

## Research Article

### An Improved Modified Carry Select towards Low Power Applications

<sup>1</sup>G.V. Arunraj and <sup>2</sup>R.Varadharaj

<sup>1</sup>Research Scholar, Department of ECE, Bharath University,

<sup>2</sup>Professor & Head, Department of ECE, Sri Lakshmiammal College of Engineering, Chennai, India

**Abstract:** The present investigation aims at Design and implementation of the modules for Carry select adders benchmarking of the results by testing the efficiency of the modules/sub-modules development of improved methodologies for 16-bit carry select adder Comparison of the Gate Count of various Carry Select Adders. The implementation of 16-bit SQR CSLA is done for all the 3 adders and its truth table and design is verified using the test-bench module. Test bench module includes simulating the designed circuitry for different kind of inputs and its output executable and waveform is verified. The implementation of each adder circuit is done according to its respective block diagram and verification is done based on module instantiation, propagation of each sub-module outputs based on hierarchy, carry propagation and selection from multiplexers outputs.

**Keywords:** 16-bit, circuit, carry select adder, input and output, module

## INTRODUCTION

The demand for high performance and low power consumption has become important in today's VLSI circuit design. The need for higher performance has led to the use of Domino circuits where conventional static CMOS circuits may not meet the demand for low critical path delay. However, Domino circuits are more susceptible to noise (for scaled technologies with low transistor threshold voltage) than static CMOS circuits because in the evaluation mode intermediate nodes of Domino circuits may be floating. Another drawback of Domino is its higher power consumption compared to standard complementary CMOS logic. Since a clock signal is necessary for every stage to pre-charge the output nodes of Domino circuits, power consumption due to clock is of concern. One of the solutions to these problems is to use skewed logic circuits, which have good noise-immunity and achieve high performance with low power consumption (Somasekhar, 1999). The circuit topology of skewed logic is the same as that of static CMOS logic, however, the PMOS or the NMOS transistors are preferentially sized to achieve fast high - to - low or low - to - high transitions. For example, to speed up high to low transition, the sizes of PMOS transistors are reduced while the NMOS transistors are sized up.

In today's VLSI circuit designs, there is a significant increase in the power consumption due to the increasing speed and complexity of the circuits. As the demand for portable equipment like laptops and

cellular phones is increasing rapidly, great attention has been focused on power efficient circuit designs (Navi *et al.*, 2009; Wang *et al.*, 2009; Weste and Eshraghian, 1993; Kang and Leblebici, 2005). Adders are the basic building blocks of the complex arithmetic circuits. Adders are widely used in Central Processing Unit (CPU), Arithmetic Logic Unit (ALU) and floating point units, for address generation in case of cache or memory access and in digital signal processing (Rabaey *et al.*, 2002; Uyemura, 1999; Weste and Eshraghian, 1993). Having adders with fast addition operation and lowpower along with low area consumption is still a challenging issue. Depending upon the area, delay and power consumption, the various adders are categorized as Ripple Carry Adder (RCA), Carry Select Adder (CSLA) and Carry Look ahead Adder (CLAA). CSLA provides a compromise between the large area with small delay of CLAA and small area and longer delay of RCA (Rawat *et al.*, 2002).

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. The Carry Select Adder

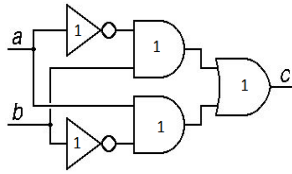


Fig. 1: Delay and area of XOR gate

(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).

Present work deals with an improved carry select adder for low power applications. Efficiency of other adders has also been discussed in this study. Brief description about other adders is given below.

**Basic adder blocks:** In this section we explained how to calculate delay and area theoretically. The AND, OR and Inverter (AOI) implementation of an XOR gate is shown in Fig. 1. 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. Basic adder block 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 1.

