

## Implementation of Low Power and Area Efficient Carry Select Adder

<sup>1</sup>Geeta A Sannakki , <sup>2</sup>Madhu. B. C

<sup>1</sup>PG Student, M-Tech , <sup>2</sup>Assistant Professor

<sup>1,2</sup>VLSI Design and Embedded Systems, Shridevi Institute of Engineering and Technology  
 Tumkur, India

**ABSTRACT:** Carry Select Adder (CSLA) is one of the fastest adders used in many data 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 delay of the CSLA. Based on this modification 8-, 16-, 32-, and 64-b non-uniform CSLA architecture have been developed and compared with the uniform CSLA architecture. The proposed BEC design has reduced area and delay as compared with the non-uniform CSLA with decrease in the delay. This work evaluates the performance of the proposed designs in terms of delay, area, power, and their products by hand with logical effort and through custom design. The result analysis shows that the proposed BEC CSLA structure is better than the non-uniform CSLA.

### 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 non-uniform 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.

#### Existing system:

**Basic structure of non-uniform csLA:** The basic non-uniform Carry select adder has a dual ripple carry adder with 2: 1 multiplexer the main disadvantage of non-uniform CSLA is the large area due to the multiple pairs of ripple carry adder. The non-uniform 16-bit Carry select adder is shown in Fig. (1). It is divided into five groups with different bit size RCA. From the structure of Non-uniform CSLA, there is scope for reducing area and power consumption. The carry out calculated from the last stage i.e. least significant bit stage is used to select the actual calculated values of the output carry and sum. The selection is done by using a multiplexer.

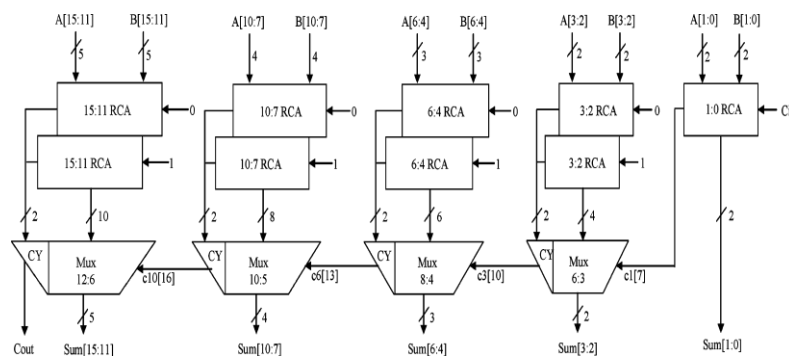


Fig 2.1 Non-uniform 16-b CSLA

**Delay and area evaluation methodology of the basic adder blocks:** The AND, OR, and Inverter (AOI) implementation of an XOR gate is shown in Fig. 2.2, 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 and listed in Table I.

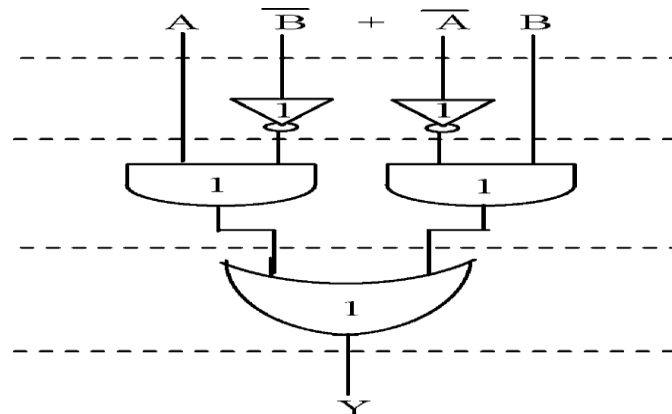


Fig 2.2 Delay and Area evaluation of an XOR gate

Table I  
Delay and area count of the basic blocks of cscla

Adder blocks	Delay	Area
XOR	3	5
2:1 Mux	3	4
Half adder	3	6
Full adder	6	13

Delay and area evaluation methodology of non-uniform 16-bit cscla

Internal structure of the all the groups of non-uniform 16-bit CSLA is shown Fig.2.1. By manually counting the number of gates used for group 3 is 87 (full adder, half adder, and multiplexer) and 13ns delay. One input to the mux goes from the RCA with Cin=0 and other input from the RCA with Cin=1.

