

A VLSI Implementation of Fast Addition Using an Efficient CSLAs Architecture

N. Salma Sulthana¹

Asst. Professor¹

Department of Electronics & Communication Engineering

Princeton Institute of Engineering & Technology for Women, Telangana State, India

Corresponding Author's Email: salmasulthana4u@gmail.com¹

Abstract

Carry Select Adder (CSLA) is a fast adder used in data processing processors for performing fast arithmetic functions. From the structure of the CSLA, the scope is reducing the area of CSLA based on the efficient gate-level modification. However, the Regular CSLA is still area-consuming due to the dual Ripple-Carry Adder (RCA) structure. For reducing area, the CSLA can be implemented by using a single RCA and an add-one circuit instead of using dual RCA. The modified CSLA architecture has developed using Binary to Excess-1 converter (BEC). This paper proposes an efficient method which replaces the BEC using D latch. Experimental results are compared and the result analysis shows that the proposed architecture achieves the two folded advantages in terms of area and delay. This project was aimed for implementing high performance optimized FPGA architecture. Modelsim 10.0c is used for simulating the CSLA and synthesized using Xilinx PlanAhead13.4. Then the implementation is done in Virtex5 FPGA Kit.

Keywords: - *FPGA, CSLA, SQRT CSLA, BEC, AREA EFFICIENT and D-LATCH*

INTRODUCTION

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 was summed and a carry propagated

into the next position. Bedriji proposed [1] that the problem of carry propagation delay is overcome by independently generating multiple radixes 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 [4]. Chang and Hsiao proposed [3] that instead of using dual Ripple Carry Adder a Carry Select Adder scheme using an add one circuit to replace one RCA. Youngioon 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 (SQRT CSLA) scheme based on a new first zero detection logic [9]. 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 [2]. Ramkumar and Harish proposed [8] BEC technique, which is a simple and efficient gate level modification to significantly reduce the area of SQRT CSLA. Padma Devi et al proposed [10] Modified CSLA designed in different stages which reduces the area. CSLA is used in many computational systems to relieve the problem of carry propagation delay by independently generating 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). Multiple carries and then select a carry to generate the sum [1].

BINARY TO EXCESS-1 CONVERTER (BEC)

BEC is a circuit used to add 1 to the input numbers. A circuit of 3-bit BEC and the function table is shown in Fig 1.5 and Table 1 respectively. The main objective of this project is to reduce the gate level by using Binary to Excess-1 Converter. In order to reduce the delay and power we use $n+1$ Binary to Excess-1 Converter instead of n RCA.

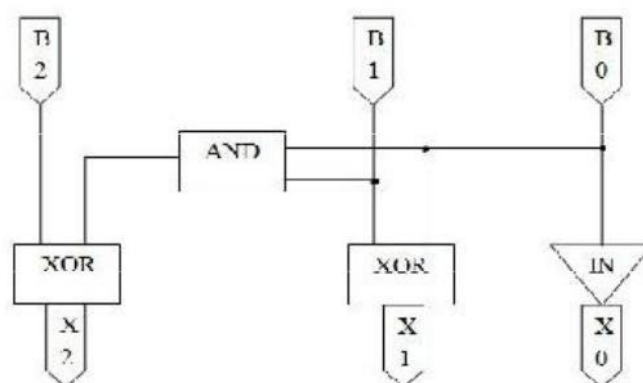


Fig.1 3-Bit Binary to Excess-1 Converter

The Boolean expression of the 3-bit BEC is shown below

$$X_0 = \sim B_0 \quad (1)$$

$$X_1 = B_0 \quad (2)$$

$$X_2 = B_2 \quad (3)$$

Table: 1 Function Table of 3-Bit Binary to Excess-1 Convertor

BINARY [3:0]			EXCESS-1 [3:0]		
B ₂	B ₁	B ₀	X ₂	X ₁	X ₀
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

AREA EVALUATION METHODOLOGY OF THE BASIC ADDER BLOCKS

The AND, OR, and Inverter (AOI) implementation of an XOR gate is shown in Fig 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. 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. The delay calculation is done by using the parallel performance of work in XOR gate. Based on this approach, the CSLA adder blocks of 2:1 mux, Half Adder (HA), and Full Adder (FA) are evaluated and listed in Table 2.

