ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 8, August 2015

Improved Performance and Simplistic Design of CSLA with Optimised Blocks

E S BHARGAVI[1] N KIRANKUMAR[2] H CHANDRA SEKHAR[3] L RAMAMURTHY[4]

Abstract

There have been many advances in updating the adders, initially, from conventional CSLA (Carry Select Adder) to the recent update. The proposed method compares the logic formulae and data dependencies of all the existing methods and comes up with the method where we can reduce the data dependency along with the logic operation. The main theme is to analyze these logic operations and data dependency of previous methods to make some changes, whose outcome is more enhancing features that, includes Area-Delay-Power consumption. In this project, particularly we have considered latest adder that has been designed based on the logic equations based CSLA and conventional CSLA. We have reduced all the redundant logic operations and proposed new logic approach for CSLA.

***Keywords*:CSLA, power-delay-Area, Data-dependencies, Logic equations, Verilog, Xilinx**

I. INTRODUCTION

Adder plays a major role in many areas like designing arithmetic unit, Digital Integrated circuits etc., at every part of design adder has its own importance. As adder is considered as central element of designing any arithmetic unit like multiplication, division, subtraction etc., it is always important to design adder with most effective features of Area- Delay-Power consumption. Any VLSI circuit contains atleast thousands of transistors that consumes power, area and processes delay to generate the output.

AUTHORS INFO

Author 1 E S Bhargavi*
Address: Dept. of ECE, VEMU Institute of Technology, P.Kothakota, Chittoor.

Author 2 N Kiran Kumar
Address: Dept. of ECE, VEMU Institute of Technology, P.Kothakota, Chittoor

Author 3 H Chandrasekhar, Co-ordinator.
Address: Dept. of ECE, VEMU Institute of Technology, P.Kothakota, Chittoor

Author 4 L Ramamurthy, HOD.
Address: Dept. of ECE, VEMU Institute of Technology, P.Kothakota, Chittoor

Of all these area has more importance as it impacts on power and delay and area consumed by the circuit is calculated by the number gates used in the circuit. Reducing the area and increased speed data path logic units are main area of research in VLSI design.

On the basis of requirements such as area, delay and power consumption there are some complex adders such as Ripple Carry Adder, Carry look-Ahead Adder and Carry Select Adder, carry save adder etc., Ripple carry adders exhibits the most compact design but less in speed. Whereas carry look ahead is the fastest one but consumes more area. Carry select adders act as a compromise between the two adders. The CSLA is used in many computational systems to improve 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 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).Ripple Carry Adder, Carry Look Ahead Adder, Carry Save Adder, Carry Skip Adder which have its own advantages and also drawbacks. The major speed limitation in any adder is in the production of carries and many researchers considered the addition problem. CSLA is cultivated to solve the carry propagation delay which extensively decreases area and delay. Design of circuit resembles the speed of particular circuit. Performance of any circuit is limited by speed most of the times as it includes propagation of carry. As sum of each bit is an elementary adder is generated sequentially

only after the previous bit carry is generated that includes previous bit sum to be generated later a carry propagate in to next position. To reduce the area and power consumption in Carry Select Adder, square root CSLA architecture has been designed. A conventional carry select adder (CSLA) is an RCA–RCA configuration that generates a pair of sum words and output carry bits corresponding the anticipated input-carry (Cin =0 and 1) and selects one out of each pair for final-sum and final-output-carry. Carry-select method has deemed to be a good compromise between cost and performance issues.

II ADDERS

Many different adders has been designed based on the requirement as the requirement changes from application to application for example in few cases delay is the major factor to be considered and in few area play an important role therefore based on the requirement the adder used changes. We have different type of adders like

- Half Adder
- Full Adder
- Parallel Adder
- Carry Look Ahead Adder
- Ripple Carry Adder
- Floating point Adder
- Carry Save Adder
- Carry Select Adder

These are the different adders designed based on different requirements. Out of which few adders have excellent speed but lacks in area and power and few other has excellent design feature of compact area but whose delay and power increases drastically.

