



DESIGN OF A LOW-POWER 16-BIT CSLA WITH BINARY TO EXCESS-1 CONVERTERS

K Prakash Rao¹, A Uday Kumar²

¹Assistant Professor, ²Associate Professor, Department of ECE,
SVCET, Etcherla, Srikakulam, Andhra Pradesh, (India)

ABSTRACT

Carry Select Adder (CSLA) is one of the fastest adders used in many data processing processors to perform fast arithmetic functions. CSLA is an application specific integrated circuit (ASIC) developed with the modification of a regular square root (SQRT) CSLA architecture. The main objective is to design an adder that has reduced area and power as compared with the regular SQRT CSLA to evaluate the performance of the proposed design in terms of delay, area, power with logical effort and through custom design. 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 selected by the multiplexers. The basic idea of this thesis is to use Binary to Excess-1 Converters (BEC) instead of RCA with in the regular CSLA to achieve lower area and power consumption. In spite of the delay, CSLA is more advantageous that require low power and area consumption.

The design proposed has been developed using Verilog-HDL and synthesized in Cadence RTL compiler using typical libraries of TSMC 0.18 um technology.

Keywords: Binary to Excess-1 Converters (BEC), Carry Select Adder (CSLA), square root (SQRT) CSLA architecture

I. INTRODUCTION:

In recent years, power consumption in CMOS circuit has become major design consideration for very large scale integration (VLSI) system. In VLSI systems, power consumption includes dynamic power and static power consumption. Major portion of power consumption in any VLSI system consists of dynamic power consumption. Moreover, the scaling trends of MOSFETs lead to the development towards Nano-scale processes as driven by Moore law. With these developments the leakage currents are also increasing and the static power component of power dissipation is playing a vital role in the total power consumption. Static power is dissipated mainly due to the source and drain leakage currents and controlling the bulk terminal of CMOS device offers improved performance in the terms of power dissipation and delay [1].

The objective of this thesis is to design a high-speed adder along with low power and smaller area as a prime consideration circuit level design optimizations are required for reducing the power consumption in CMOS circuits. As stated above the PDP exhibited by the full adder would affect the systems' overall performance. Thus taking this fact into consideration, the design of a full adder having low power consumption and low propagation delay results of great interest for the implementation of modern digital systems. A regular Square

Root Carry Select Adder (SQRT CSLA) is considered and a modified Carry Select Adder is developed with an inspiration to improve the power consumption of full adder with reduced transistor counts. The adder is developed using Verilog HDL and synthesized in Cadence RTL compiler using typical libraries of 180 nm technology from Taiwan Semiconductor Manufacturing Company (TSMC). The basic idea of this work is to use Binary to Excess-1 Converter instead of Ripple Carry Adders with $C_{in}=1$ in the regular CSLA to achieve low area and power consumption

II. BASIC ADDER BLOCKS

2.1 Carry Select Adder

The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known [3]. The $O\sqrt{n}$ delay is derived from uniform sizing, where the ideal number of full-adder elements per block is equal to the square root of the number of bits being added, since that will yield an equal number of MUX delays.