**Ripple Carry Adders (RCA):** Arithmetic operations like addition, subtraction, multiplication, division are basic operations to be implemented in digital computers using basic gates like AND, OR, NOR, NAND etc. Half Adders can be used to add two one bit binary numbers. It is also possible to create a logical circuit using multiple full adders to add N-bit binary numbers. Each full adder inputs  $C_{in}$  which is the  $C_{out}$  of the previous adder. This kind of adder is a Ripple Carry Adder, since each carry bit "ripples" to the next full adder. The first full adder may be replaced by a half adder. The structure of a normal ripple carry adder is simple, which allows for fast design time; however, the ripple carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder. The gate delay can easily be calculated by inspection of the full adder circuit. Each full adder requires three levels of logic. In a 16-bit

Table 1: Delay and area evolution of CSLA

DESIGN	DELAY	AREA
XOR	3	5
2:1 MUX	3	4
HALF ADDER	3	6
FULL ADDER	6	13

[ripple carry] adder, there are 16 full adders, so the critical path (worst case) delay is  $15 \times 2$  (for carry propagation)  $+ 3$  (for sum) = 33 gate delays.

**Regular carry select adder using ripple carry adder:** Hence to overcome the delays and the slowness of the ripple carry adder, we use the SQRT Carry select adder. There are different ways to implement the CSLA. A carry-select adder is a particular way to implement an adder, which is a logic element that computes the  $(n+1)$ -bit sum of two  $n$ -bit numbers. The carry-select adder is simple but rather fast, having a gate level depth of  $O(\sqrt{n})$ . 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. The number of bits in each carry select block can be uniform, or variable. In the uniform case, the optimal delay occurs for a block size of  $\lceil \sqrt{n} \rceil$ . When variable, the block size should have a delay, from addition inputs A and B to the carry out, equal to that of the multiplexer chain leading into it, so that the carry out is calculated just in time. 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. The block diagram for the regular SQRT CSLA is shown below (Fig. 2).

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. The structure of the 16-bit SQRT CSLA using RCA is shown in Fig. 2. It has five groups of different size RCA. The delay and area evaluation of each group are shown in the Fig. 2 which the numerals within specify the delay values.

**Delay Calculation of SQRT CSLA:** In total, the carry propagation time through an  $n$ -bit adder block is reduced from  $O(n)$  to the number of stages times the delay of the multiplexers. Naturally, using  $n$  blocks of 1-bit carry-select adders would incur a complexity of  $n$  multiplexers, again resulting in  $O(n)$  delay. Therefore, a partition with (slowly) increasing block-size is chosen.

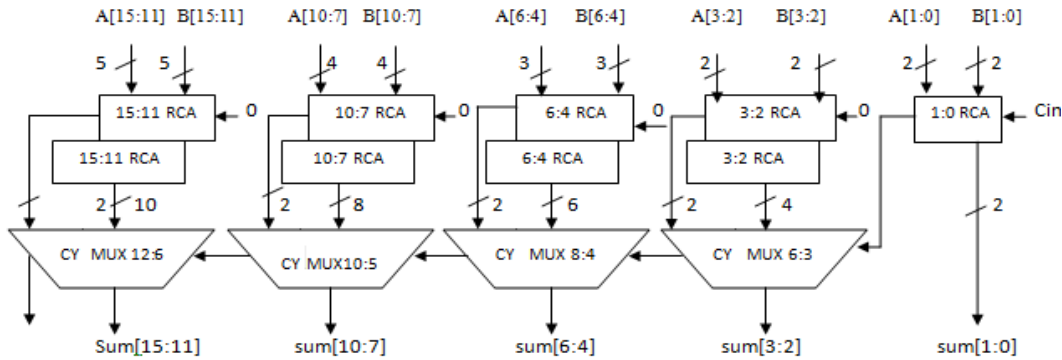


Fig. 2: 16 bit SQRT carry select adder block diagram

Table 2: Asymptotic time and area of different adder

Type	Time	AREA
RCA	$O(n)$	$O(n)$
CSA	$O(\sqrt{n})$	$O(n)$

Table 3: Function table of the 4-bit BEC

Binary [3:0]	Excess-1 [3:0]
0000	0001
0001	0010
:	:
:	:
1110	1111
1111	0000

In the example, the first (least-significant) block consists of a simple full adder, followed by a 3-bit carry-select block and finally a 4-bit carry-select block. A common choice for a 16-bit carry-select adder is to use a 5-4-3-2-2 bit partitioning. While the delay of the standard ripple-carry adder with n-bits is  $O(n)$ , the delay through the carry-select adder behaves as  $O(\sqrt{n})$  at a hardware cost of  $O(3*n)$ . The name SQRT comes from the fact that the delay is of the order of  $O(\sqrt{n})$ . Table 2 presents the asymptotic time and area of different adder.