Similarly, the estimated maximum delay and area of the other groups in the non-uniform SQRD CSLA are evaluated and listed in Table II

TABLE II  
DELAY AND AREA COUNT OF NON-UNIFORM CSLA GROUPS

Group	Delay	Area
Group2	11	57
Group3	13	87
Group4	16	117
Group5	19	147

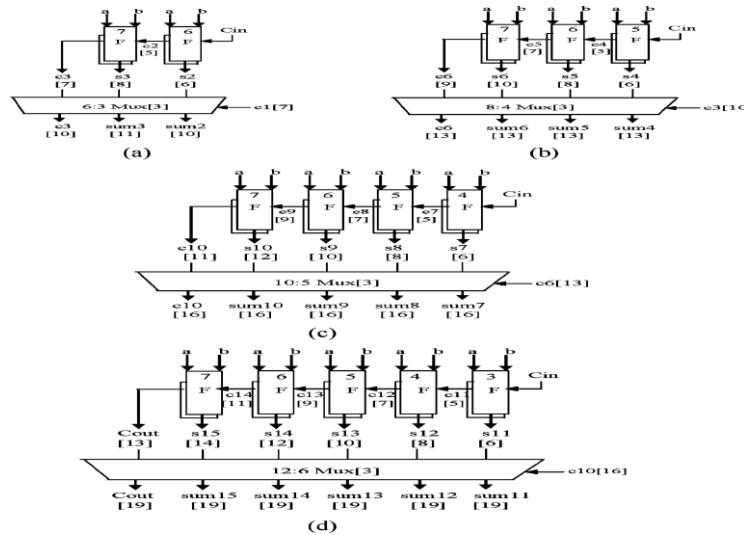


Fig 2.3 Delay and area evaluation of non-uniform SQRT CSLA: (a) group 2, (b) group 3, (c) group 4, and (d) group 5. F is a Full Adder.

The structure of the 16-b non-uniform CSLA is shown in Fig.2.1 It has five groups of different size RCA. The delay and area evaluation of each group are shown in Fig.2.3, in which the numerals within [ ] specify the delay values, e.g., sum2 requires 10 gate delays. The steps leading to the evaluation are as follows.

- [1] The group2 [see Fig. 2.3(a)] has two sets of 2-b RCA. Based on the consideration of
- [2] delay values of Table I, the arrival time of selection input  $c1[t=7]$  of 6:3 mux is earlier than  $s3[t=8]$  and later than  $s2[t=6]$ . Thus,  $sum3[t=11]$  is summation of  $s3$  and  $mux[t=3]$  and  $sum2[t=10]$  is the summation of  $c1$  and  $mux$ .
- [3] Except for group2, the arrival time of mux selection input is always greater than the
- [4] Arrival time of data outputs from the RCA's. Thus, the delay of group3 to group5 is determined, respectively as follows:
  - a.  $\{c6, sum[6:4]\} = c3[t=10] + mux$
  - b.  $\{c10, sum[10:7]\} = c6[t=13] + mux$
  - c.  $\{cout, sum[15:11]\} = c10[t=16] + mux$ .
- [5] The one set of 2-b RCA in group2 has 2 FA for  $Cin = 1$  and the other set has 1 FA and 1
- [6] HA for  $Cin = 0$ . Based on the area count of Table I, the total number of gate counts in group2 is determined as follows:
  - a. Gate count = 57 (FA + HA + Mux)
  - b. FA = 39(3 \* 13)
  - c. HA = 6(1 \* 6)
  - d. Mux = 12(3 \* 4).

[7] Similarly, the estimated maximum delay and area of the other groups in the non-uniform [8] CSLA are evaluated and listed in Table III.

Problems in existing system

- [1] The problem in CSLA design is the number of full adders is increased then the circuit
- [2] Complexity also increases.
- [3] The number of full adder cells are more thereby power consumption of the design also
- [4] increases.
- [5] Number of full adder cells doubles the area of the design also increased.

Binary to excess-1 code  
Solution to the problem:

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 power consumption of the non-uniform CSLA. To replace the n-bit RCA, an n+1-bit BEC is required. A structure and the function table of a 4-bit BEC are shown in Fig.3.1 and Table III, respectively.

