

Design and Implementation of Low Power and Area Efficient CSA

Shrikant Ganorkar

PG Student Dept. of E&TC, SITS Narhe, Pune

Sandip Shrote

Asst. Prof. Dept. of E&TC, SITS Narhe, Pune

Abstract

Carry Select Adder (CSA) is one of the high speed adders used in many computational systems to perform fast arithmetic operations. However, conventional CSA is still area-consuming due to the dual ripple carry adder (RCA) structure. From the structure of regular CSA there is still scope to reduce area in CSA by introduction of some add-one scheme. This work uses a simple and efficient gate-level modification to significantly reduce the logic resources and power of the CSA. The Proposed Architecture is designed for 16-bit, 32-bit and 64-bit and then compared with conventional CSA architectures. The Proposed Architecture shows reduction in area and power consumption in comparison with conventional CSA. This project is aimed for implementing high performance optimized FPGA architecture. Modelsim 6.3f is used for simulating the CSA and synthesized using Xilinx ISE 12.3. Then the implementation is done on Spartan-3 FPGA Kit.

1. Introduction

Addition is the heart of computer arithmetic, and the arithmetic unit is often the work horse of a computational circuit. They are the necessary component of a data path, e.g. in microprocessors or a signal processor. Design of area efficient high speed data path logic systems are one of the most essential areas of research in VLSI. In digital adders, the speed of addition is controlled 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. O. J. Bedriji [2] proposed that the problem of carry propagation delay is overcome by independently

generating multiple radix carries and using this carries to select between simultaneously generated sums. Akhilash Tyagi introduced a scheme to generate carry bits with block carry in 1 from the carries of a block with block carry in 0 [3]. Chang and Hsiao [4] proposed that instead of using dual Ripple Carry Adder a Carry Select Adder scheme using an add one circuit to replace one RCA. Youngjoon Kim and Lee Sup Kim introduced a multiplexer based add one circuit was proposed to reduce the area with negligible speed penalty. Yajuan He et al proposed an area efficient Square-root CSLA scheme based on a new first zero detection logic [5]. Ramkumar et al proposed a Binary to Excess-1 Converter (BEC) method to reduce the maximum delay of carry propagation in final stage of carry save adder [1]. Ramkumar and Harish [6] proposed BEC technique, which is a simple and efficient gate level modification to significantly reduce the area of SQRT CSA. CSA is used in many computational systems to modify the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum [2]. However, the CSLA is not area efficient because it uses multiple pairs of RCA to generate partial sum and carry by considering carry in 0 and carry in 1, then the final sum and carry are selected by the multiplexers (Mux).

The basic idea of this work is to use BEC instead of RCA with carry in 1 in the regular CSA to achieve lower area and power [1], [3] and [7]. The main benefit of BEC comes from the lesser number of logic gates than the n-bit Full Adder (FA). The details of BEC are discussed in section III. Section II deals with the area evaluation methodology of the basic adder blocks and presents the structure of the XOR gate. Section III presents the detailed structure of BEC logic. The regular CSA has been chosen for comparison with the

proposed design as it has a lower area and power [9]. The area evaluation methodology of the regular CSA and modified CSA are presented in section IV and V respectively. The FPGA implementation details and results are analyzed in section VI. Finally this work is concluded in section VII.

2. Area Evaluation Methodology of Basic Adder Blocks

The implementation of an XOR gate by using AND, OR and Inverter is shown in fig. 1. The gates between the dotted lines are performing the operation in parallel. The area evaluation methodology considers all gates to be made up of AND, OR and Inverter (AOI), each having area equal to one unit. The area evaluation is carried out by counting the total number of AOI gates required for each logic block. By using this approach, the CSA blocks of 2:1 Mux, Half Adder (HA) and Full Adder (FA) are evaluated and listed in Table I.

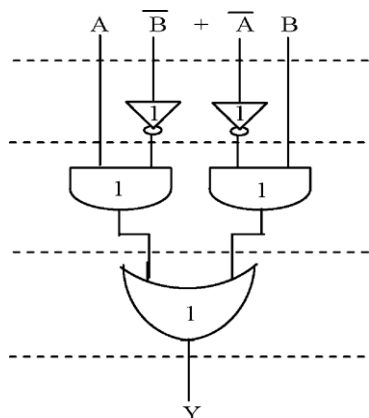


Fig. 1. Area evaluation of an XOR gate

TABLE I

AREA COUNT OF THE BASIC BLOCKS OF CSA

Adder Blocks	Area Count
XOR	5
2:1 MUX	4
Half Adder	6
Full Adder	13

3. Binary to Excess-1 Converter (BEC)

The main idea of this work is to use BEC instead of RCA with carry in=1 in order to reduce area of the conventional CSA. To replace the n-bit RCA an n+1 bit BEC is required. A structure and function of 3-bit BEC are shown in Fig. 2 and Table II respectively.