Half Adders are the basic adder that can be used in any other adder design, where each bit is added and generates both sum and carry of each bit. Drawback of this adder design is that it does not have scope for input carry, where Full Adder is designed to generate sum and carry based on input carry. When we consider carry lookahead adder design its speed is its advantage where the sum is generated with high speed but the design of such adder consumes more space hence the design area limits the carry look ahead adder. When we see the design of Ripple carry adder the design of this adder consumed less area when compared to carry look ahead adder design, but as its waits for carry to be generated at each stage the delay has been increased thus it limits the speed of generating the sum and output carry. When carry select adder comes into picture its design and features are the compact between carry lookahead adder and Ripple carry adder. Research on carry select adder design has resulted in different architectures of carry select adder.

III CARRY SELECT ADDER (CSLA)

Starting from conventional CSLA the design of carry select adder has been modified and all the research works has designed more accurate architectures. To list few CSLA designs we have Add-One circuit CSLA, SQRT-CSLA, Common Boolean Logic CSLA, BEC-based CSLA, Carry Select adder with sharing etc.,

A. Conventional CSLA:

General carry select adder performs the addition operation assuming carry as 1 and 0 to avoid delay. Conventional CSLA consists of 2-RCA (Ripple carry Adder) units and carry select unit (CS) , where 1 RCA generates sum and carry considering input carry as 1 (Cin=1) and another RCA generates sum and carry considering input carry as 0 (Cin=0).

First let us see the design structure of RCA:

Ripple carry adder is designed by group of full adders where each bit carry is given to the next full adder, since the carry ripples from right to left the circuit is named as ripple carry adder. In Fig: 1 the circuit diagram of Full adder is shown, where the sum(S) and carry (Cout) is generated based on the input and input carry.

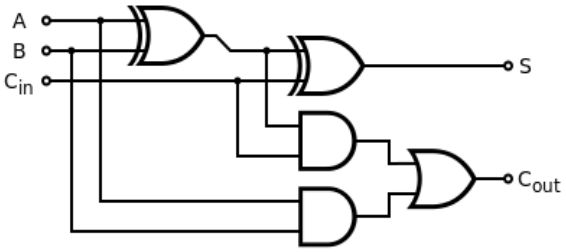


Fig. 1: Full Adder.

i. Ripple carry adder:

As Ripple carry adder consists of sequence of full adders the structure is shown in Fig: 2

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 8, August 2015

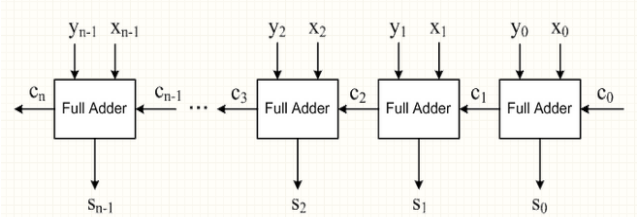


Fig. 2: Ripple Carry Adder.

The number of gates used here depends on the bit-length.

Design of conventional CSLA:

A conventional carry select adder consists of Ripple carry adder and carry select (CS) unit as shown in Fig: 3

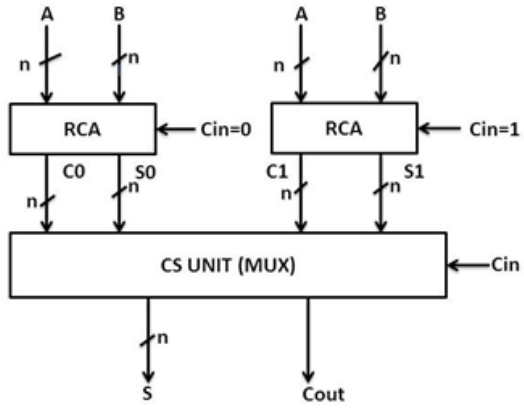


Fig. 3: Conventional CSLA.

As shown in fig: 3, a conventional CSLA consists of two n-bit Ripple carry adders where 1 RCA performs addition assuming carry input Cin= 1 and other RCA assuming Cin=0, Where n represents the input bit length.

