ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2, Issue 2, Ver. 3 (April - June 2014)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

# Modified Structure for Reduced Power and Area in Carry Select Adder

**Ganesh Kumar Veerabathiran**
Dept. of Electronics and Communication Engineering, Panimalar Engineering College, Anna University, Chennai, India

**Abstract**
*Design of area- and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. 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. 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 power consumption in the CSLA. This work uses a simple and efficient gate-level modification to significantly reduce the area and power of the CSLA. Based on this modification 8-, 16-, 32-, and 64-b square-root CSLA (SQRT CSLA) architecture have been developed and compared with the regular SQRT CSLA architecture. The proposed design has reduced area and power as compared with the regular SQRT CSLA with only a slight increase in the delay.*

**Keywords**
*Application-specific integrated circuit (ASIC), area-efficient, CSLA, low power.*

## I. Introduction

Design of area- and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. 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 CSLA is used in many computational systems to alleviate 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 $C_{in}$=0 and $C_{in}$=1, then the final sum and carry are selected by the multiplexers (mux).

The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA with $C_{in}$=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.

This brief is structured as follows. The delay and area evaluation methodology of the basic adder blocks. The detailed structure and the function of the BEC logic are represented. 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 ASIC implementation details and results are analyzed.

## II. Literature Review

The CSLA is uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input CIN=0 and CIN=1, then the final sum and carry are selected by the multiplexers (mux). The speed of addition is limited by the time required to propagate a carry through the adder. This system has developed to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum.

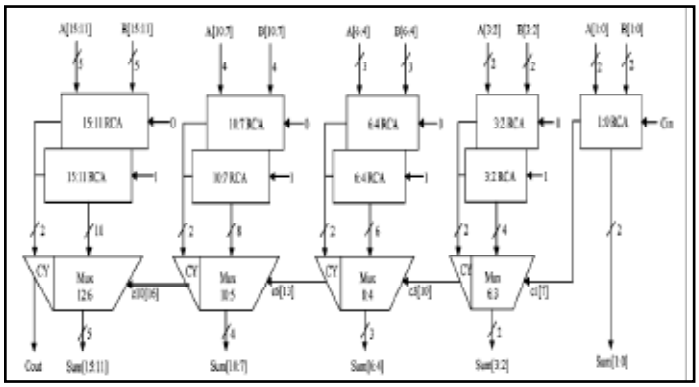


Fig. 1: Regular 16-b SQRT CSLA

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 CSLA is used in many computational systems to alleviate 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 CIN=0 and CIN=1, then the final sum and carry are selected by the multiplexers (mux).

Basic Adder Block

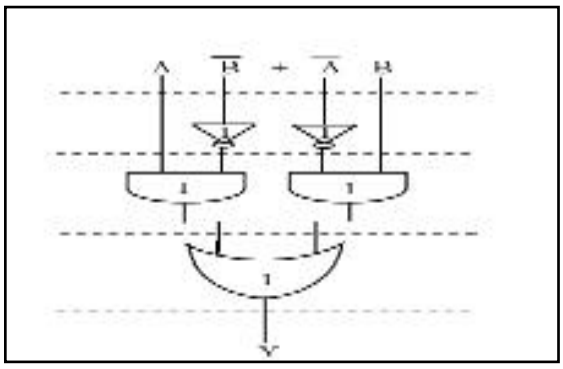


Fig. 2: Delay & Area Evaluation of XOR

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

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. 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. 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 & listed in table below.

Table 1: Delay & Area Evaluation of CSLA.

| Adder Blocks | Delay | Area |
|---|---|---|
| XOR | 3 | 5 |
| 2:1 MUX | 3 | 4 |
| Half Adder | 3 | 6 |
| Full Adder | 6 | 13 |

## III. Algorithm

The AND, OR, and Inverter (AOI) implementation of an XOR gate is required. 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.

Based on this approach, the CSLA adder blocks of 2:1 mux, Half Adder (HA), and FA are evaluated.