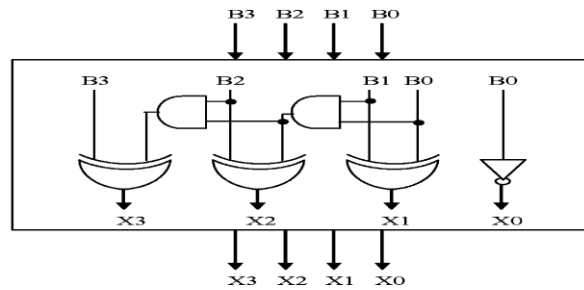


Fig 3.1 4-bit BEC

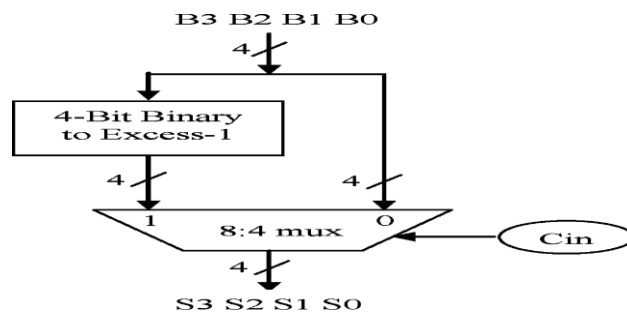


Fig 3.2 4-bit BEC with 8:4 mux

Fig. 3.2 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 its 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)

$$\begin{aligned}
 X0 &= \sim B0 \\
 X1 &= B0 \wedge B1 \\
 X2 &= B2 \wedge (B0 \& B1) \\
 X3 &= B3 \wedge (B0 \& B1 \& B2).
 \end{aligned}$$

TABLE III  
FUNCTION TABLE OF THE 4-b BEC

B[3:0]	X[3:0]
0000	0001
0001	0010
⋮	⋮
1110	1111
1111	0000

Delay and area evaluation methodology of modified 16-b non-uniform csla.

The structure of the proposed 16-b non-uniform CSLA using BEC for RCA with  $C_{in} = 1$  to optimize the area and power is shown in Fig. 3.3. We again split the structure into five groups. The delay and area estimation of each group are shown in Fig. 3.4

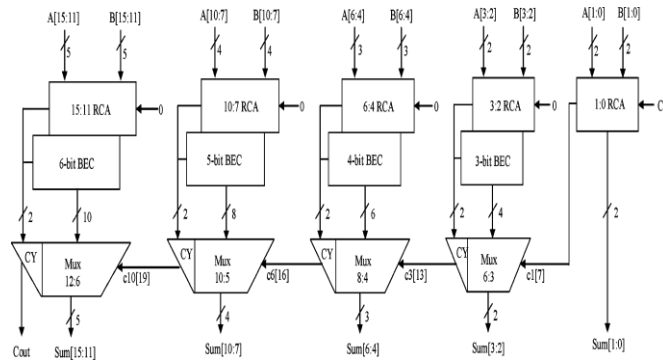


Fig 3.3. Modified 16-b non-uniform CSLA. The parallel RCA with  $C_{in} = 1$  is replaced with BEC.

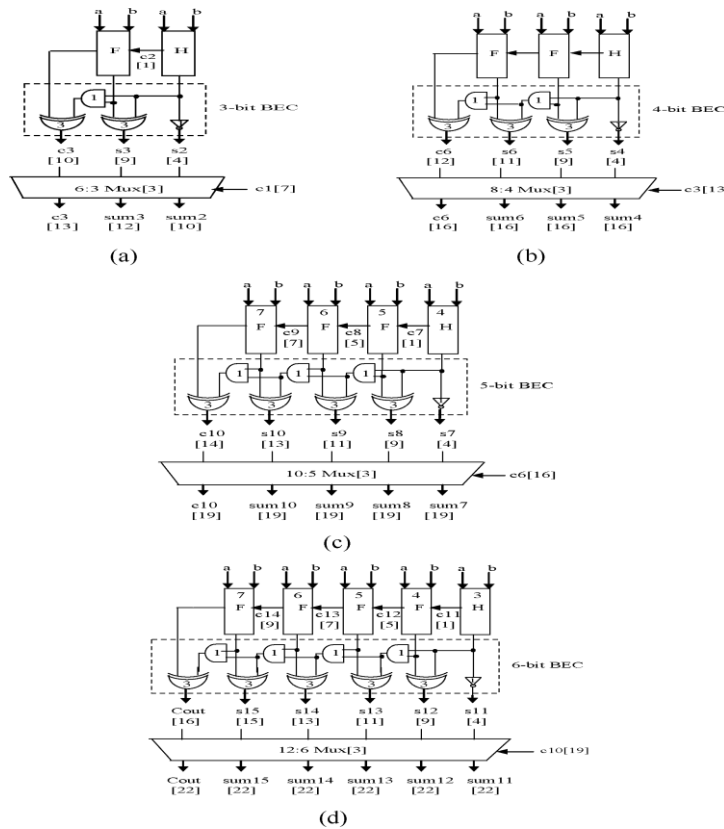


Fig. 3.4 Delay and area evaluation of modified non-uniform CSLA: (a) group2, (b)group3, (c) group4, and (d) group5. H is a Half Adder.