Logic Expressions:

$$S_0^0(k) = A(k) \oplus B(k)\,;\, C_0^0(k) = A(k) \bullet B(k) \quad 1(a)$$

$$S_1^0(k) = S_0^0(k) \oplus C_1^0(k-1) \quad 1(b)$$

$$C_1^0(k) = C_0^0(k) \oplus S_0^0(k) \bullet C_1^0(k-1), C_{out}^0 = C_1^0(n-1) \quad 1(c)$$

$$S_0^1(k) = A(k) \oplus B(k)\,;\, C_0^1(k) = A(k) \bullet B(k) \quad 1(d)$$

$$S_1^1(k) = S_0^1(k) \oplus C_1^1(k-1) \quad 1(e)$$

$$C_1^1(k) = C_0^1(k) \oplus S_0^1(k) \bullet C_1^1(k-1), C_{out}^1 = C_1^1(n-1) \quad 1(f)$$

$$where C_1^0(-1) = 0; C_1^1(-1) = 1; and 0 \le k \le n-1 \quad 1(g)$$

ii. Design of BEC-based CSLA:

Conventional CSLA is modified by using n-bit Binary to Excess-1 code converters (BEC) to improve the speed of addition. This logic can be implemented with any type of adder to further improve the speed. We use the Binary to Excess-1 Converter (BEC) instead of RCA with Cin=1 in the regular CSLA to achieve lower area and power consumption. To replace the n-bit RCA, an n+1-bit BEC is required.

The basic idea of this modified work is to use Binary to Excess-1 Converter (BEC) instead of RCA with Cin=1 in the regular CSLA to achieve less area and power consumption with only a slight increase in the delay. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure

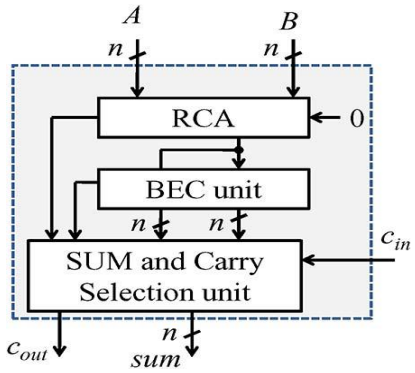


Fig. 4: BEC based CSLA.

Logic expressions:

$$S_1^1(0) = \overline{S_1^0(0)}\,;\, C_1^1(0) = S_1^0(0) \quad 2(a)$$

$$S_1^1(k) = S_1^0(k) \oplus C_1^1(k-1) \quad 2(b)$$

$$C_1^1(k) = S_1^0(k) \bullet C_1^1(k-1) \quad 2(c)$$

$$C_{out}^1 = C_1^0(n-1) \oplus C_1^1(n-1) \quad 2(d)$$

$$for 1 \le k \le n-1$$

iii. Existing CSLA design:

Existing CSLA is based on the logic formulation. It consists of one HSG unit, one FSG unit, one CG unit, and one CS unit. The CG unit is composed of two CGs (CG0 and CG1) corresponding to input-carry '0' and '1'. The HSG receives two n-bit operands (A and B) and generate half-sum word S0 and half-carry word C0 of width n bits each. Both CG0 and CG1 receive S0 and C0from the HSG unit and generate two n-bit full-carry words $C_0^1$and C_1^1corresponding to input-carry '0' and '1', respectively. The logic diagram of the HSG unit is shown in Figure: 5.The logic circuits of CG0 and CG1

are optimized to take advantage of the fixed input-carry bits.

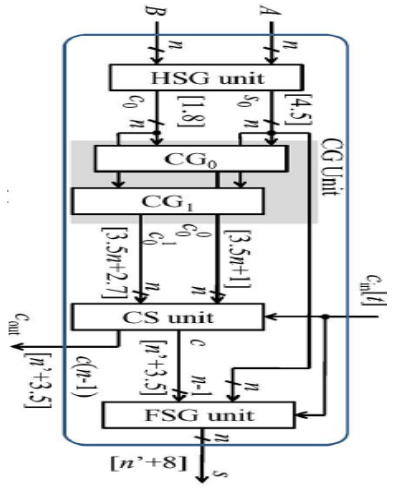


Fig. 5: Existing CSLA

Logic expressions:

$$S_0(k) = A(k) \oplus B(k)\ ; C_0(k) = A(k) \bullet B(k) \quad 3(a)$$

$$C_1^0(k) = C_1^0(k-1) \bullet S_0(k) + C_0(k)\ for C_1^0(0) = 1\ for C_1^0(0) = 0 \quad 3(b)$$