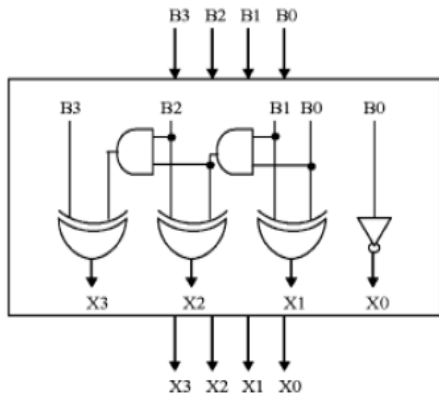


Fig. 3: 4 Bit BEC

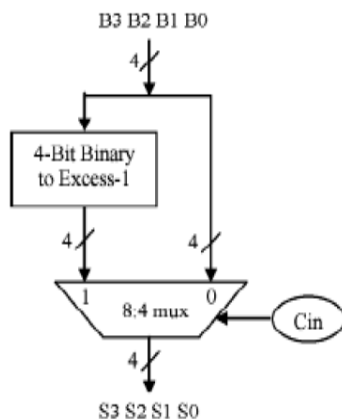


Fig. 4: Structure of 4-bit BEC with 8:4 mux

**Modified SQRT carry select adder:** The main idea of this study is to use BEC instead of the RCA with  $Cin = 1$  in order to reduce the delay and area utilization of the regular SQRT CSLA. To replace the n-bit RCA, an n+1 bit BEC is required. The structure of a 4-bit BEC is shown in Fig. 3 and the function table given in Table 3. Figure 3 illustrates how the basic function of the CSLA is obtained by using the 4-bit BEC together with the mux. In this structure 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 outputs in parallel according to the control signal Cin. 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 expression of the 4-bit BEC is listed as:

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

The modified 16-bit SQRT CSLA using BEC is shown in Fig. 4. The structure is again divided into five groups with different sizes of Ripple carry adder and BEC. The group2, group3, group4 and group5 of 16-bit SQRT CSLA are shown in Fig. 5. The parallel Ripple carry adder with  $Cin = 1$  is replaced with BEC. One input to the multiplexer goes from the RCA with