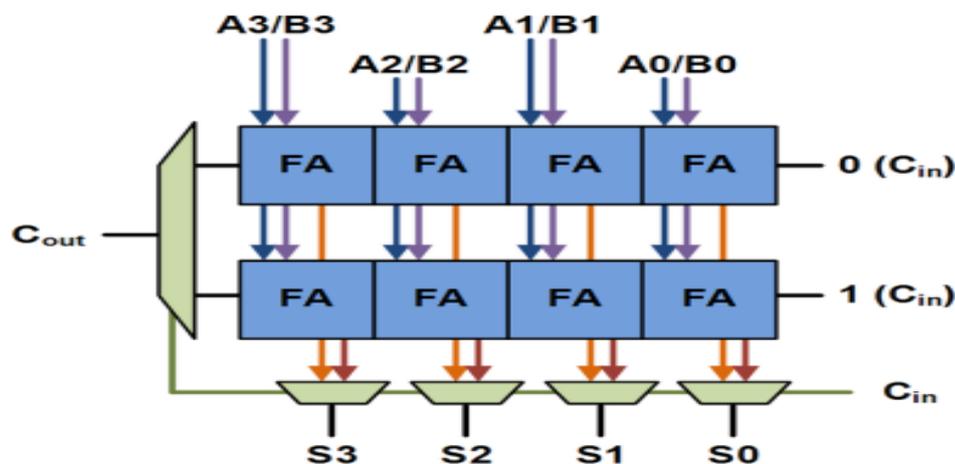


Figure 1. Carry Select Adder

CSLA can be implemented in different styles. The circuit styles are static CMOS, Domino, skewed CMOS, and DTSL. Figure 2 shows some examples of how to adjust the sizes of each circuit style, where W_p and W_n represent the optimal sizes of PMOS/NMOS of an inverter. Fig.2(a) shows a 2 input NAND gate, whose optimal channel widths are W_p and W_n . Fig.2(b) presents a 2 input Domino AND with a keeper transistor.

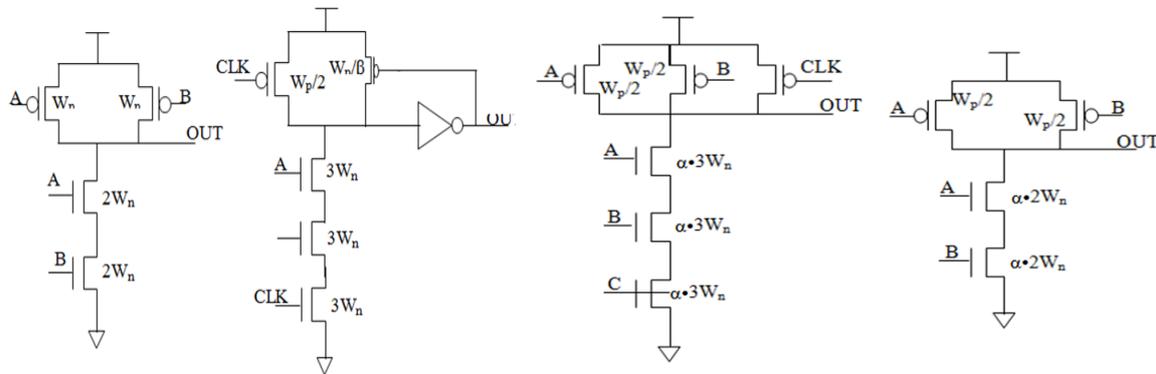


Figure 2. Width ratios of sizes of PMOS/NMOS for each circuit style. (a) Static CMOS (b) Domino circuit (c) Skewed CMOS with Clock (d) Skewed CMOS without Clock

Domino circuits are suitable for high performance because of the reduced junction capacitance at the output node due to the reduction in the number of PMOS transistors.

The dynamic node of Domino circuits can be floating during evaluation, which may cause noise problems in wide input OR gates. In order to prevent this, a larger keeper transistor may be required which can have an impact on performance. Hence, the size of the keeper transistor has to be selected carefully to optimize performance.

The circuit topology of a skewed logic is the same as that of the conventional static CMOS logic, however, the sizes of PMOS and NMOS transistors are decided based on the preferred transition. Fig.c and d show two different skewed 2 input NAND gates, in which transistors are sized to make fast high-to-low transitions. Skewed circuits have performance comparable to Domino circuits. However, skewed circuits require pre-charging (or selectively pre-charging) to each gate just like Domino circuit. However, this increases the clock load and hence, the power consumption. The other solution is to use DTSL (Dual Transition Skewed Logic) not require clock signal. Fig 3 shows an example of DTSL that achieves high performance by duplicating signal paths: one signal path is for fast rising transitions while other for fast falling transition. If the input of the first stage of the logic block toggles from high to low, faster data transition takes place through the top data path. On the other hand, if the input toggles from low to high, the data transits faster through the bottom path than through top path. The arrows represent the skew direction. The combiner detects earliest transition, latches it, and then transfers the data to the next stage.