$$C_1^1(k) = C_1^0(k-1) \bullet S_0(k) + C_0(k)\ for C_1^1(0) = 1\ for C_1^1(0) = 0 \quad 3(c)$$

$$C(k) = C_1^0(k)\ when C_{in} = 0 \quad 3(d)$$

$$C(k) = C_1^1(k)\ when C_{in} = 1 \quad 3(e)$$

$$C_{out} = C(k-1) \quad 3(f)$$

$$S(0) = S_0(0) \oplus C_{in};\ S(k) = S_0(k) \oplus C(k-1) \quad 3(g)$$

From the existing logic equations and block diagram we can observe that there has been a drastic improvement in logic expressions and hence it improves the performance of CSLA by Are-Power-Delay.

iv. Proposed CSLA design:

After a deep study of the existing CSLA it is observed that there is still redundancy in the block diagram designed and the equation formed. Therefore to solve this we are analysing the logic equations and block diagram from which we can get a more significant model of CSLA.

Design of Proposed CSLA:

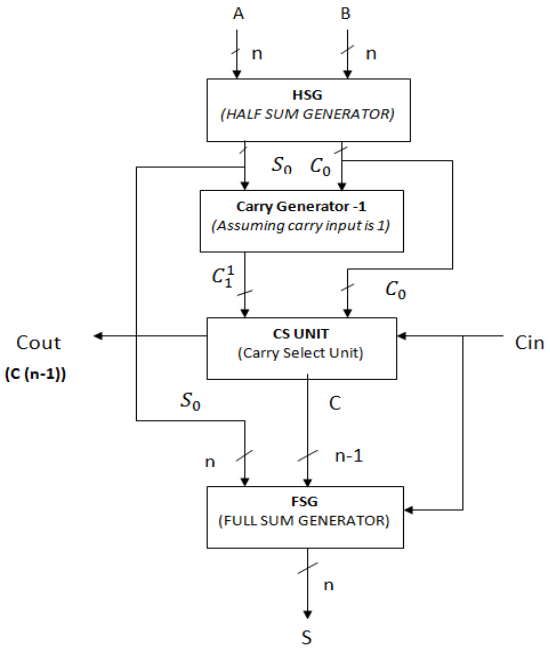


Fig. 6: Proposed CSLA

Internal blocks of proposed CSLA:

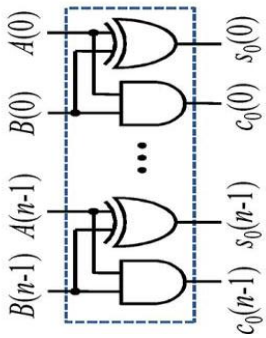


Fig 6(b): HSG Unit

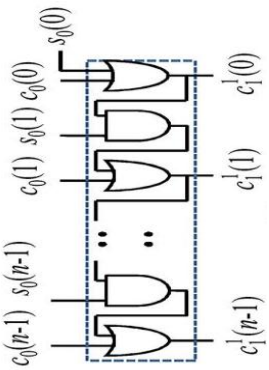


Fig 6(c): Carry generator for Cin=1

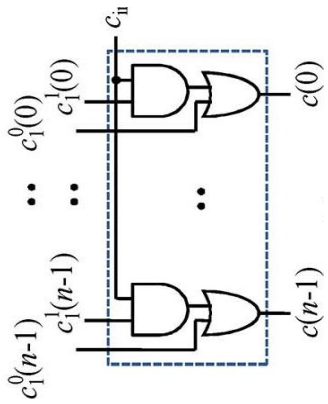


Fig 6(d): Carry Select Unit

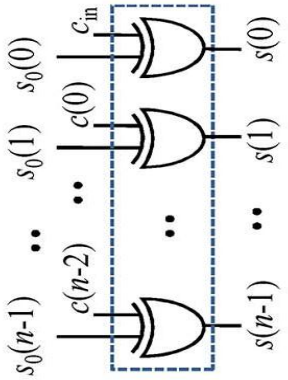


Fig 6(e): Full Sum Generator

Design function and logic expressions:

From the existing design of CSLA we can observe that the carry generate if Cin=0 block is redundant as the carry generated from HSG unit is same as the CG0 unit. In the proposed design first the addition operation is performed without input carry that is general addition operation(HSG) [figure:6(b)] is performed between the inputs A and B whose output is fed to carry generate unit which performs the operation of carry generate for input carry Cin=1, for input carry Cin=0 we can use the carry generated by the HSG unit by which we can reduce the gates used for the logic operation of carry select when carry in is 0.carry generate block is followed by the carry select unit whose inputs are carry generate output along with the input carry Cin. Therefore the output of carry of HSG unit is directly connected to the carry select unit. Carry select unit is a Multiplexer that performs the operation of MUX, where input carry acts as the selection signal. Hence the output of carry select unit is the final carry and the last bit is considered as the final carry output that is Cout. Finally the n-1 bits of carry select unit is given to FSG unit along with the sum generated from the HSG unit including the carry input Cin, The full sum generator performs the XOR operation between carry input along with carry generated from the carry select unit and the sum generated from the HSG unit whose output is the final sum.

Logic Expressions:

From the existing method of CSLA, the logic equations are modified to get the proposed CSLA design. Following equations are derived to get the proposed CSLA design.

$$S_0(k) = A(k) \oplus B(k)\ \ C_0(k) = A(k) \bullet B(k) \quad 4(a)$$

$$C_1^1(k) = C_1^1(k-1) \bullet S_0(k) + C_0(k)\ for C_1^1(0) = 1 \quad 4(b)$$

$$C(k) = C_0(k)\ when C_{in} = 0 \quad 4(c)$$

$$C(k) = C_1^1(k)\ when C_{in} = 1 \quad 4(d)$$

$$C_{out} = C(k-1) \quad 4(e)$$

$$S(0) = S_0(0) \oplus C_{in}\ \ S(k) = S_0(k) \oplus C(k-1) \quad 4(f)$$

From the above logic expressions the Proposed CSLA has been designed and proved to be more efficient in all the aspects of Area-Power Consumption–Delay.

IVRESULTS

A. Theoretical Estimations:

First we will calculate the theoretical values of Area, Delay and ADP from the proposed design along with the existing designs.

To calculate the Area and delay of the proposed design, first we need to calculate the number of gates required to design the circuit. Therefore we will convert all the gates to be designed from AND, OR, and NOT. In the design we use XOR gate that is made of 2 AND, 1 OR, 2 NOT gates that are included while calculating the area of the design.

All the theoretical values are calculated using the data sheets of SAED (Synopsys Armenia Educational Department) 90-nm cell library.

TABLE I. Area and Delay of AND, OR, NOT gates in SAED datasheet.

	AND-gate	OR-gate	NOT-gate
Area ($\mathbf{um^2}$)	7.37	7.37	6.45
Delay (ps)	180	170	100

The area and delay of design is calculated using the below equations:

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 8, August 2015

Area = $\sum$[standard area. Number of gates used] =

$$a \bullet N_a + r \bullet N_r + n \bullet N_n \quad 5(a)$$

Delay = $\sum$[critical Path. standard Delay] =

$$ta \bullet T_a + tr \bullet T_r + tn \bullet T_n \quad 5(b)$$

N-bit CSLA:

Gate count and Delay of the different carry select adders are calculated using Table 2, where (N_a, N_o, N_n) are the gate counts of AND, OR, NOT used while designing the circuit and the delays at different stages are represented by(T_{fs},T_{cout}) where T_{fs} is the delay generated at the final stage and T_{cout} is the delay at output carry.

From above Table 2, n- represents the bit width and t=0 for single stage CSLA. Also the delay and gate count of different CSLAs can be compared for different bit widths.