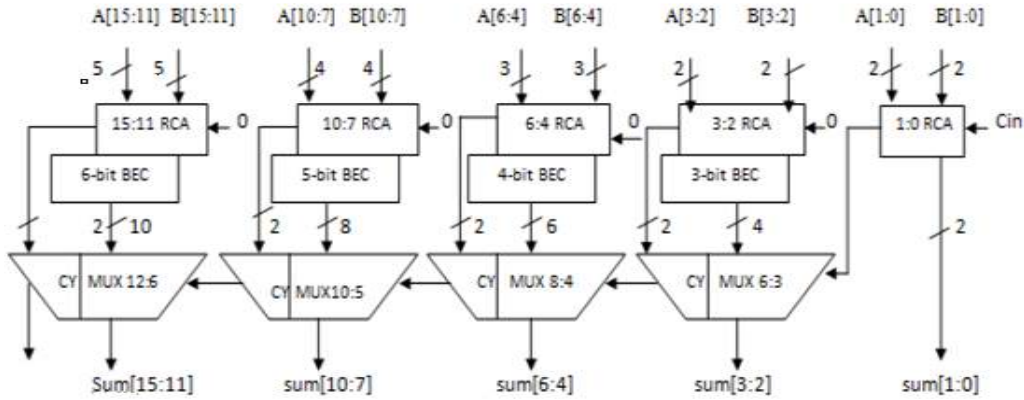


Fig. 5: Modified Sqrt CSLA with BEC

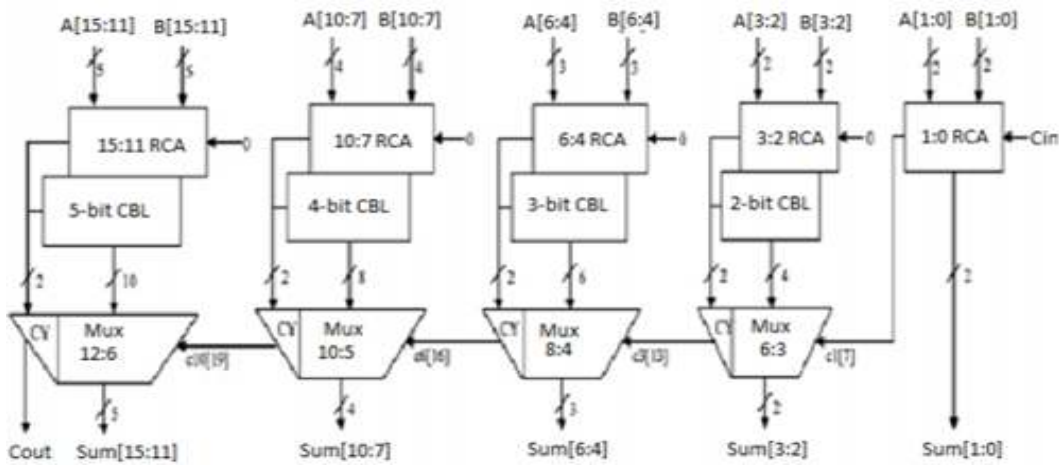


Fig. 6: Proposed carry select adder-block diagram

$C_{in} = 0$  and other input from BEC. Comparing the individual groups of both regular and modified Sqrt CSLA, it is clear that the BEC structure reduces delay. But the disadvantage of BEC method is that the area is increasing than the regular Sqrt CSLA. The block diagram of modified Sqrt CSLA with BEC is shown below. The implementation of the modified CSLA is also done similar to the block diagram.

### MATERIALS AND METHODS

In this study a new architecture has been proposed for power efficiency.

**Proposed carry select adder:** In proposed architecture, an area-efficient carry select adder by sharing the common Boolean logic term to remove the duplicated adder cells in the conventional carry select adder is shown in this way; it saves many transistor counts and achieves a low power. Through analyzing the truth table of a single bit full adder, to find out the output of summation signal as carry-in signal is logic '0' is the inverse signal of itself as carry-in signal is logic '1'. By

sharing the common Boolean logic term in summation generation, a proposed carry select adder design is illustrated in Fig. 6. To share the common Boolean logic term, it only needs to implement one OR gate with one INV gate to generate the carry signal and summation signal pair. Once the carry-in signal is ready, then select the correct carry-out output according to the logic state of carry-in signal. This method replaces the BEC add one circuit by Common Boolean Logic. The proposed 16-bit Sqrt CSLA architecture is shown in Fig. 6. The summation and carry signal for full adder which has  $C_{in} = 1$ , generate by INV and OR gate. Through the multiplexer, the correct output result is selected according to the logic state of carry-in signal. The internal structure of the group3 of proposed CSLA is shown in Fig. 6. One input to the mux goes from ripple carry adder block with  $C_{in} = 0$  and other input from the Common Boolean logic. The group3 performed a three bit addition which are A[4] with B[4], A[5] with B[5] and A[6] with B[6]. This is done by one half adders (HA) and two full adders (FA). The CBL block has a 4:2 multiplexer to select the appropriate carryout and summation signal for carry-in

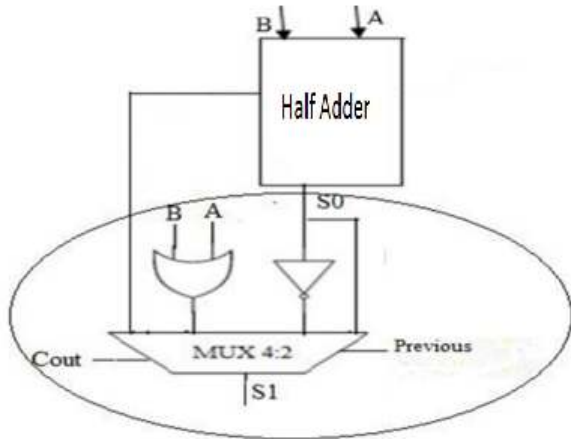


Fig. 7: Single bit full adder with CB

signal “1”. Through 2:1 multiplexer the carry signal is propagate to the next adder cell. The 6:3 multiplexer and 4:2 multiplexer is the combination of 2:1 multiplexer.