The steps leading to the evaluation are given here

- [1] The group2 [see Fig. 3.4(a)] has one 2-b RCA which has 1 FA and 1 HA for  $C_{in} = 0$ .
- [2] 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 of Table I, 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  $s3$  and mux and partial  $c3$  (input to mux) and mux, respectively. The  $sum2$  depends on  $c1$  and mux.

[3] 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.

[4] The area count of group2 is determined as follows:

[5] Gate count = 43 (FA + HA +Mux + BEC)

[6] FA = 13(1 \* 13)

[7] HA = 6(1 \* 6)

[8] AND = 1

[9] NOT = 1

[10] XOR = 10(2 \* 5)

[11] Mux = 12(3 \* 4).

Similarly, the estimated maximum delay and area of the other groups of the modified non-uniform CSLA are evaluated and listed in Table IV.

### Synthesis

This Chapter deals with the Synthesis and FPGA implementation of the Arithmetic module. The FPGA used is Xilinx Spartan3E (Family), XC3S500 (Device), FG320 (Package), -4 (Speed Grade)

Here, the RTL view its description, the device used and its Hardware utilization summary is given for each module, starting from the most basic component.

### RCA 2-BIT BLOCK

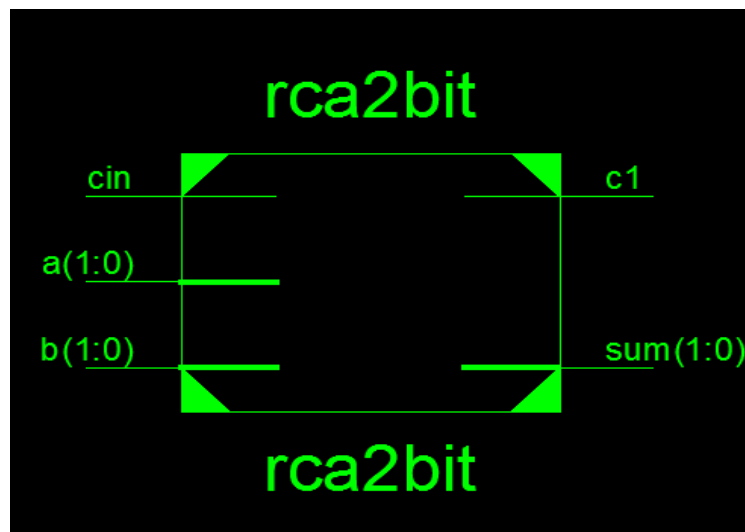


Fig 4.1 Black box view of RCA-2bit

### Description

A	input data 2-bit
B	input data 2-bit
Cin	input data 1-bit
C1	output data carry 1-bit
Sum	output data 2-bit