Table 2 Delay and Area count of the blocks of CSLA

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

AREA EVALUATION METHODOLOGY OF REGULAR 16-BIT LINEAR AS WELL AS SQRT CSLA

The structure of the 16-bit regular Linear CSLA is shown in Fig.3. It has 4 groups of same size RCA. Each group contains dual RCA and Mux. It accomplishes the addition by adding small portions of bits (each of equal size) and wait for the carry to complete the calculation. Both sum and carry are calculated for both possible solutions. The linear CSLA is constructed by chaining a number of equal length adder stages. Here the equal size of inputs is given to each block of the adder. The steps leading to the evaluations are given here. In the regular Linear CSLA, the group2 has two sets of 4-bit RCA. Then the area count of group2 is determined as follows:

$$\text{Gate count} = 117 (\text{FA} + \text{HA} + \text{MUX})$$

$$\text{HA} = 6 (1 * 6)$$

$$\text{Mux} = 20 (5 * 4)$$

$$\text{FA} = 91 (7 * 13)$$

Similarly the estimated area of other groups in the regular Linear CSLA are evaluated and listed in Table 3.

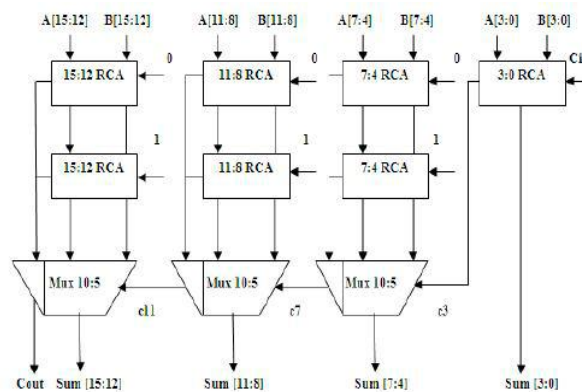


Fig. 3. Regular 16-bit Linear CSLA

Table 3 Area Count of 16-Bit Regular Linear CSLA Groups

GROUP	AREA COUNT
Group1	52
Group2	117
Group3	117
Group4	117

The structure of the 16-bit regular Sqrt CSLA is shown in Fig. 4. It has 5 groups of different size RCA. Each group contains dual RCA and Mux. The linear carry select adder has two disadvantages there are high area usage and high time delay. These disadvantages of linear carry select adder can be rectified by Sqrt CSLA. It is an improved one of linear CSLA. The time delay of the linear adder can decrease by having one more input into each set of adders than in the previous set. This is called a Square Root Carry Select Adder. Square Root carry select adder is constructed by equalizing the delay through two carry chains and the block-multiplexer signal from previous stage. The steps leading to the evaluations are given here. In the regular Sqrt CSLA, 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 otherwise it select second RCA output. The output of group2 are Sum [3:2] and carryout, c3. Then the area count of group2 is determined as follows:

$$\text{Gate count} = 57 (\text{FA} + \text{HA} + \text{Mux})$$

$$\text{FA} = 39 (3 * 13)$$

$$\text{HA} = 6 (1 * 6)$$

$$\text{Mux} = 12 (3 * 4)$$

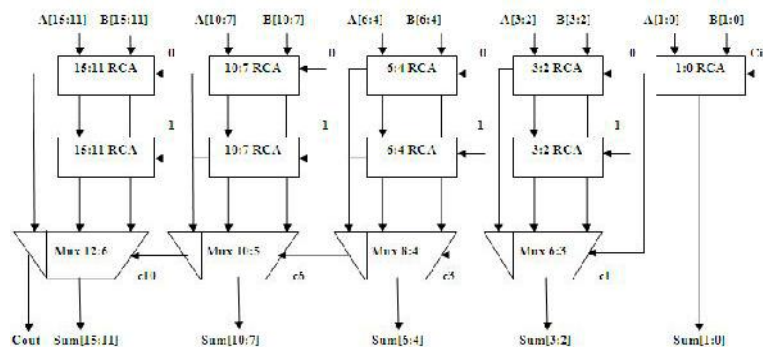


Fig.4 Regular 16-bit Sqrt CSLA

Similarly the estimated area of the other groups in the regular SQRT CSLA are evaluated and listed in Table 4