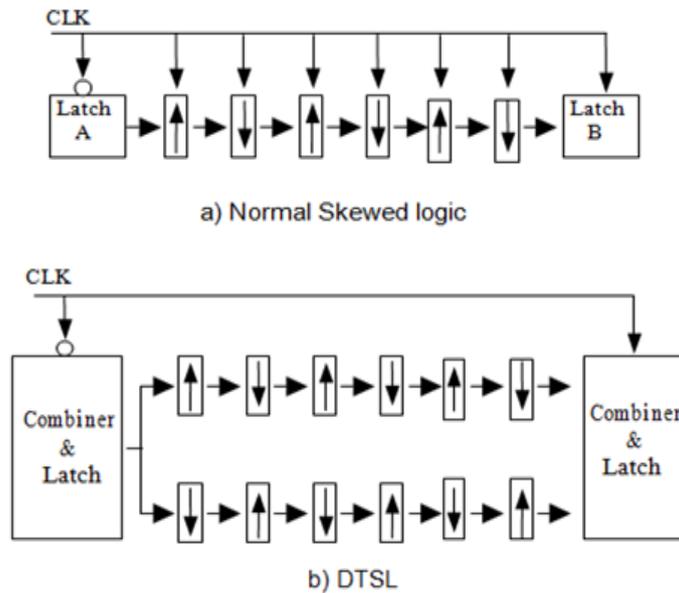


Figure 3. DTSL block structures.