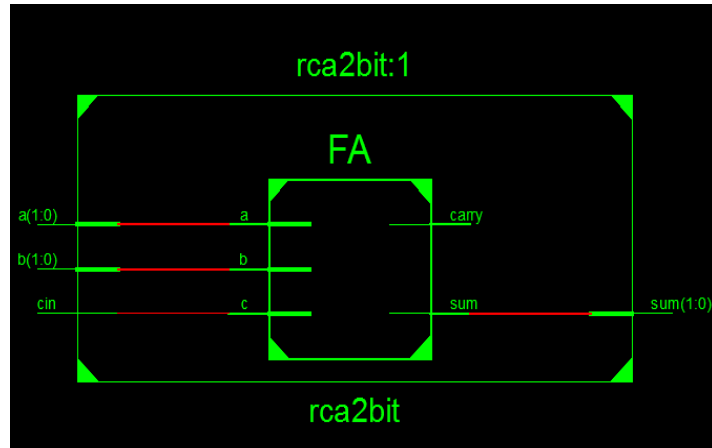


Fig. 4.2 RTL view of RCA 2-bit

CSLA 2-bit BLOCK

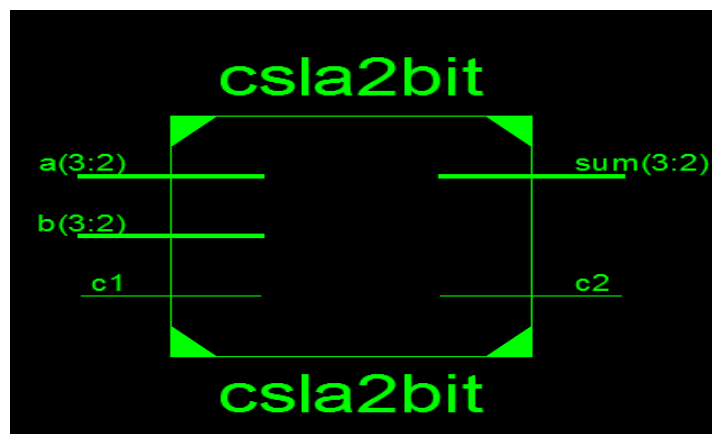


Fig 4.3 Black box view of CSLA-2bit

Description

- A input data 2-bit
- B input data 2-bit
- C1 input carry 1-bit
- Sum output data 2-bit
- C2 output carry 1-bit

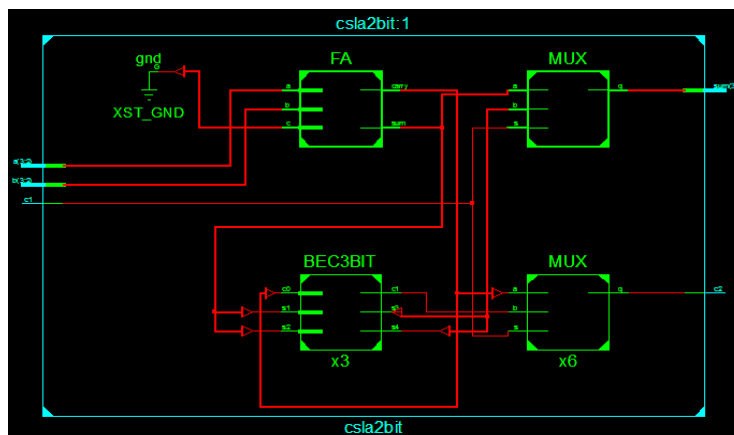


Fig. 4.4 RTL view of CSLA 2-bit

CSLA 3-bit Block

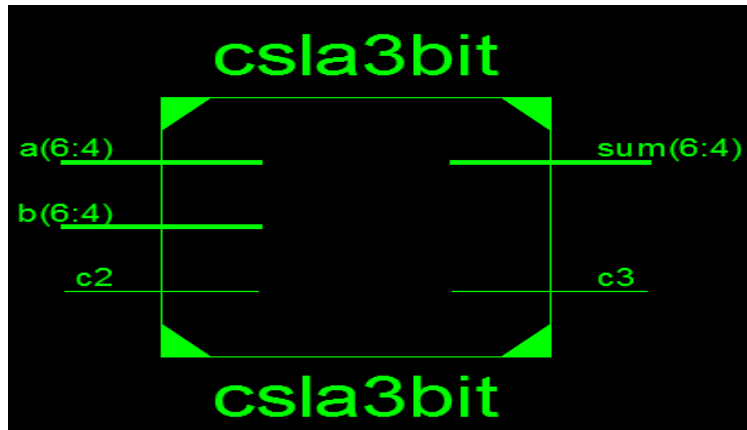


Fig 4.5 Black box view of CSLA-3bit

Description

A	input data 3-bit
B	input data 3-bit
C2	input carry 1-bit
Sum	output data 3-bit
C3	output carry 1-bit