**Common Boolean logic explanation:** From the truth table of full adder it can be seen that when the Carry input is ‘0’, the sum will be the ‘xor’ of inputs A and B, while the carry out will be the ‘and’ of A and B. When Carry input is ‘1’, the sum is the ‘not’ of what we got when  $C_{in} = 0$  and the carry out will be the ‘or’ of A and B. So the full adder can be implemented using a XOR, NOT and AND OR gates. Here the logic for the required carry is selected using 2:1 muxes.

Normal equation for full adder is:

$$SUM = A \oplus B \oplus C_{in}$$

$$C_{out} = A \& B \& C_{in}$$

Equation for modified full adder is:

When  $C_{in} = 0$

$$S_i = A \oplus B$$

$$C_i = A \& B$$

When  $C_{in} = 1$

$$S_j = \sim S_i$$

$$C_j = A + B$$

The above truth table is implemented using a Half Adder, an Inverter, OR gate and a 4:2 multiplexer made from 2:1 mux (Fig. 7).

Implementation of above adders is done using verilog HDL. The simulations are carried out using open source software named iverilog (Icarus Verilog). Icarus Verilog is a Verilog simulation and synthesis tool. It operates as a compiler, compiling source code written in Verilog (IEEE-1364) into some target format. For batch simulation, the compiler can generate an intermediate form called *vvp assembly*. This intermediate form is executed by the “vvp” command. For synthesis, the compiler generates net lists in the desired format.

**Implementation of a RCA using iverilog:** The implementation of SQR CSLA requires the development of various submodules. The various sub modules are mentioned in the order of hierarchy from bottom level to the top level in which they are required and implemented. Each of these modules are separately designed and tested with verilog Hardware Descriptive Language and integrated in order to obtain the CSLA:

- Half Adder
- Full Adder
- Ripple Carry Adder – generic for n – bit addition
- Multiplexer
- Carry Select Adder

## RESULTS AND DISCUSSION

**Gate calculation for the 3 adders:** The common and useful combinational logic circuit can be constructed using basic logic gates. Any binary adder is made up from Inverter and, OR and XOR gates. In this study the architectures of Regular, Modified and proposed carry select adders gate counts are given in terms of Inverter, NAND and NOR Gates. The number of gates required for AND gate and OR gates in terms of Inverter, NAND and NOR are 1 1 0 and 1 0 1 respectively. Similarly the gate count required for XOR is given as follows.