2.2 Implementation of CSA using DTSL

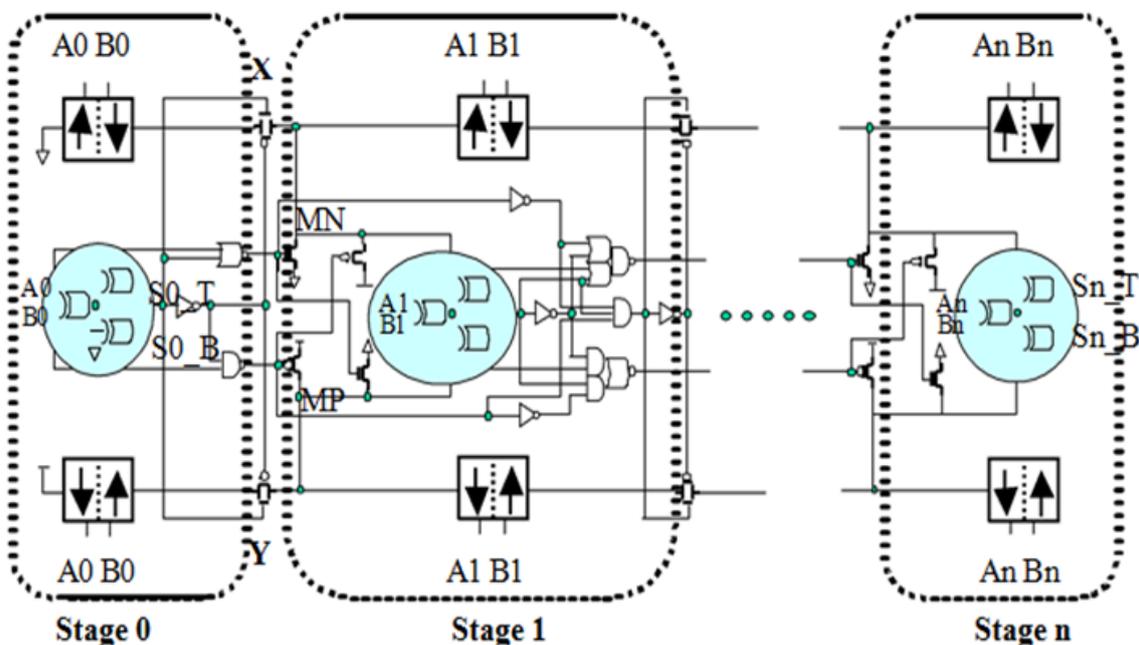


Figure 4. Block Diagram of CSLA using DTSL

DTSL has better noise immunity than the domino logic and comparatively high performance. In general the number of transistors used in DTSL is more than twice the number of the transistors used in Domino logic, however, the total size of the PMOS and NMOS transistor compared to static logic style is not high. In some application this number can be reduced.

Fig.4 shows the implementation of the carry propagation logic of CSA using DTSL. It consists of two data paths for carry propagation, logic for generating SUM, and control logic. Control logic consists of transmission gates (X,Y) between each carry propagation circuit on the data path, switching transistors (MN,

MP), and some static CMOS gates to control the transmission gates and switching transistors.

Fig. 5 shows the implementation of one stage of the carry propagation logic of CSA using DTSL. The arrows indicate the skew direction. The logic in the circle is for generating SUM. As shown in Fig. 4, the carry propagation logic of each block of CSA has two data paths: one has '0' as its CARRY input to the first stage and the other has '1' as its CARRY input.

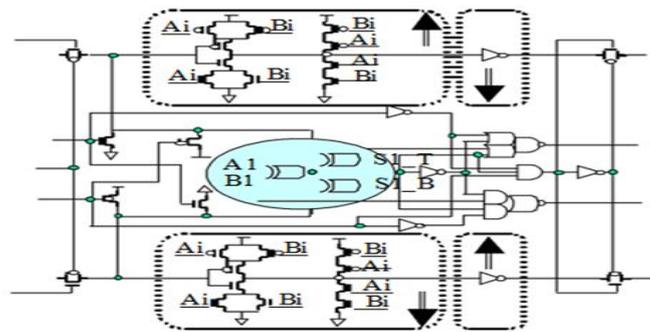


Figure 5. Block Diagram of one stage of CSA using DTSL

Therefore, the performance can be improved by properly skewing CMOS logic in the upper and lower logic blocks. The skew direction (high→low) on the top data path should be opposite to that (low→high) on the bottom.

III. PROPOSED ADDER DESIGN

Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. A regular CSLA circuit uses two RCAs through which the sum and carry outputs are generated. One of the RCA block propagates a carry '0' and the other RCA block propagates carry '1'. Multiplexers are provided at each RCA block to select the carry output [1]. This selection of carry outputs increases the speed of operation of the adder but require more area as it uses two RCAs and as a result it consumes more power. In order to reduce the area and power consumption a modified CSLA is proposed.

The modified CSLA proposed in this project is a simple and efficient gate-level modification to significantly reduce the area and power consumption. The main idea is to replace one of the RCA circuit with a Binary to Excess-1 Converter (BEC). The RCA module which propagates carry '1' is replaced by a BEC. The carry of RCA with carry '0' is passed through a BEC which is equivalent to the RCA with carry '1' and in turn reduces the area and power consumption [2]. 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 and layout in 0.18- μ m CMOS process technology. From the results and analysis it can be observed that the proposed CSLA structure is better than the regular SQR CSLA.

3.2 Binary to Excess-1 Converter (BEC)

Excess-3 binary coded decimal or Stibitz code, also called biased representation of Excess-N, is a complementary BCD cod and numeral system. It is a way to represent values with a balanced number of positive and negative numbers using a pre-specified number N as a biasing value. It is a non-weighted code. In

XS-3, numbers are represented as decimal digits, and each digit is represented by four bits as the digit value plus 3 (the "excess" amount):

1. The smallest binary number represents the smallest value. (i.e. 0 – Excess Value).
2. The greatest binary number represents the largest value. (i.e. 2^{N+1} – Excess Value – 1).

