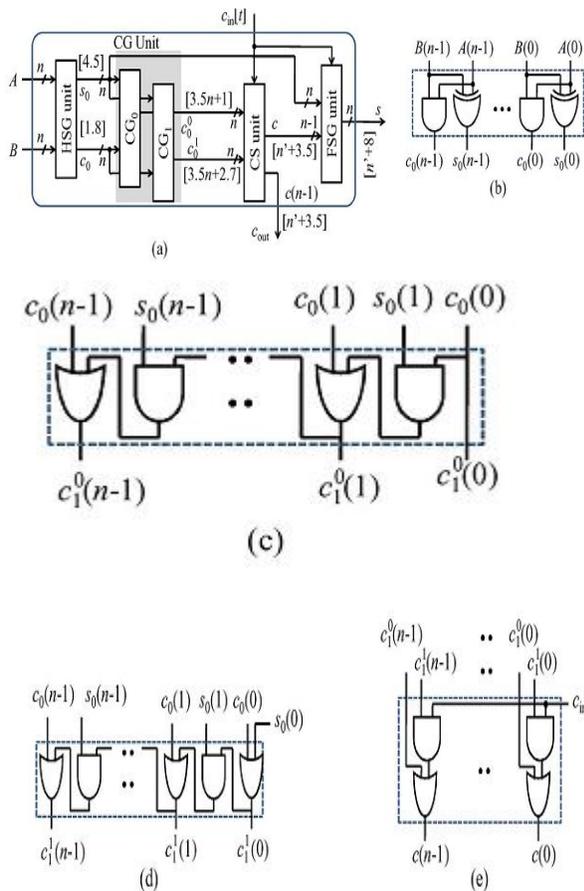


Fig. 7. (a) Proposed CS adder design, where  $n$  is the input operand bit-width, and  $[*]$  represents delay (in the unit of inverter delay),  $n = \max(t, 3.5n + 2.7)$ . (b) Gate-level design of the HSG. (c) Gate-level optimized design of (CG0) for input-carry=0. (d) Gate-level optimized design of (CG1) for input-carry=1. (e) Gate-level design of the CS unit. (f) Gate-level design of the final-sum generation (FSG) unit.

## V. RESULTS

The implemented design in this work has been simulated using Verilog-HDL. The adders (of various size 16, 32, 64) are designed and simulated using Xilinx ISE 13.2. The simulated 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 CSLA of different sizes.

Simulation Results:  
16 bit:

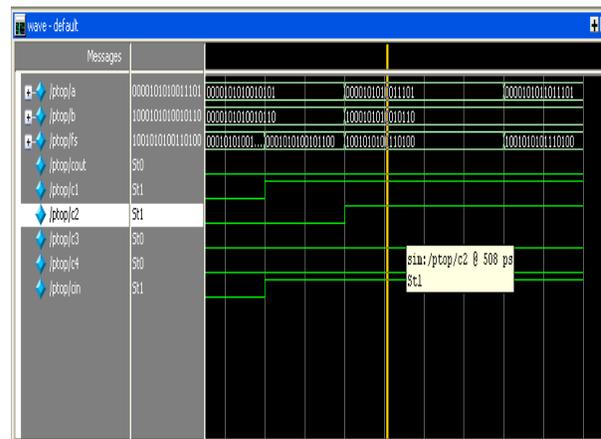


Fig 8. 16 bit CSLA

64 bit:

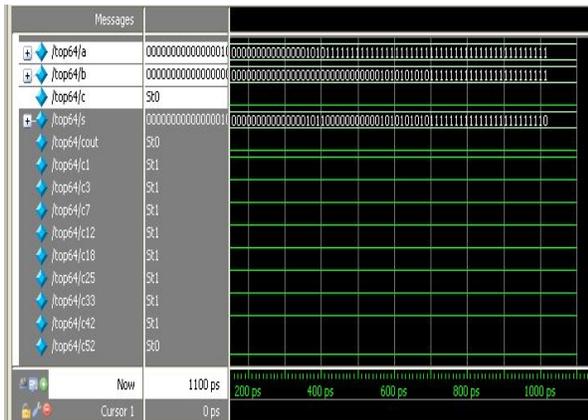


Fig 9. 64 bit CSLA

Synthesis Results:

RTL schematic:

16 bit:

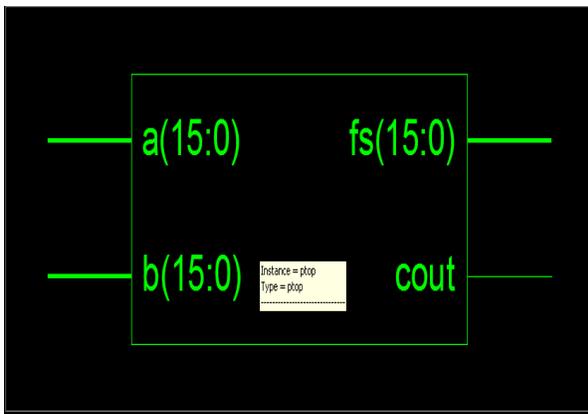


Fig 10. 16 bit CSLA

64 bit:

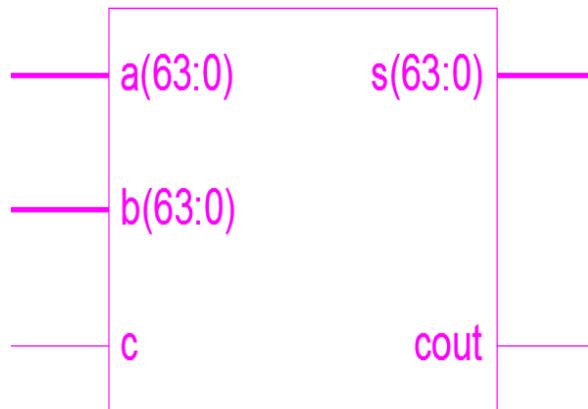


Fig 11. 64 bit CSLA

Design Summary:

16 bit:

reel Project Status (04/22/2015 - 12:59:54)			
Project File:	reel.isc	Current State:	Synthesized
Module Name:	ptop	• Errors:	No Errors
Target Device:	xc3s500e-5fg320	• Warnings:	5 Warnings
Product Version:	ISE 10.1 - Foundation Simulator	• Routing Results:	
Design Goal:	Balanced	• Timing Constraints:	
Design Strategy:	>Xilinx Default (unlocked)	• Final Timing Score:	

reel Partition Summary	
No partition information was found.	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	22	4656	0%
Number of 4 input LUTs	40	9312	0%
Number of bonded IOBs	49	232	21%

Fig 12. 16 bit CSLA

64 Bit:

ww Project Status (09/12/2013 - 10:50:00)			
Project File:	ww.isc	Current State:	Synthesized
Module Name:	top64	• Errors:	No Errors
Target Device:	xc3s500e-5fg320	• Warnings:	1 Warning
Product Version:	ISE 10.1 - Foundation Simulator	• Routing Results:	
Design Goal:	Balanced	• Timing Constraints:	
Design Strategy:	>Xilinx Default (unlocked)	• Final Timing Score:	

ww Partition Summary	
No partition information was found.	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	118	4656	2%
Number of 4 input LUTs	223	9312	2%
Number of bonded IOBs	203	232	87%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Thu Sep 12 10:49:59 2013	0	1 Warning	0
Translation Report					
Map Report					
Place and Route Report					

Fig 13. 64 bit CSLA

## VI. Conclusion

Power, delay and space square measure the constituent factors in VLSI style that limits the performance of any circuit. This work presents a straightforward approach to scale back the space, delay and power of CSLA design. The traditional carry choose adder has the disadvantage of additional power consumption and occupying additional additional chip space. The projected SQRT CSLA mistreatment common Boolean logic has low power, less delay and reduced space than all the opposite adder structures. It's additionally bit quicker than all the opposite adders. During this method, the semiconductor count of projected SQRT CSLA is reduced having less space and low power that makes

it straightforward and economical for VLSI hardware implementations.

### References

- [1] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput. 1962.
- [2] B. Ramkumar, H.M. Kittur, and P. M. Kannan, "ASIC implementation of modified faster carry save adder," Eur. J. Sci. Res., vol. 41, no. 1, 2009.
- [3] 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.
- [4] Y. Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," Electron. Lett. vol. 36, no. 9, May 2000.
- [5] J. M. Rabaey, Digital Integrated Circuits—A Design Perspective. Upper Saddle River, NJ: Prentice-Hall, 2001.
- [6] Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for low power applications," in Proc. IEEE Int. Symp. Circuits Syst., 2005, vol. 4, pp. 4082–4085.
- [7] I-Chyn Wey, Cheng-Chen Ho, Yi-Sheng Lin, and Chien-Chang Peng, "An Area-Efficient Carry Select Adder Design by Sharing the Common Boolean Logic Term", Proceedings of the International Multi Conference of Engineers and Computer Scientist 2012 Vol II, IMCES 2012, HongKong, March 14-16 2012.
- [8] Ms. S.Manjui, Mr. V. Sornagopae, "An Efficient SQRT Architecture of Carry Select Adder Design by Common Boolean Logic", IEEE, 2013.