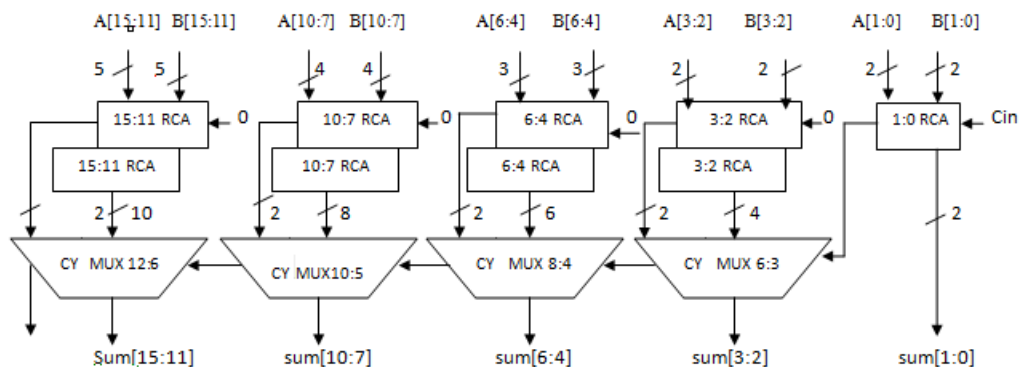


Fig. 8: Regular SQR CSLA with dual RCA



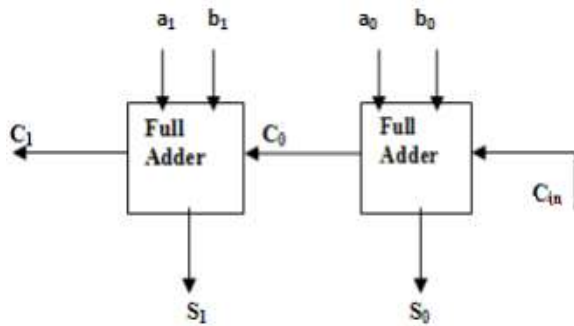


Fig. 9: Group (1) of the adder circuit

Table 4: Gate count of the basic circuits

Name/Gate type	Inverter	NAND	NOR
AND	1	1	0
OR	1	0	1
XOR	5	2	1
2x1 Mux	4	2	1
HALF ADDER	6	3	1
FULL ADDER	8	4	2

Table 5: Gate counts of five groups of modified carry select adder

Group number	Gate count
Group (1)	28
Group (2)	64
Group (3)	95
Group (4)	126
Group (5)	157

There are 2 AND gates, 1 OR gate, 2 Inverters present. The total gate do not count in terms of Inverter, Nand, Nor gates are 5 2 1. Similarly the gate counts of different gates that are used in the Regular, Modified and Common Boolean Logic carry select adders architectures are given in the Table 4. The Architecture of Regular 16-bit carry select adder is shown in (Fig. 8). It has five groups, each is of different size.

The Gate count evaluation of group (1) and group (2) is shown in Fig. 9. The group (1) has two full adders. The structure of it is shown in Fig. 9. The Gate count in terms of Inverter is determined as follows:

$$\begin{aligned} \text{Gate count} &= 2 * \text{Full Adders (FA)} \\ &= 2 * 14 \\ &= 28 \end{aligned}$$

**Modified 16-bit carry select adder using BEC:** The architecture of Modified 16-bit carry select adder is shown in Fig. 10.

The difference between the regular and modified 16-bit carry select adders lies at  $c_{in} = 1$ , that is, in regular carry select adder ripple carry adder is used where as in modified 16-bit carry select adder Binary to excess one converter is used for  $c_{in} = 1$ . In Modified carry select adder also there are five different groups. The Gate count evaluation is shown in Table 5.

The total number of gates for Modified 16-bit Carry select adder is 470. By comparing the gate counts of above two architectures, the number of gates of the latter architecture is reduced by 60.

**Proposed carry select adder using common Boolean logic:** The proposed carry select adder is constructed by using the common Boolean logic (Fig. 11).

**Comparison of results:** A comparison of different 16-bit carry select adders in terms of inverter, Nand and Nor gates are given where the 16-bit regular and modified carry select adders have different groups for  $c_{in} = 0$  and  $c_{in} = 1$  and the proposed common Boolean logic carry select adder has sixteen similar groups. The below graph (Fig. 12) shows the comparison of gate counts of five different groups of regular and modified carry select adders.

Gate count comparison shows comparison of total number of gates required for 16-bit regular, modified and proposed carry select adders. Comparing 16-bit regular and modified with proposed carry select adders the number of gates for the proposed carry select adder reduced by 98 and 38 respectively (Fig. 12).