BEC (Binary to Excess-1 Converter) 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 $C_{in}$. 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)

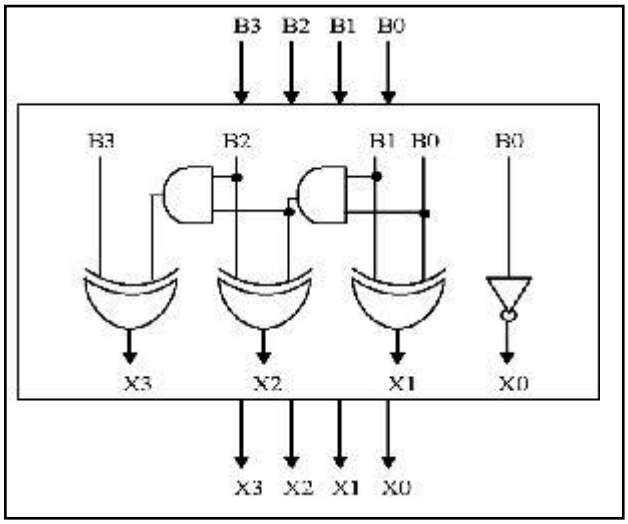


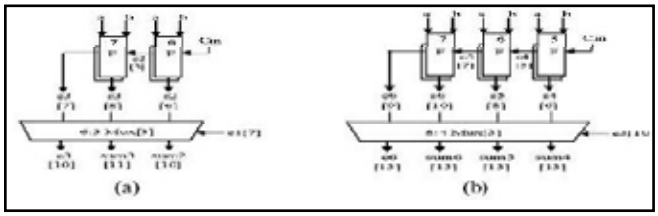
Fig. 3: 4-b BEC with 8:4 mux

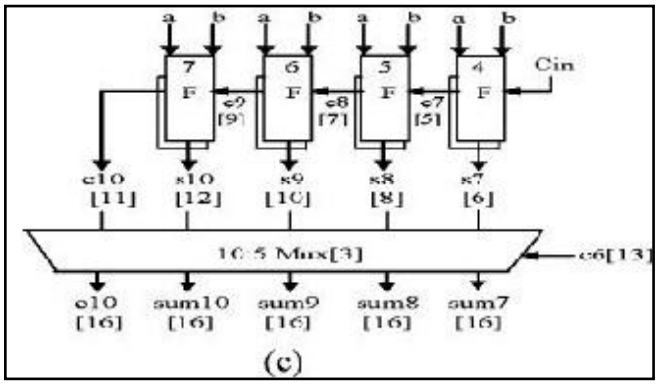
## IV. Delay And Area Evaluation Methodology Of Regular 18-B Sqrt Csla

The structure of the proposed 16-b SQRT CSLA using BEC for RCA with $C_{in}$=1 to optimize the area and power is shown below. We again split the structure into five groups. The delay and area estimation of each group are shown in below. The steps leading to the evaluation are given here.

1) The group2 has one 2-b RCA which has 1 FA and 1 HA for $C_{in}$=0 . Instead of another 2-b RCA with $C_{in}$=1 a 3-b BEC is used which adds one to the output from 2-b RCA. Based on the consideration of delay values, the arrival time of selection input C1[time(t)=7] of 6:3 mux is earlier than the s3[t=9] and c3[t=10] and later than the s2[t=4]. Thus, the sum3 and final C3(output from mux) are depending on s3and mux and partial C3(input to mux) and mux, respectively. The sum2 depends on C1 and mux.

2) For the remaining group's 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 groups depends on the arrival time of mux selection input and the mux delay.



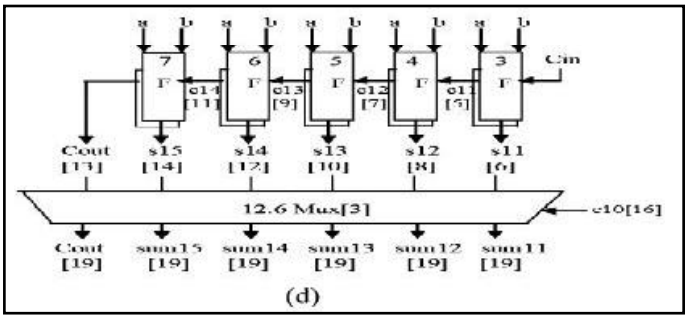



Fig 4: Delay and area evaluation of regular SQRT CSLA: (a) group2, (b)group3, (c) group4, and (d) group5

3) Except for group2, the arrival time of mux selection input is always greater than the arrival time of data outputs from the RCA s. The area count of group2 is determined as follows:

Gate Count = 43(FA+HA+Mux+BEC)
FA=13 (1*13)
HA=6 (1*6)
AND=1
NOT=1
XOR=10 (2*5)
Mux=12 (3*4)

4) Similarly, the estimated maximum delay and area of the other groups of the modified SQRT CSLA are evaluated. It is clear that the proposed modified SQRT CSLA saves 113 gate areas than the regular SQRT CSLA, with only 11 increases in gate delays. To further evaluate the performance, we have resorted to ASIC implementation and simulation.