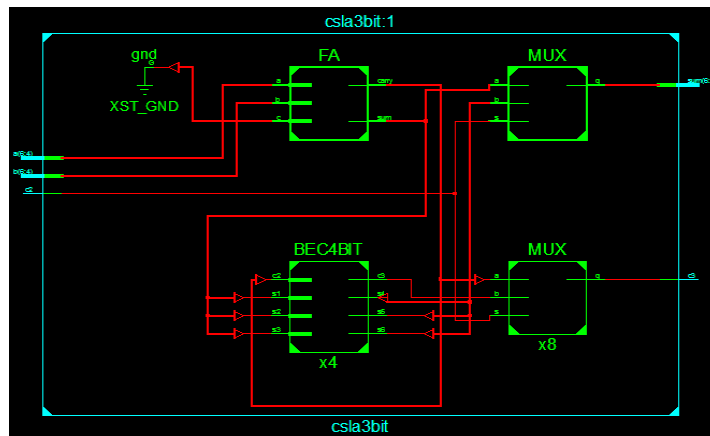


Fig. 4.6 RTL view of CSLA 3-bit  
CSLA 4-bit Block

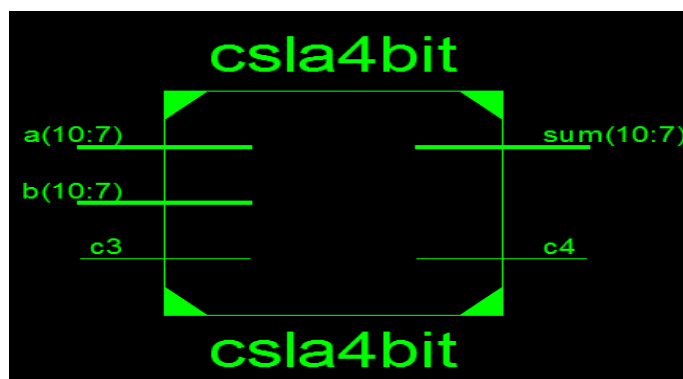


Fig 4.7 Black box view of CSLA-4bit



Description

A	input data 4-bit
B	input data 4-bit
C3	input carry 1-bit
Sum	output data 4-bit
C4	output carry 1-bit

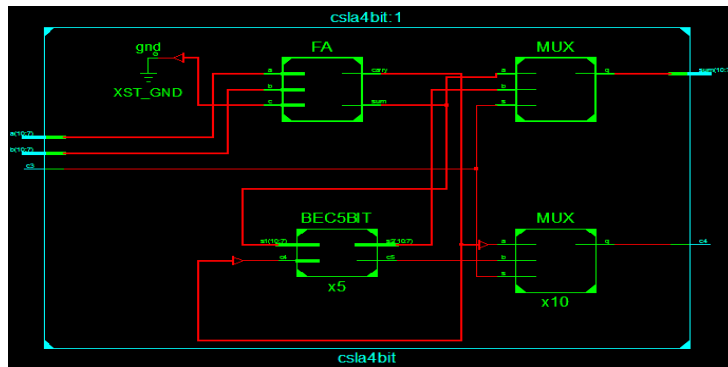


Fig. 4.8 RTL view of CSLA 4-bit  
CSLA 5-bit Block

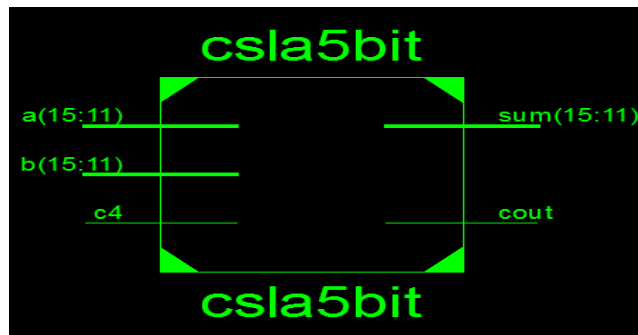


Fig 4.9 Black box view of CSLA-5bit

Description

A	input data 5-bit
B	input data 5-bit
C4	input carry 1-bit
Sum	output data 5-bit
Cout	output carry 1-bit