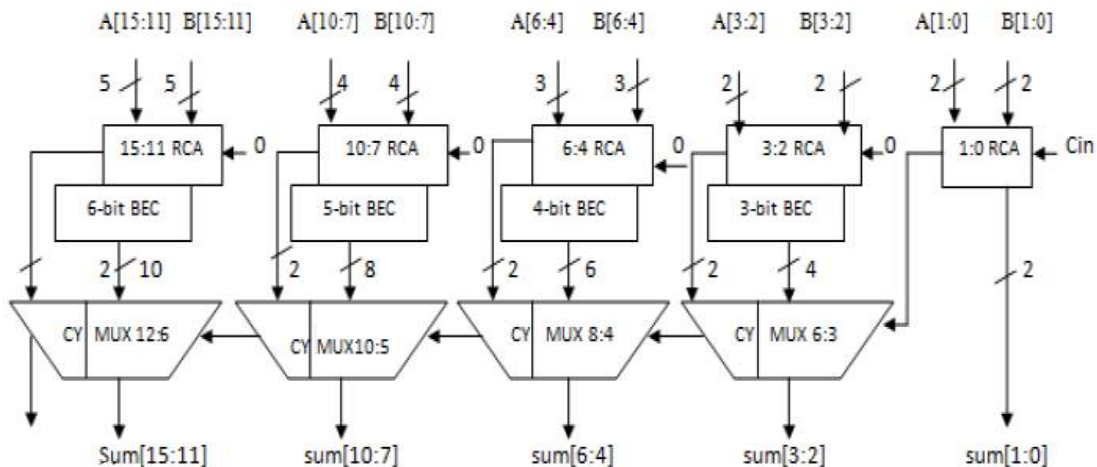


Fig. 10: Block diagram of the modified carry select adder

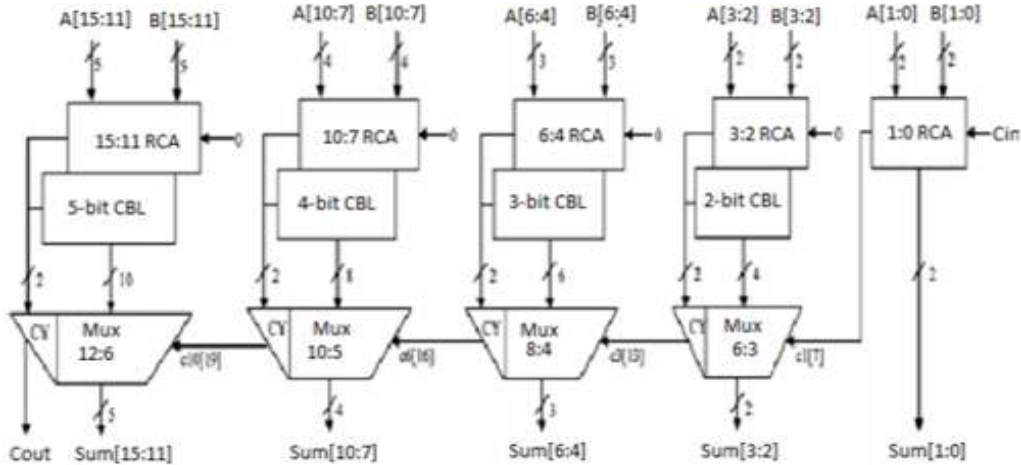


Fig. 11: Block diagram of the proposed carry select adder

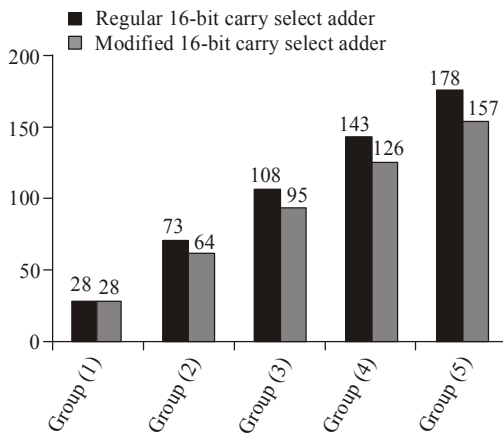


Fig. 12: Gate count comparison

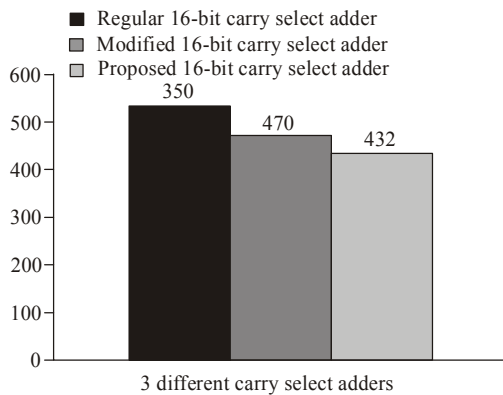


Fig. 13: Gate counts of carry select adders

The below chart (Fig. 13) shows comparison of total number of gates required for 16-bit regular, modified and proposed carry select adders. Comparing 16-bit regular and modified with proposed carry select adders the number of gates for the proposed carry select adder reduced by 98 and 38 respectively.