For n=8, 16 the Area and delay of different CSLA designs are calculated and tabulated in the Table 3. Using Table 2

Table 3: For n=8, 16 Area and delay of CSLA design

Design	Width (n)	Area ($\mathbf{um^2}$)	Delay (ns)		ADP
			f_s	C_{out}	
Conventional	8	1438.12	3.45	3.35	4.96
	16	2934.28	6.25	6.15	18.34
BEC-based	8	1282.45	4.08	4.08	5.23
	16	2587.01	6.88	6.88	17.80
Existing	8	951.09	3.87	3.42	3.68
	16	1924.30	6.67	6.22	12.83
Proposed	8	847.59	2.35	2.0	1.99
	16	1702.55	3.95	3.6	6.7

B. Simulation Results:

Different designs of CSLA are coded in VERILOG and synthesized using XILINX tool. Delay of different designs are calculated with various bit widths of n=8, 16, 32, 64,128 and tabulated in Table 4 of Existing and proposed designs. Hence the practical and theoretical values of design can be compared from Table 3 and Table 4 to conclude that desired values are obtained and there is an improvement in terms of delay and Area.

Table 2: Comparison between different CSLA designs with respect to gate count and delay.

Design (CSLA)	AND-gate ($\mathbf{N_a}$)	OR-gate ($\mathbf{N_o}$)	NOT-gate ($\mathbf{N_n}$)	Delay at final sum ($\mathbf{T_{fs}}$)	Delay at final Carry ($\mathbf{T_{cout}}$)
Conventional	14n-4	7n-3	5n-1	Max(t,3.5n+2)+4.5	Max(t,3.5n+1)+4.5
BEC-based	11n-2	5n-1	7n	Max(t,3.5n+8.3)+4.5	Max(t,3.5n+8.3)+4.5
Existing	8n-2	5n-1	4n	Max(t,3.5n+2.7)+8	Max(t,3.5n+2.7)+3.5
Proposed	7n-1	4n	4n	Max(t,2n+1.5)+6	Max(t,2n+1.5)+2.5

Table 4: Implementation results of Existing and proposed CSLA designs

Design	Width (n)	Delay (ns)
Existing	8	2.849
	16	4.613
	32	8.141
	64	15.91
	128	29.31
Proposed	8	2.292
	16	3.196
	32	5
	64	8.62
	128	15.85

C. SQRT-CSLA:

The SQRT (square root)-CSLA is designed in order to reduce the carry propagation path that reduces the final delay generated at output stage.

General design of SQRT-CSLA:

The basic SQRT-CSLA is designed using chains of CSLAs with different stages hence it is also called as multi stage CSLA. Here the first stage of design always consists of 2-bit RCA followed by increasing bit width of proposed CSLAs. That is the 16-bit CSLA can be obtained by 2-bit RCA followed by cascaded 2,3,4,5 bit proposed CSLAs and 32-bit SQRT-CSLA can be implemented by 2-bit RCA followed by cascaded 2,3,4,6,7,8 CSLAs and it can also be implemented for other bit width. The design of 16-bit SQRT-CSLA is showed in Figure 7.

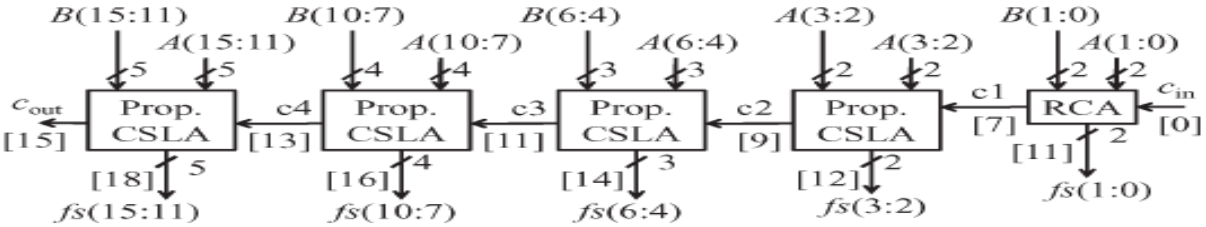


Fig 7: Design of 16-bit SQRT-CSLA.

i. Theoretical Estimations:

Theoretical values of area and delay of SQRT-CSLA are obtained by calculating the gate count and substituting them in equations 5(a) and 5(b) for different bit widths, the obtained values are tabulated in Table 5.