Fig. 4.10 RTL view of RCA 5-bit

BEC 3-bit block



Fig 4.11 Black box view of BEC-3bit

Description  
C0, S1, S2  
C1, S3, S4

input data 1-bit  
output data 1-bit

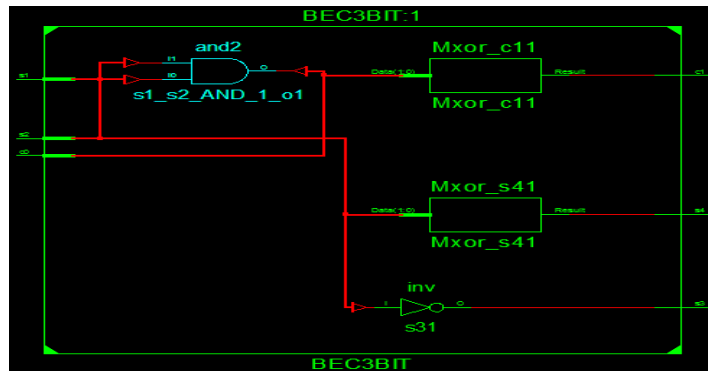


Fig. 4.12 RTL view of BEC 3-bit  
BEC 4-bit block



Fig 4.13 Black box view of BEC-4bit

Description  
C2, S1, S2, S3  
C3, S4, S5, S6

input data 1-bit  
output data 1-bit

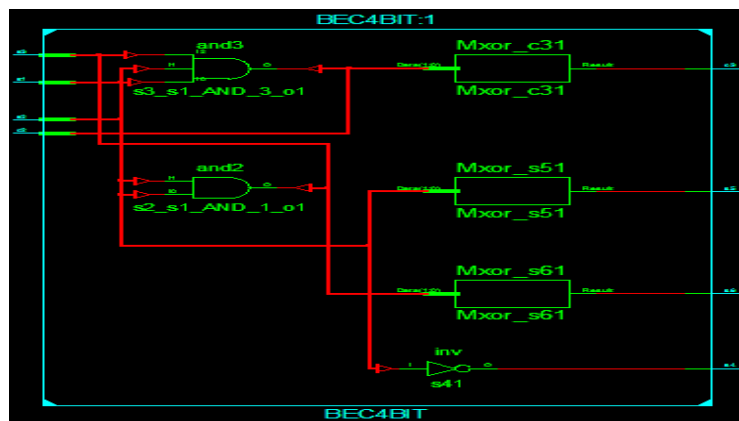


Fig. 4.14 RTL view of BEC 4-bit  
BEC 5-bit block

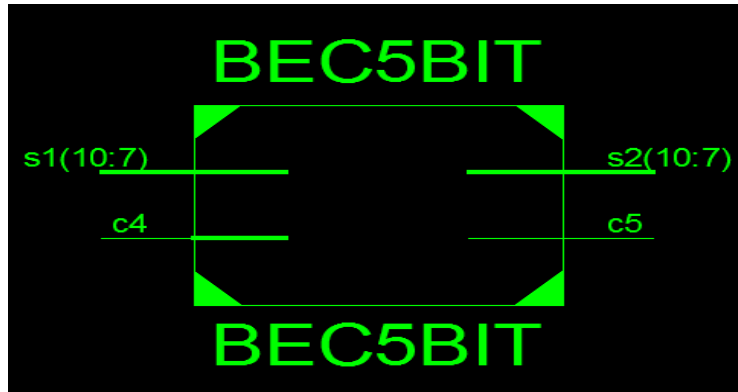


Fig 4.15 Black box view of BEC-5bit

Description

- S1 input data 3-bit
- C4 input data 1-bit
- S2 output data 3-bit
- C5 output data 1-bit

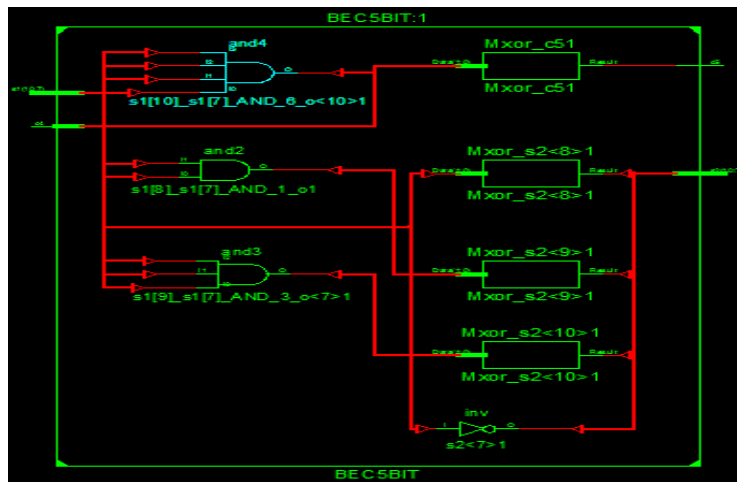


Fig. 4.16 RTL view of BEC 5-bit  
BEC 6-bit block

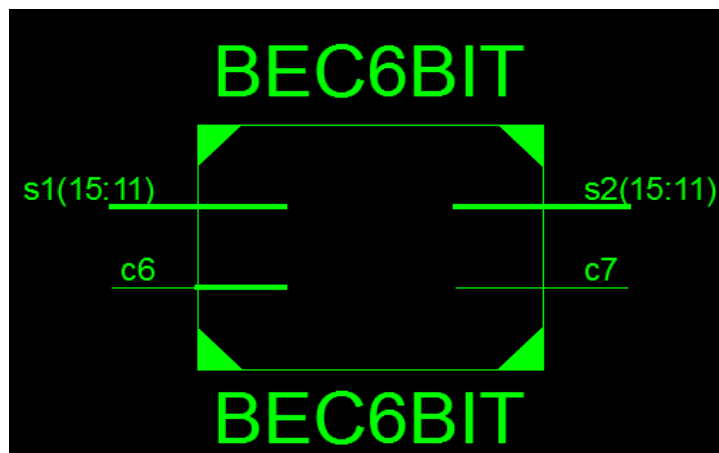


Fig 4.17 Black box view of BEC-6 bit

Description

- S1 input data 3-bit
- C6 input data 1-bit
- S2 output data 3-bit
- C7 output data 1-bit