The above gate comparison shows that there is a significant reduction in the area occupied by the logic

implementation of the 3 Square Root Carry Select Adders. Comparison of different 16-bit carry select adders are given and is determined that proposed carry select adder has reduced gate count than the other two. The percentage decrease in gate count of proposed 16-bit common Boolean logic carry select adder when compared with 16-bit regular and modified carry select adders are 18.5% and 8% respectively.

**Area calculation:** Slices are the basic building block components in the FPGA fabric. However each slice contains a number of LUT's, flip-flops and carry logic element which make up the logic of your design before mapping. After mapping, all of the LUT's and flip-flops are packed into slices, but not necessarily filling the slices. i.e., a slice with two LUT's and two flip-flops may be in use for just one LUT. In the map report any slice that is used even partially is counted in the "occupied slices". You will usually notice that the percentage of usage of slices is greater than the larger of LUT's and flip-flops. i.e., your design may use about 25% of LUT's and flip-flops but because of sparse packing it can have nearly 50% occupied slices. It may be possible to fit the design into fewer slices, but if it wasn't necessary (i.e., there are still slices left over) the mapper will not try to pack the logic any further. Area estimation technique employed at the design as well as the implementation phase play a significant role in realizing efficient FPGA resources. Fast and accurate resource estimation technique for an FPGA-based design is essential for the efficient utilization of the hardware resources in any design. In FPGA based design the hardware area utilized is provided in terms of look-up table (LUT's) or configurable logic blocks (CLB's) slices.

### CONCLUSION

With the use of BEC (Binary to Excess 1 Converter) and CBL (Common Boolean Logic), the

use of 2<sup>nd</sup> group of Ripple Carry Adder (RCA) is eliminated from the CSLA and hence there is reduction in the number of gates and full adder circuitry. The BEC makes sure that the  $C_{in} = 1$  carry input is propagated through to the output adding an excess of one using the Boolean logic mentioned in the above chapter. Common Boolean Logic further reduces the number of gates used in place of the RCA in regular SQR CSLA and BEC in modified SQR CSLA.

The implementation of 16-bit SQR CSLA is done for all the 3 adders and its truth table and design is verified using the test-bench module. Test bench module includes simulating the designed circuitry for different kind of inputs and its output executable and waveform is verified. Gate count level calculation and analysis is done theoretically based on the number of gates required for each sub-module and the overall circuitry. Area, Delay analysis and Power Consumption can be done with the help of the various tools from Xilinx mentioned using the command line tools.

#### REFERENCES

- Kang, S. and Y. Leblebici, 2005. CMOS Digital Integrated Circuit Analysis and Design. 3rd Edn., McGraw-Hill, New York, NY, USA.
- Navi, K., M.H. Moaiyeri, R.F. Mirzaee, O. Hashemipour and B.M. Nezhad, 2009. Two new low-power Full Adders based on majority-not gates. *Microelectron. J.*, 40(1): 126-130.
- Rabaey, J.M., A. Chandrakasan and B. Nikolic, 2002. *Digital Integrated Circuits, a Design Perspective*. Prentice Hall, Englewood Cliffs, NJ, USA.
- Rawat, K., T. Darwish and M. Bayoumi, 2002. A low power and reduced area carry select adder. *Proceeding of the 45th Midwest Symposium on Circuits and Systems (MWSCAS)*, pp: 1467-1470.
- Somasekhar, D., 1999. Power and dynamic noise considerations in high-performance CMOS VLSI. Ph. D. Thesis, Purdue University.
- Uyemura, J.P., 1999. *CMOS Logic Circuit Design*. Kluwer Academic, Boston.
- Wang, D., M.F. Yang, W. Cheng, X.G. Guan, Z.M. Zhu and Y.T. Yang, 2009. Novel low power full adder cells in 180nm CMOS technology. *Proceeding of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA '09)*. China, pp: 430-433.
- Weste, N.E. and K. Eshraghian, 1993. *Principles of CMOS VLSI Design: A Systems Perspective*. Addison-Wesley Pub. Co., Reading, Mass.