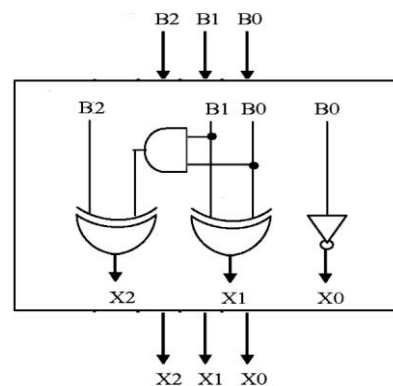


Fig. 2. 3-bit BEC

TABLE II

FUNCTION TABLE OF 3-BIT BEC

B[2:0]	X[2:0]
000	001
001	010
010	011
011	100
100	101
101	110
110	111
111	000

The Boolean expressions of the 3-bit BEC are shown below (note the functional symbols ~ NOT, & AND, ^ XOR)

$$X0 = \sim B0$$

$$X1 = B0 \wedge B1$$

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

4. Area Evaluation Methodology of 16-bit Regular CSA

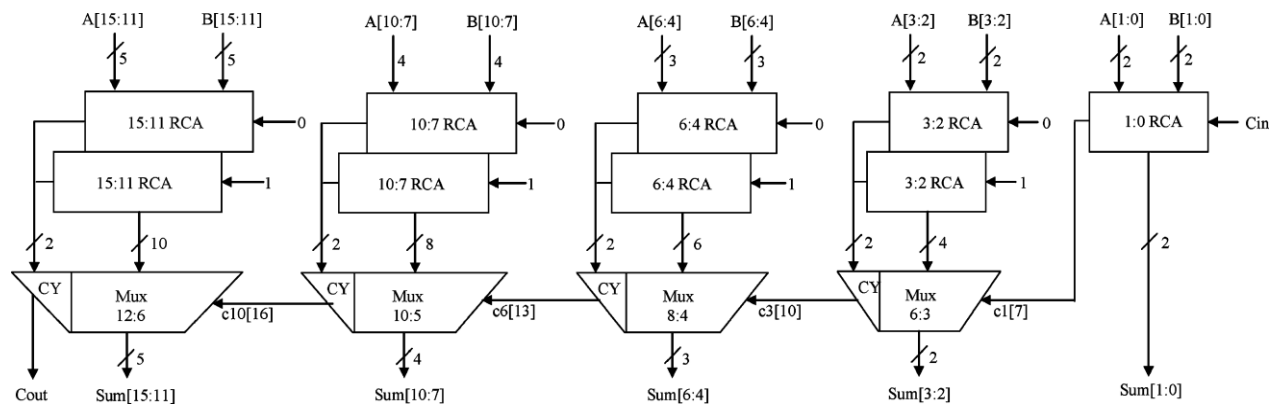


Fig. 3. Regular 16-bit Sqrt Architecture

The structure of the 16-bit regular CSA is shown in Fig. 3. It has 5 groups of same size RCA. Each group contains two RCA and a MUX. It achieves the addition by adding small portions of bits each having equal size and waits for the carry to complete the calculation. Sum and Carry both are calculated for possible solutions. The regular Sqrt CSA is constructed by equalizing the delay through two carry chains and the block multiplexer signal from previous stage. The steps leading to evaluation are as below. In the regular Sqrt CSA, the group2 has two sets of 2-bit RCA. The selection input of 3:2 Mux is C1. If the C1 = 0, the Mux select first RCA output else it selects second RCA output. The output of group2 are Sum [3:2] and carryout, C3. The area count of group2 is determined as follows:

$$\begin{aligned} \text{Gate count} &= 57 \text{ (FA + HA + MUX)} \\ \text{FA} &= 39 \text{ (3 * 13)} \\ \text{HA} &= 6 \text{ (1 * 6)} \\ \text{Mux} &= 12 \text{ (3 * 4)} \end{aligned}$$

Similarly the estimated area of the other groups in the regular Sqrt CSA are evaluated and listed in Table III.