Table 2: Gate count Comparison

| GROUP | REGULAR | MODIFIED |
|---|---|---|
| 2 | 57 | 43 |
| 3 | 84 | 61 |
| 4 | 117 | 84 |
| 5 | 147 | 107 |

## V. Result

The design proposed in this paper has been developed using Verilog- HDL and synthesized in Cadence RTL compiler using typical libraries of TSMC 0.18 um technology. The synthesized Verilog netlist and their respective design constraints file (SDC) are imported to Cadence SoC Encounter and are used to generate automated layout from standard cells and placement and routing. Parasitic extraction is performed using Encounter's Native RC extraction tool and the extracted parasitic RC (SPEF format) is back annotated to Common Timing Engine in Encounter platform for static timing analysis. For each word size of the adder, the same value changed dump (VCD) file is generated for all possible input conditions and imported the same to Cadence Encounter Power Analysis to perform the power simulations. The similar design flow is followed for both the regular and modified SQRT CSLA.

Table below exhibits the simulation results of both the CSLA structures in terms of delay, area and power. The area indicates the total cell area of the design and the total power is sum of the leakage power, internal power and switching power. The percentage reduction in the cell area, total power, power-delay product and the area–delay product as function of the bit size are shown in Fig. (a). Also plotted is the percentage delay overhead in Fig. (b). It is clear that the area of the 8-, 16-, 32-, and 64-b proposed SQRT CSLA is reduced by 9.7%, 15%, 16.7%, and 17.4%, respectively. The total power consumed shows a similar trend of increasing reduction in power consumption 7.6%, 10.56%, 13.63%, and 15.46 % with the bit size. Interestingly, the delay overhead also exhibits a similarly decreasing trend with bit size. The delay overhead for the 8, 16, and 32-b is 14%, 9.8%, and 6.7% respectively, whereas for the 64-b it reduces to only 3.76%. The power–delay product of the proposed 8-b is higher than that of the regular SQRT CSLA by 5.2% and the area-delay product is lower by 2.9%. However, the power-delay product of the proposed 16-b SQRT CSLA reduces by 1.76% and for the 32-b and 64-b by as much as 8.18%, and 12.28% respectively. Similarly the area-delay product of the proposed design for 16-, 32-, and 64-b is also reduced by 6.7%, 11%, and 14.4% respectively.

Table 3: Comparison of The Regular And Modified SQRT CSLA

| Word Size (bit) | Adder | Delay (ns) | Area $(\mu m)^2$ | Leakage Power $(\mu m)^2$ | Switching Power $(\mu m)^2$ | Total Power $(\mu m)^2$ | Power-Delay Product $(10^{-15})$ | Area-Delay Product $(10^{-21})$ |
|---|---|---|---|---|---|---|---|---|
| 8 | Regular CSLA | 1.719 | 991 | 0.004 | 101.9 | 203.9 | 350.5 | 1703.5 |
| 8 | Modified CSLA | 1.958 | 895 | 0.006 | 94.2 | 188.4 | 368.8 | 1752.4 |
| 16 | Regular CSLA | 2.775 | 2272 | 0.017 | 263.7 | 527.5 | 1463.8 | 6304.8 |
| 16 | Modified CSLA | 3.048 | 1929 | 0.013 | 235.9 | 471.8 | 1438.0 | 5879.6 |
| 32 | Regular CSLA | 5.137 | 4783 | 0.036 | 563.6 | 1127.3 | 5790.9 | 24570.2 |
| 32 | Modified CSLA | 5.482 | 3985 | 0.027 | 484.9 | 969.9 | 5316.9 | 21845.7 |
| 64 | Regular CSLA | 9.174 | 9916 | 0.075 | 1212.4 | 2425.0 | 22246.9 | 90969.3 |
| 64 | Modified CSLA | 9.519 | 8183 | 0.057 | 1025.0 | 2050.1 | 19514.9 | 77893.9 |