The primary advantage of XS-3 coding over non-biased coding is that a decimal number can be nines' complemented (for subtraction) as easily as a binary number can be ones' complemented (to invert all bits).

As stated earlier the main idea of this work is to use BEC instead of the RCA with in order to reduce the area and power consumption of the regular CSLA. To replace the n-bit RCA, an n+1 bit BEC is required. A structure and the function table of a 4-b BEC are shown in Fig. 6 and Table 1, respectively.

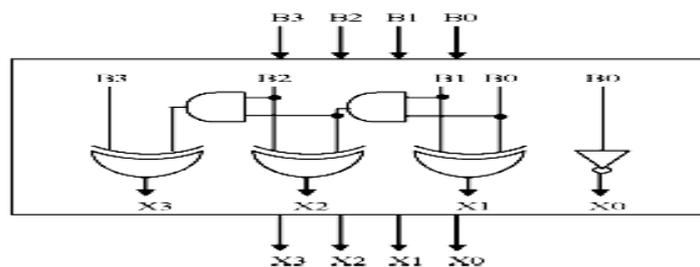


Figure.6 4-bit BEC

TABLE 1 Function table of 4-bit BEC

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

This figure 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).

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

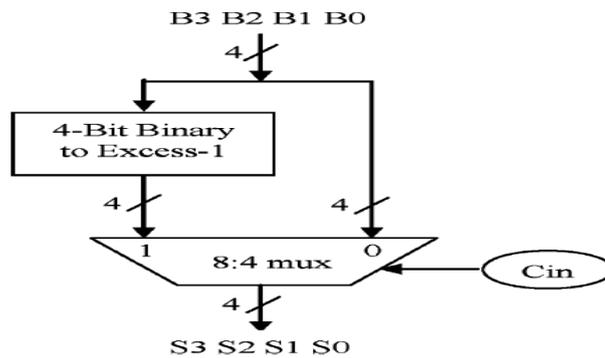


Figure.7 4-bit BEC with 8:4 MUX

3.3 Delay and Area Evaluation Methodology of 16-bit SQRT CSLA

The structure of the 8-b regular SQRT CSLA is shown. It has five groups of different size RCA [5]. The delay and area evaluation of each group are shown, 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 [Fig. 8(a)] has two sets of 2-b RCA. Based on the consideration of delay values of Table, the arrival time of selection input c1 [time (t)=7] of 6:3 mux is earlier than s3 [t=8] and later than s2 [t=6]. Thus, sum [t=11] is summation of s3 and mux [t=3] and sum2 [t=10] is summation of c1 and mux.

2) Except for group2, the arrival time of mux selection input is always greater than the arrival time of data outputs from the RCA's. Thus, the delay of group3 to group5 is determined, respectively as follows.

$$\{c6, \text{sum}[6:4]\} = c3[t=10] + \text{mux}$$

$$\{c10, \text{sum}[10:7]\} = c6[t=13] + \text{mux}$$

$$\{\text{cout}, \text{sum}[15:11]\} = c10[t=16] + \text{mux}$$

3) The one set of 2-b RCA in group2 has 2 FA for $C_{in}=1$ and the other set has 1 FA and 1 HA for $C_{in}=0$. Based on the area count of Table, the total number of gate counts in group2 is determined as follows.

$$\text{Gate count} = 57 (\text{FA} + \text{HA} + \text{Mux}), \text{FA} = 39(3 \times 13), \text{HA} = 6(1 \times 6), \text{Mux} = 12(3 \times 4)$$

4) Similarly, the estimated maximum delay and area of the other groups in the regular SQRT CSLA are evaluated and listed in Table 2.