TABLE III
AREA COUNT OF THE 16-BIT REGULAR CSA GROUPS

Group	Area Count
Group 1	26
Group 2	57
Group 3	87
Group 4	117
Group 5	147

5. Area Evaluation Methodology of 16-bit Modified CSA

The structure of the modified 16-bit CSA with carry in = 1 to optimize the area is shown in Fig. 4. The 16-bit modified CSA has 5 groups of different size RCA and BEC. Each group contains one RCA, one BEC and a Mux. In the modified CSA the group 2 has one 2-bit RCA which has 1 FA and 1 HA for carry in = 0. A 3-bit BEC is used which adds one to the output from 2-bit RCA. The selection input of 6:3 Mux is C3. The area count of group2 is determined as follows:

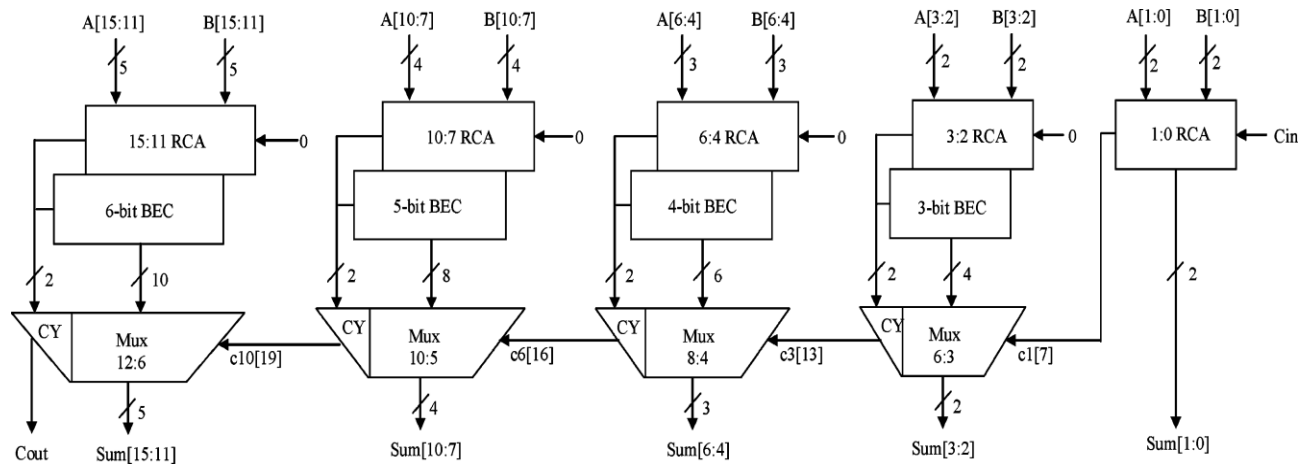


Fig. 4. Modified 16-bit Sqrt Architecture

Gate count = 43 (FA + HA + Mux + BEC)

FA = 13 (1 * 13)

HA = 6 (1 * 6)

Mux = 12 (3 * 4)

NOT = 1

AND = 1

XOR = 10 (2 * 5)

BEC (3-BIT) = NOT + AND + XOR = 12

Similarly the estimated area of the other groups in the modified Sqrt CSA are evaluated and listed in Table IV.

TABLE IV
AREA COUNT OF THE 16-BIT MODIFIED CSA GROUPS

Group	Area Count
Group 1	26
Group 2	43
Group 3	62
Group 4	86
Group 5	109

Comparing Tables III and IV, it is clear that the proposed modified Sqrt CSA saves 82 gate areas than the regular Sqrt CSA.

6. FPGA Implementation Results

This work has been developed using VHDL and synthesized using Xilinx ISE 12.3. The simulations are performed in Modelsim 6.3f simulator. The design is implemented on Spartan-3 FPGA kit. Fig. 5(a) shows the and Fig. 5(b) shows RTL block schematic of 64-bit CSA. The percentage reduction in area for 16-b, 32-b and 64-b is reduced by 21.21%, 33.03 %, 38 %. The total power consumption shows a similar trend of increasing reduction in power consumption by 34 %, 45 % and 42 % respectively. The Table-V exhibits the simulation results of regular CSA and modified CSA in terms of area and power. The gate reduction in the area as a function of the bit size is shown in Fig. 6. It has been proved from the Table-V that the modified 64-bit CSA is more area and power efficient for VLSI Implementations.