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

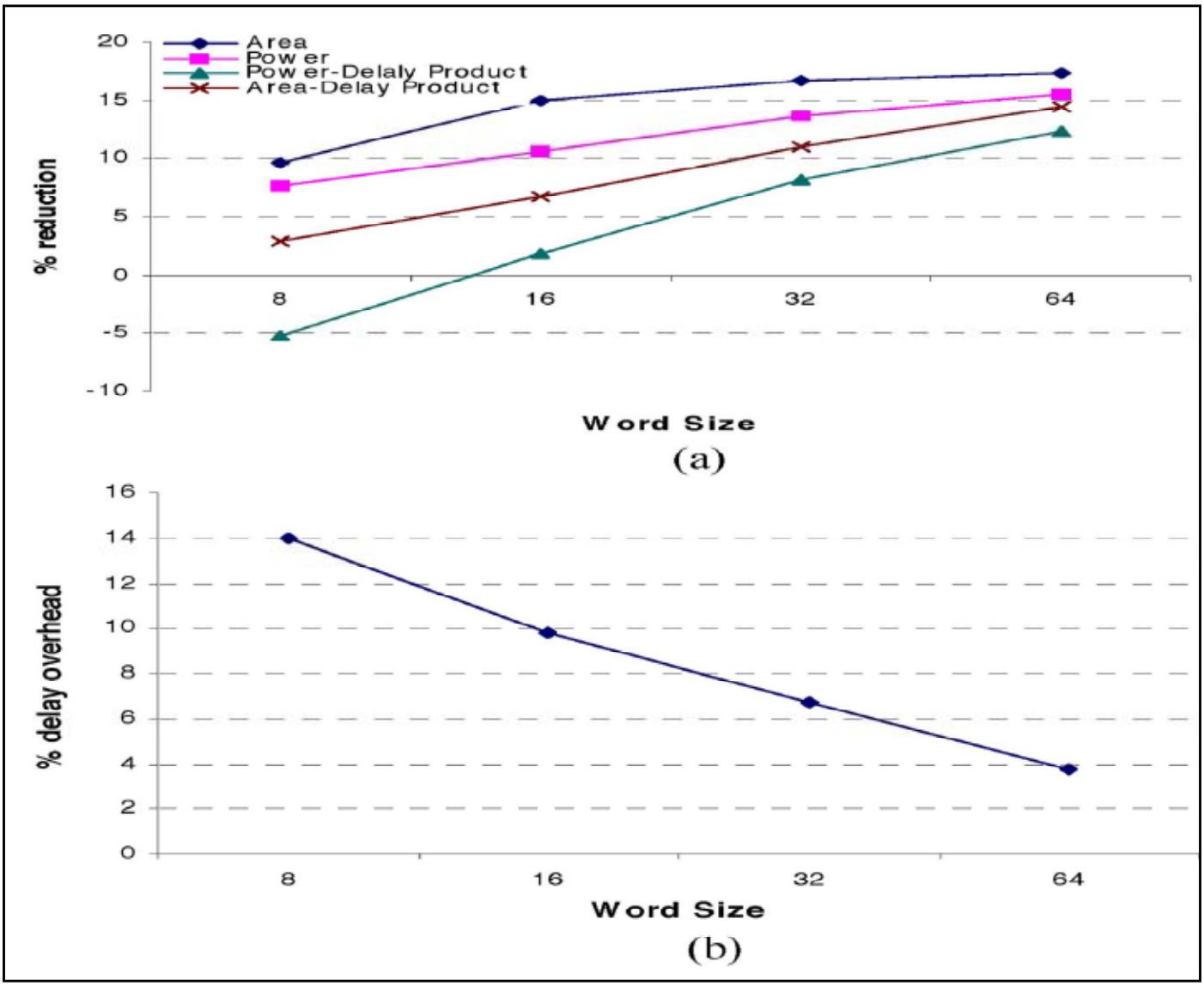



Fig 5: (a) Percentage reduction in the cell area, total power, power–delay product, and area–delay product. (b) Percentage of delay overhead.

## VI. Conclusion

A simple approach is proposed in this paper to reduce the area and power of SQRT CSLA architecture. The reduced number of gates of this work offers the great advantage in the reduction of area and also the total power. The compared results show that the modified SQRT CSLA has a slightly larger delay (only 3.76%), but the area and power of the 64-b modified SQRT CSLA are significantly reduced by 17.4% and 15.4% respectively. The power-delay product and also the area-delay product of the proposed design show a decrease for 16-, 32-, and 64-b sizes which indicates the success of the method and not a mere tradeoff of delay for power and area. The modified CSLA architecture is therefore, low area, low power, simple and efficient for VLSI hardware implementation. It would be interesting to test the design of the modified 128-b SQRT CSLA.

## References

[1] *O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput., pp. 340–344, 1962.*

[2] *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.*

[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. 37, no. 10, pp. 614–615, May 2001.*

[5] *J. M. Rabaey, Digtal 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 lowpower applications," in Proc. IEEE Int. Symp. Circuits Syst., 2005, vol. 4, pp. 4082–4085.*

[7] *Cadence, "Encounter user guide," Version 6.2.4, March 2008.*