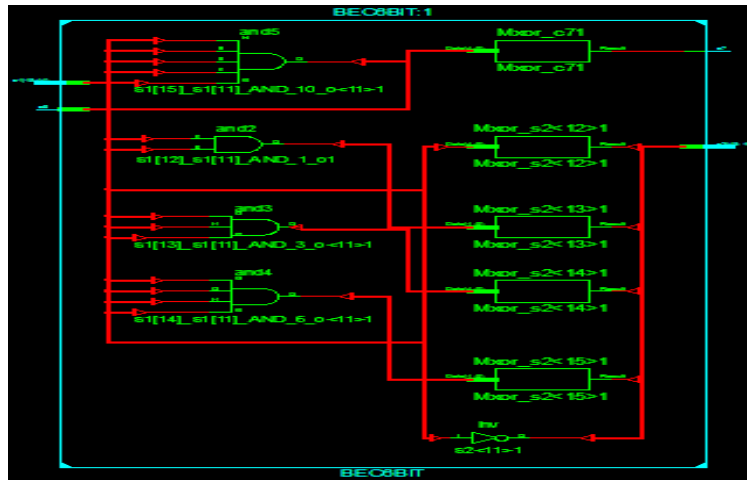


Fig. 4.18 RTL view of BEC 6-bit

Simulation and Result

Simulation of modified 16-bit bec csla:

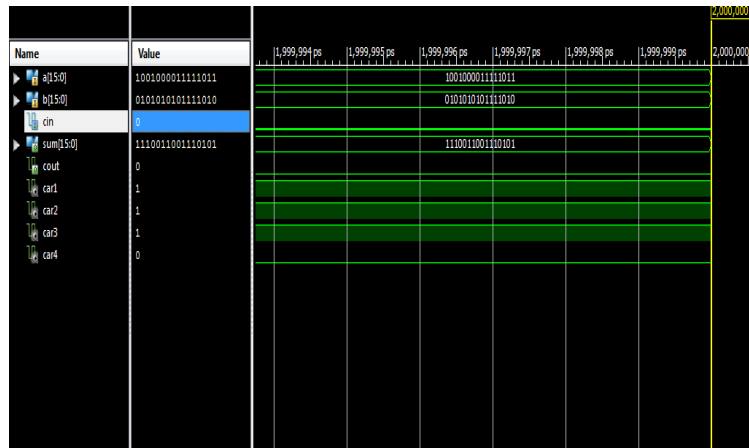


Fig 4.1 Simulation waveform of modified 16-bit BEC CSLA

Description

- A input data 16-bit
- B input data 16-bit
- Cin input data 1-bit
- Sum output data 16-bit
- Car1 group1 carry 1-bit
- Car2 group2 carry 1-bit
- Car3 group3 carry 1-bit
- Car4 group4 carry 1-bit

## Comparison of the non-uniform 16-bit CSLA and modified BEC 16-bit CSLA

Word size	Adder	Delay(ns)	Number of slice LUT's
16-bit	Non-uniform CSLA	5.424	34
16-bit	Modified BEC CSLA	4.441	32

## Advantages

- Low power consumption
- Less area
- More speed compared to non-uniform CSLA
- Less complexity

## Applications

- Arithmetic logic units
- High Speed multipliers
- Advanced microprocessor design
- Digital signal process

**II. CONCLUSION:**

A simple approach is proposed in this paper to reduce the area and power of non-uniform CSLA architecture. The reduced number of gates offers the great advantage in the reduction of area and also the total power. The compared results show that the modified non-uniform CSLA (BEC) has lesser delay. The power-delay product and also the area-delay product of the proposed design show a decrease for 16-, 32-, and 64-bit 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 delay, simple and efficient for VLSI hardware implementation. It would be interesting to test the design of the modified 128-bit non-uniform CSLA.

**REFERENCES:**

- [1] Ramkumar B and Harish M Kittur,( 2012)" Low-Power and Area-Efficient Carry Select Adder", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 2
- [2] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput. , pp. 340–344, 1962.
- [3] 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
- [4] T. Y. Chang and M. J. Hsiao, "Carry-select adder using single ripple-carry adder," Electronics Letters, vol. 34, no. 22, pp. 2101 – 2103, Oct. 1998
- [5] Morinaka, H. , Makino, H. , Nakase, Y. et. al, "A 64 bit Carry Look-ahead CMOS adder using Modified Carry Select". Cz/stoin Integrated Circuit Conference, 1995, pages 585-588
- [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.