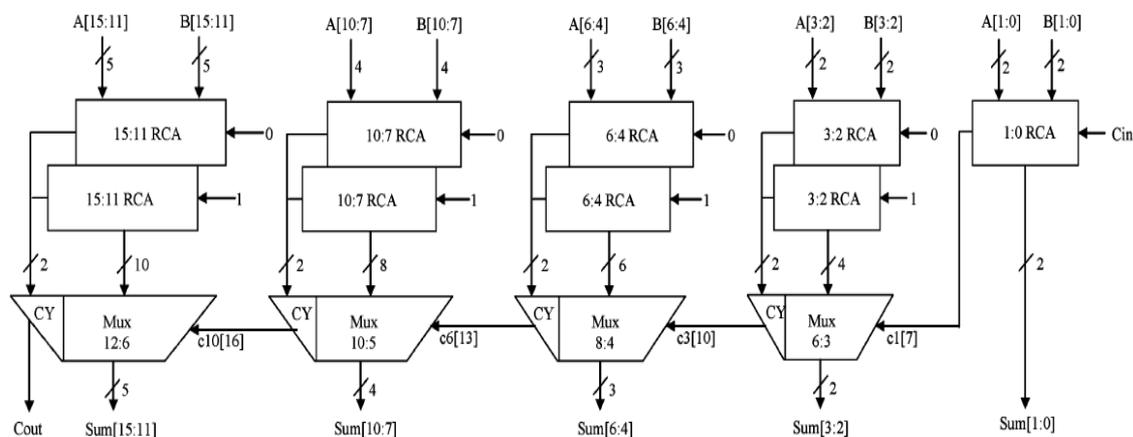


Figure.8 Regular 16-bit SQRT CSLA

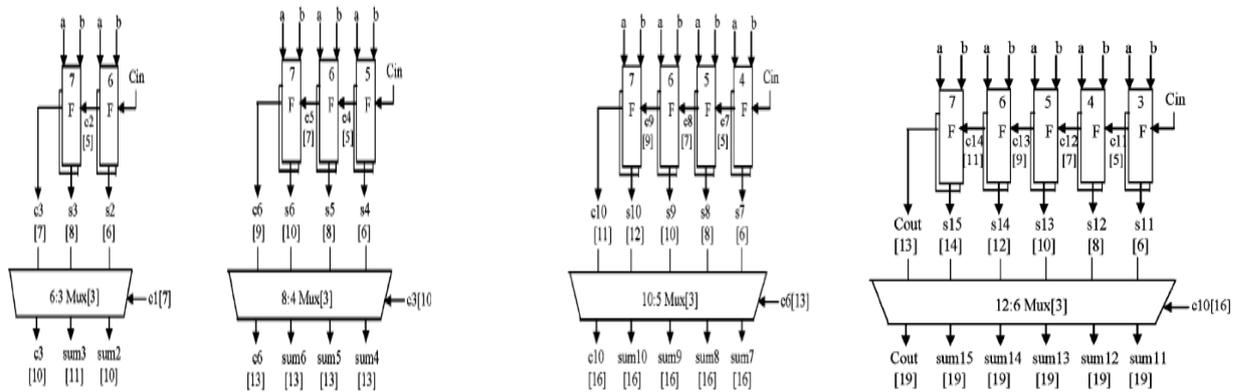


Figure.9 Delay and area evaluation of 16-b Sqrt CSLA: (a) group2, (b) group3, (c) group4, (d) group5.

TABLE 2. Delay and area count of regular Sqrt CSLA groups.

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

3.4 Delay and Area Evaluation Methodology of modified 16-bit CSLA

The structure of the proposed 16-bit Sqrt CSLA using BEC for RCA with $C_{in}=1$ to optimize the area and power is shown in Fig. 9. The structure is given into five groups. The delay and area estimation of each group are shown in Fig. 10. The steps leading to the evaluation are as follows.