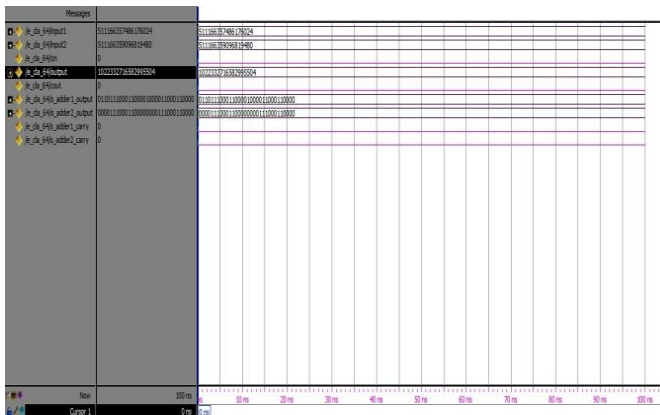


Fig. 5(a). Simulation result of 64-bit Modified CSA

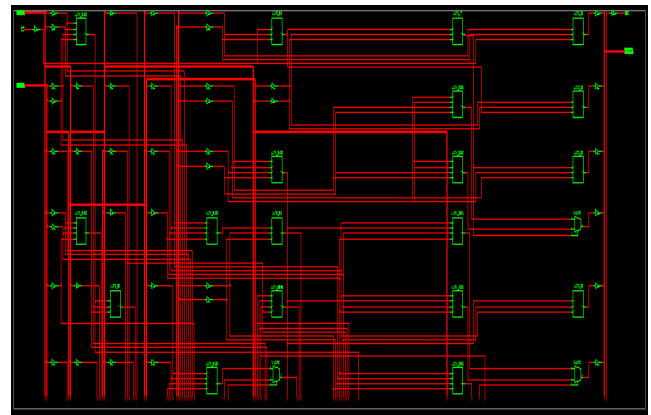


Fig. 5(b). RTL Schematic of 64-bit Modified CSA

TABLE V
COMPARISON OF THE REGULAR AND MODIFIED CSA

BIT SIZE	CSA TYPES	AREA (GATE COUNT)	POWER (mW)
16-bit	Regular	99	0.35
	Modified	78	0.23
32-bit	Regular	227	0.77
	Modified	152	0.42
64-bit	Regular	491	1.42
	Modified	302	0.82

7. Conclusion

A simple approach is proposed in this project to reduce the area and power of the CSA architecture. From the results it can be seen that the reduced number of gates of this work offers a great advantage in the reduction of area and power. The area count of 64-bit modified SQRT CSA is significantly reduced by 189 gates when compared with the 64-bit regular CSA architecture. Totally from the result analysis the area and power of the modified SQRT CSA are significantly reduced. The area of the proposed design shows a decrease for 16-bit, 32-bit and 64-bit sizes which indicate the success of the method and not a mere tradeoff of delay for area and power. The MCSA is therefore low area, low power, simple and efficient for VLSI hardware implementation.

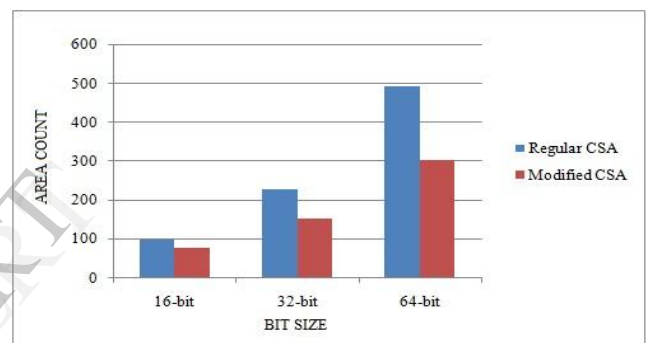


Fig. 6. Comparison of CSAs based on Area count

8. References

- [1] 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. 53–58, 2010.
- [2] O. J. Bedrij, "Carry-select adder," *IRE Trans. Electron. Comput.*, pp. 340–344, 1962.
- [3] Akhilesh Tyagi, "A Reduced-Area Scheme for Carry-Select Adders," *IEEE Transactions on Computers*, Vol.42, No.10, pp.1163–1170, 1993.
- [4] 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.
- [5] 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.
- [6] Ramkumar, B. and Harish M Kittur, "Low Power and Area Efficient Carry Select Adder", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp.1-5, 2012.
- [7] Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for lowpower applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, vol. 4, pp. 4082–4085.
- [8] W. Goh, S. Rofail, and K. Yeo, "Low-Power Design: An Overview", Prentice Hall.
- [9] J. M. Rabaey, *Digital Integrated Circuits—A Design Perspective*. Upper Saddle River, NJ: Prentice-Hall, 2001.