Table 4 Area Count of the 16-Bit Regular SQRT CSLA Groups

GROUP	AREA COUNT
Group1	26
Group2	57
Group3	87
Group4	117
Group5	147

AREA EVALUATION METHODOLOGY OF MODIFIED 16-BIT LINEAR AS WELL AS SQRT CSLA

The structure of the proposed 16-bit Linear and SQRT CSLA using BEC for RCA with carry in = 1 to optimize the area is shown in Fig. 5 and Fig. 6 respectively. The 16-bit modified Linear CSLA has 4 groups of same size RCA and BEC. Each group contains one RCA, one BEC and Mux. In the modified Linear CSLA, the group2 has one 4-bit RCA which has 3 FA and 1 HA for carry in = 0. Instead of another 4-bit RCA with carry in = 1 a 5-bit BEC is used which adds one to the output from 4-bit RCA. Then the area count of group2 is determined as follows:

$$\text{Gate count} = 89 (\text{FA} + \text{HA} + \text{Mux} + \text{BEC})$$

$$\text{FA} = 39 (3 * 13)$$

$$\text{HA} = 6 (1 * 6)$$

$$\text{Mux} = 20 (5 * 4)$$

$$\text{NOT} = 1$$

$$\text{AND} = 3 (3 * 1)$$

$$\text{XOR} = 20 (4 * 5)$$

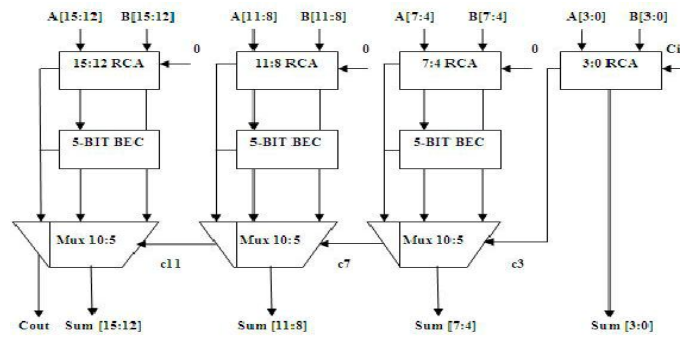


Fig.5 Modified 16-bit Linear CSLA

Similarly the estimated area of the other groups in the modified Linear CSLA are evaluated and listed in Table 5

Table 5 Area Count of the 16-Bit Modified Linear CSLA Groups

GROUP	AREA COUNT
Group1	52
Group2	89
Group3	89
Group4	89

The structure of the 16-bit modified Sqrt CSLA is shown in Fig. 6. It has 5 groups of different size RCA and BEC. Each group contains one RCA, one BEC and MUX. In the modified Sqrt CSLA, the group2 has one 2-bit RCA which has 1 FA and 1 HA for carry in = 0. Instead of another 2-bit RCA with carry in = 1 a 3-bit BEC is used which adds one to the output from 2-bit RCA. Then the area count of group2 is determined as follows:

$$\text{Gate count} = 43 (\text{FA} + \text{HA} + \text{Mux} + \text{BEC})$$

$$\text{FA} = 13 (1 * 13)$$

$$\text{HA} = 6 (1 * 6)$$

$$\text{Mux} = 12 (3 * 4)$$

$$\text{NOT} = 1$$

$$\text{AND} = 1$$

$$\text{XOR} = 10 (2 * 5)$$

$$\text{BEC (3-BIT)} = \text{NOT} + \text{AND} + \text{XOR} = 12$$

Similarly the estimated area of the other groups in the modified SQRT CSLA are evaluated and listed in Table 6