- 1) The group2 [Fig. 10(a)] has one 2-bit RCA which has 1 FA and 1 HA for $C_{in}=0$. Instead of another 2-bit RCA with $C_{in}=1$ a 3-bit BEC is used which adds one to the output from 2-bit RCA[4]. Based on the consideration of delay values of Table, 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.
- 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.
- 3) The area count of group2 is determined as follows.

$$\text{Gate Count} = 43 (\text{FA} + \text{HA} + \text{Mux} + \text{BEC}), \text{FA} = 13(1 * 13), \text{HA} = 6(1 * 6)$$

$$\text{AND} = 1, \text{NOT} = 1, \text{XOR} = 10(2 * 5), \text{Mux} = 12(3 * 4).$$

- 4) Similarly, the estimated maximum delay and area of the other groups of the modified Sqrt CSLA are evaluated and listed in Table 3.

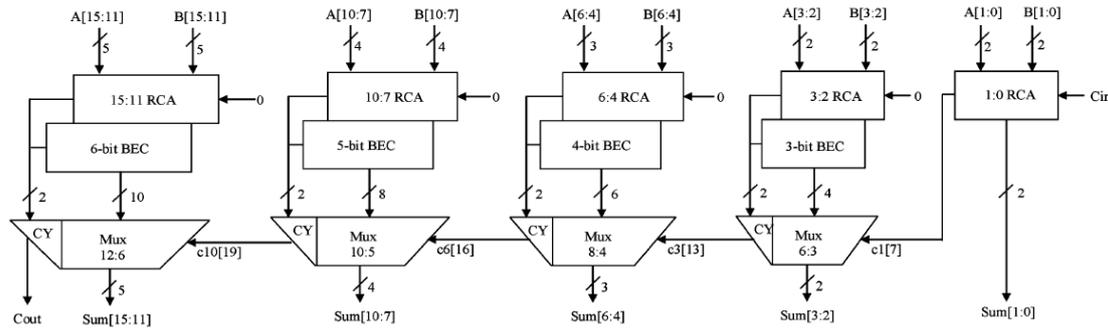


Figure.10 Modified 16-bit Sqrt CSLA

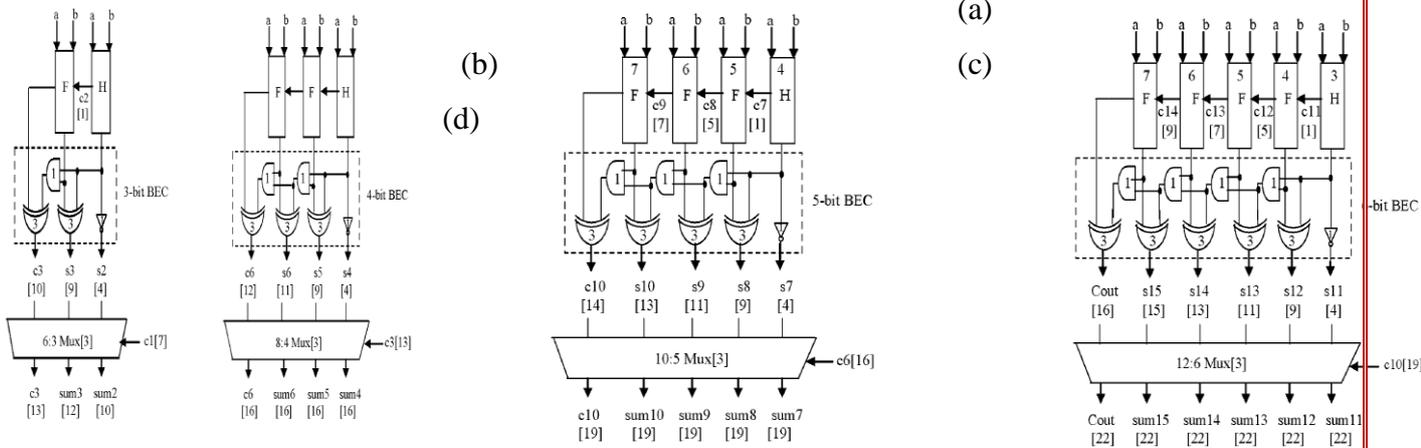


Figure.11 Delay and area evaluation of modified 16-bit Sqrt CSLA (a) group2 (b) group3 (c) group4 (d) group5.

TABLE 3. Delay and area count of modified Sqrt CSLA groups.

Group	Delay	Area
Group2	13	43
Group3	16	61
Group4	19	84
Group5	22	107

4.3 Schematic of Modified CSLA circuit

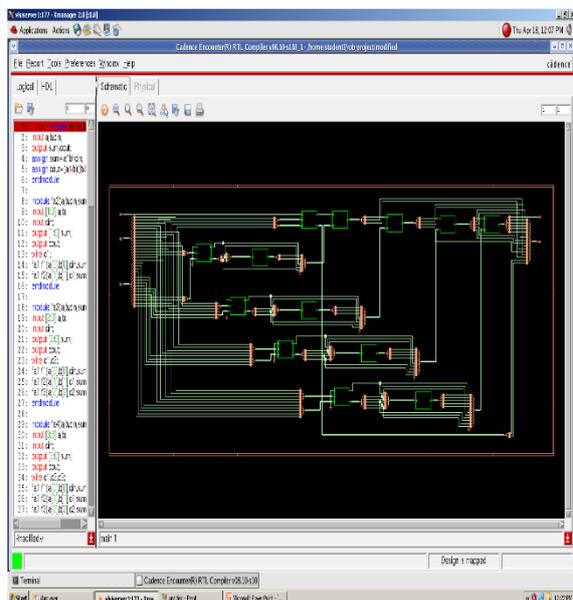


Figure.16 Modified CSLA schematic in Cadence Encounter RTL Compiler

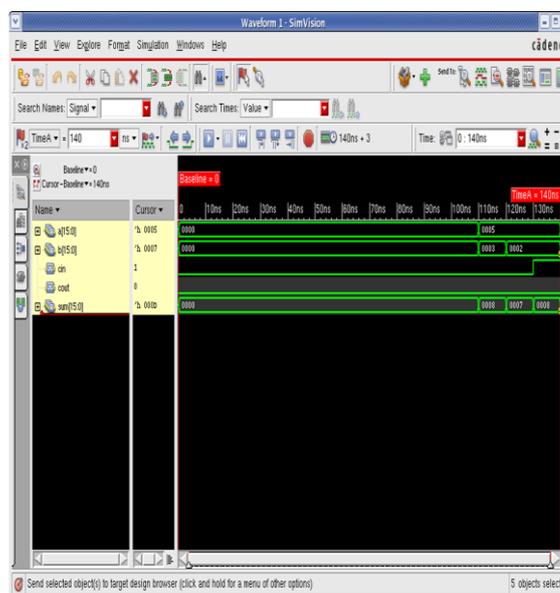


Figure.17 Modified CSLA output in Cadence

V. CONCLUSION

A simple approach is proposed in this project to reduce the area and power of SQRT CSLA architecture. The reduced number of gates of this project 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, but the area and power of the modified SQRT CSLA are significantly reduced. The power-delay product and also the area-delay product of the proposed design show a decrease which indicates the success of the method and not a mere trade off of delay for power and area.

The modified CSLA architecture is therefore, low area, low power, simple and efficient for VLSI hardware implementation. The design of the modified 128-b SQRT CSLA can also be implemented which is applicable in higher arithmetic circuits as it requires less area and low power consumption.

It can be observed that for the modified CSLA, the power reduction is 9.8%, area reduction is 25% and delay reduction is 9.5% when compared to the regular SQRT CSLA.

REFERENCES

1. J. M. Rabaey, M. Pedram, "Low Power Design Methodologies", Kluwer Academic Publishers, 1996
2. K. Roy and S. C. Prasad, "Low-Power CMOS VLSI Circuit Design", John Wiley & Sons, 1999.
3. O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput., pp. 340–344, 62.
4. 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.