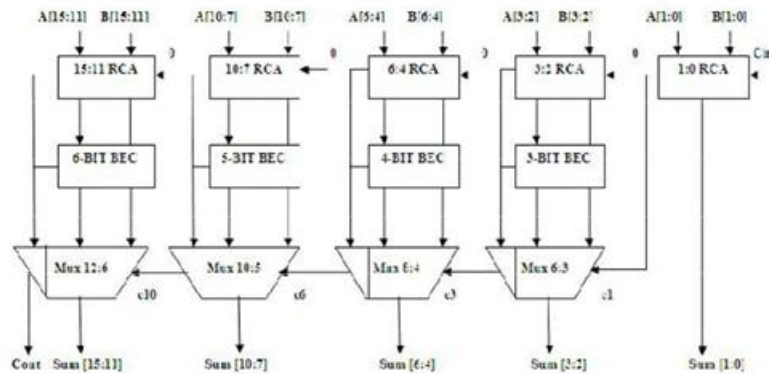


Fig.6 Modified 16-bit SQRT CSLA

Table 6 Area Count of the 16-Bit Modified SQRT CSLA Groups

GROUP	AREA COUNT
Group1	26
Group2	43
Group3	66
Group4	89
Group5	113

PROPOSED CARRY SELECT ADDER

This method replaces the BEC add one circuit by D-latch with enable signal. Latches are used to store one bit information. Their outputs are constantly affected by their inputs as long as the enable signal is asserted. In other words, when they are enabled, their content changes immediately according to their input

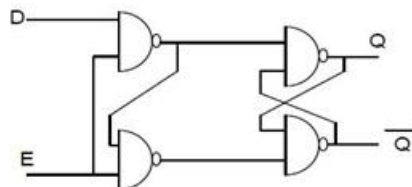


Fig.7 D-Latch

This is 16-bit adder in which least significant bit (LSB) adder is ripple carry adder, which is 2 bit wide. The upper half of the adder i.e, most significant part is 14-bit wide which works

according to the clock. Whenever clock goes high addition for carry input one is performed. When clock goes low then carry input is assumed as zero and sum is stored in adder itself. it can understand that latch is used to store the sum and carry for $C_{in}=1$. Carry out from the previous stage i.e., least significant bit adder is used as control signal for multiplexer to select final output carry and sum of the 16-bit adder. If the actual carry input is one, then computed sum and carry latch is accessed and for carry input zero MSB adder is accessed. Cout is the output carry. The internal structure of group 2 of the proposed 16-bit CSLA. The group 2 performed the two bit addition which is a_2 with b_2 and a_3 with b_3 . This is done by two full adder (FA) named FA2 and FA3 respectively. The third input to the full adder FA2 is the clock instead of the carry and the third input to the full adder FA3 is the carry output from FA2. The group 2 structure has three D-Latches in which two are used for store the sum2 and sum3 from FA2 and FA3 respectively and the last one is used to store carry. Multiplexer is used for selecting the actual sum and carry according to the carry is coming from the previous stage. The 6:3 multiplexer is the combination of 2:1 multiplexer.

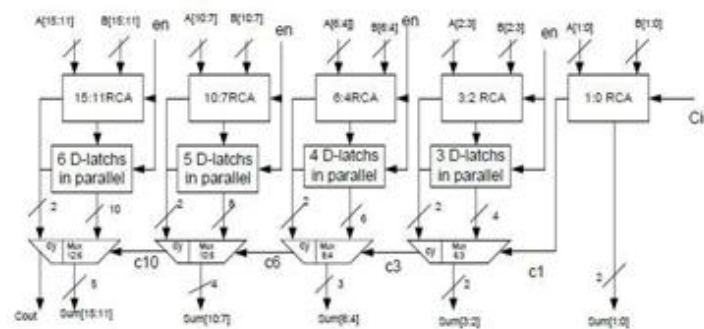


Fig.8 Proposed CSLA

When the clock is low a_2 and b_2 are added with carry is equal to zero. Because of low clock, the D-Latch is not enabled. When the clock is high, the addition is performed with carry is equal to one. All the D-Latches are enabled and store the sum and carry for carry is equal to one. According to the value of c_1 whether it is 0 or 1, the multiplexer selected the actual sum and carry.

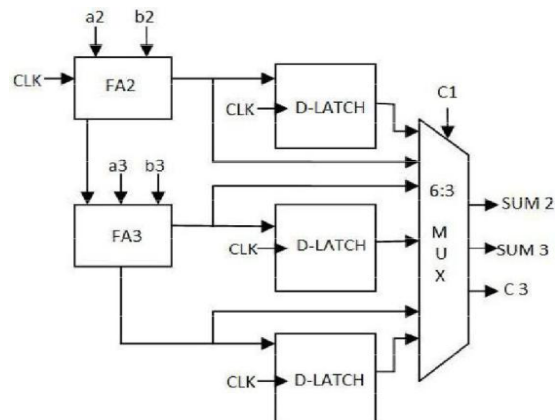


Fig.9 Gropu2 Structure

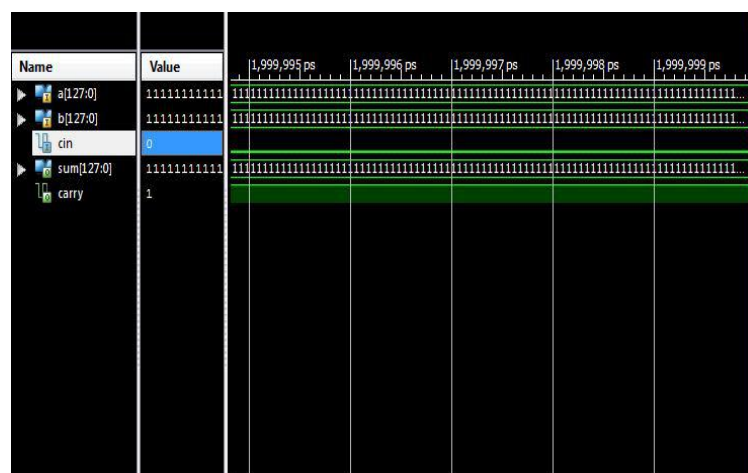


Fig.9(a) 128-Bit Regular Linear CSLA

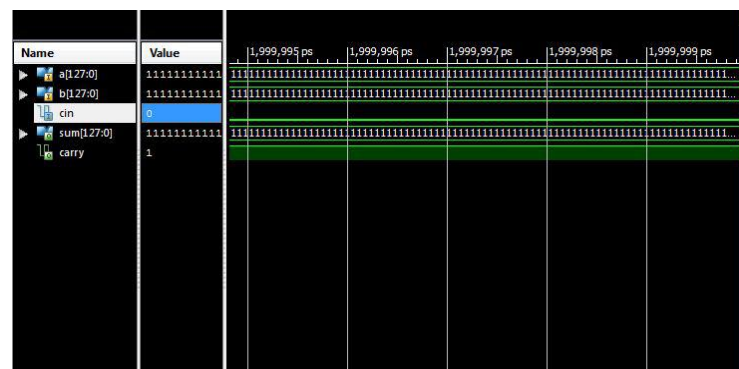


Fig.9(b) 128-Bit Regular SQRT CSLA

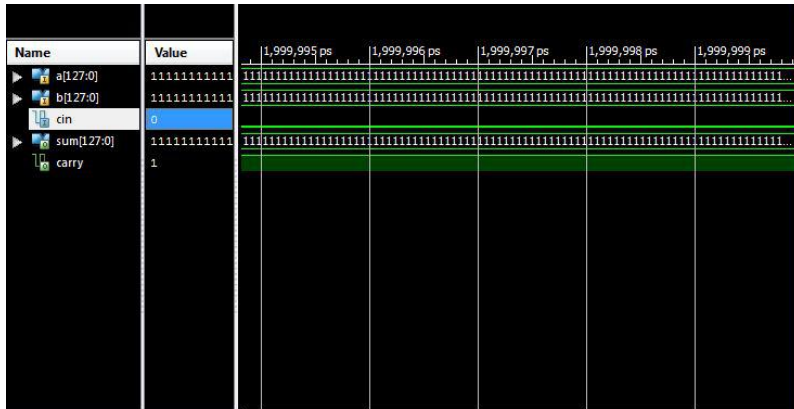


Fig.9(c) 128-bit Modified Regular Linear CSLA

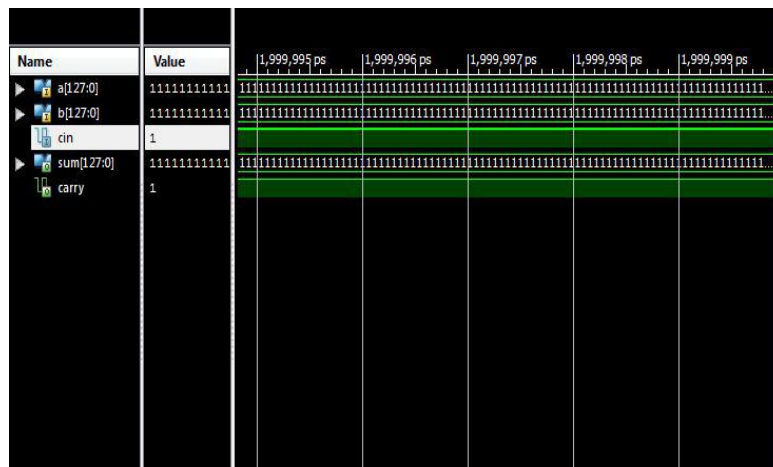


Fig.9(d) 128-bit Modified SQRT CSLA

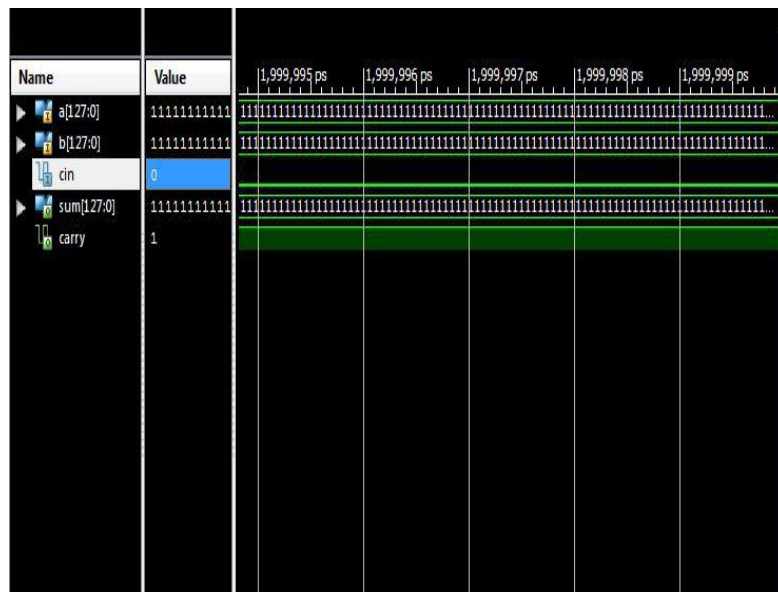


Fig.9(e) 128-bit Modified SQRT CSLA

TABLE 7

COMPARISON OF CSLAs BASED ON AREA COUNT

BIT SIZE	TYPES	AREA COUNT OF LINEAR CSLA	AREA COUNT OF SQRT CSLA
16-BIT	REGULAR	403	434
	MODIFIED	319	337
	PROPOSED	388	424
32-BIT	REGULAR	806	868
	MODIFIED	638	674
	PROPOSED	776	848
64-BIT	REGULAR	1612	1736
	MODIFIED	1276	1348
	PROPOSED	1552	1696
128-BIT	REGULAR	3224	3472
	MODIFIED	2552	2696
	PROPOSED	3104	3392

CONCLUSION

A regular CSLA uses two copies of the carry evaluation blocks, one with block carry input is zero and other one with block carry input is one. Regular CSLA suffers from the disadvantage of occupying more chip area. The modified CSLA reduces the area and power when compared to regular CSLA with increase in delay by the use of Binary to Excess-1 converter. This paper proposes a scheme which reduces the delay and area than regular and modified CSLA by the use of D-latches.

REFERENCES

1. O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Computer, 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, Digital 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 low power applications," in Proc. IEEE Int. Symp. Circuits Syst., vol. 4, pp. 4082–4085, 2005.
7. Cadence, "Encounter user guide," Version 6.2.4, March 2008.
8. 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.
9. He, Y. Chang, C. H. and Gu, J. "An Area Efficient 64-Bit Square Root Carry-Select Adder for Low Power Applications," in Proc. IEEE Int. Symp. Circuits Syst., Vol.4, pp. 4082–4085, 2005.
10. Padma Devi, Ashima Girdher and Balwinder Singh "Improved Carry Select Adder with Reduced Area and Low Power Consumption," International Journal of Computer Applications, Vol.3, No.4, pp. 14-18, 1998.
11. Akhilesh Tyagi, "A Reduced-Area Scheme for Carry-Select Adders, IEEE Transactions on Computers, Vol.42, No.10, pp.1163-1170, 1993.

AUTHOR



First Author: N.SALMASULTHANA received the B.Tech degree in Electronics and Communication Engineering in the year 2011 and M.Tech degree in VLSI System Design in the year 2014 in JNTUH, Telangana.