Table 5: Theoretical values of SQRT-CSLA with different bit width

Design	Width (n)	Delay (ns)	Area (um^2)	ADP
BEC-based	8	3.33	2287.41	7.62
	16	4.28	4853.14	20.77
	32	5.63	10006.73	43.77
Existing	8	3.0	1706.80	5.12
	16	3.85	3608.98	13.89
	32	5.27	7534.56	39.18
Proposed	8	2.45	1602.21	3.92
	16	3.36	3296.39	11.07
	32	4.62	6694.12	30.92

ii. Simulation Results:

The design of SQRT-CSLA is coded in VERILOG implemented and simulated using XILINX tool. The simulation results are tabulated in Table 6. Therefore Table 5 and Table 6 can be compared for theoretical and simulation results obtained to show that the proposed design can also be implemented for SQRT-CSLA to get good candidate for Area and Delay.

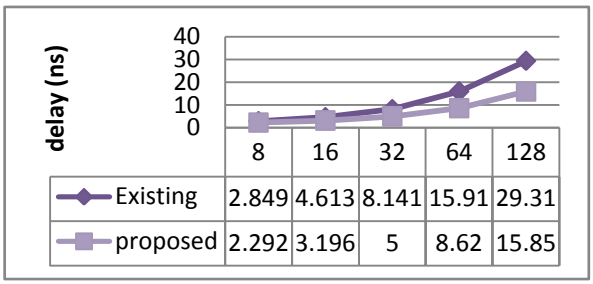
Table 6: simulation results of different SQRT-CSLA designs for n=16.

Design	width (n)	Delay (ns)
Conventional	16	5.61
BEC-based	16	5.96
Existing	16	5.55
Proposed	16	3.91

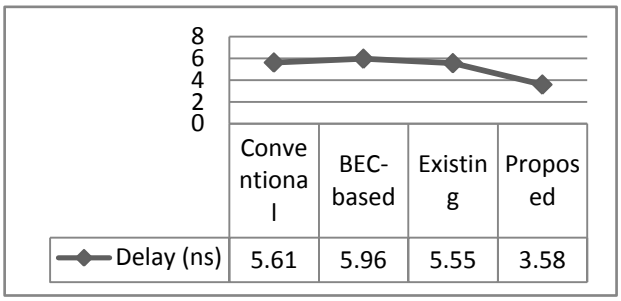
VCOMPARISION

From the proposed CSLA design the Area and Delay of design is optimized that indeed reduces the power required to operate the circuit. The values that are obtained from the simulation are tabulated and compared with existing designs of CSLAs and are represented in graphical form in graph1 for single stage CSLA with various bit widths of 8,16,32,64,128 and graph 2 for multi stage CSLA(SQRT-CSLA) of different designs.

Graph 1: comparison of delays of different CSLA designs with various bit widths.



Graph 2: Representing delays of different SQRT-CSLA design



V. Conclusion

The proposed carry select adder design has analyzed all the logic expression and dependencies involved in previous design methodologies where in there is a chance of compressing the design complexity and improving the performance. The proposed method mostly concentrates on the redundancy of logic equations that can be removed which helps in improving the performance of adder. The keen observation of output carry of HSG unit and Carry generate when Cin=0 is the main content of the proposed design based on which carry select adder with greatest improvement in Area, power consumption and delay. The experimental result of proposed method proves that the design of such carry select adder has better features than the existing ones. The design of such carry select adder can also be used to design the SQRT-CSLA for better performance results.

References

[1] Basant Kumar Mohanty, Senior Member, IEEE, and Sujit Kumar Patel IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 61, NO. 6, JUNE 2014.

[2] B. Ramkumar and Harish M Kittur," Low-Power and Area-Efficient Carry Select Adder", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, VOL. 20, NO. 2, February 2012.

[3] 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

[4] Youngjoon Kim and Lee-Sup Kim, "A low power carry select adder with reduced area", IEEE International Symposium on Circuits and Systems, vol.4, pp.218-221, May 2001

[5] Arunprasath S et al, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.2, February- 2014, pg. 181-186

[6] C.S.Manikandababu "An Efficient CSLA Architecture for VLSI Hardware Implementation" IJMIE, ISSN: 2249-0558, Volume 2, Issue 5, 2012, pp.610-622.

[7] 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.

ABOUT AUTHORS:

{1}E S BHARGAVI: born in Chittoor, India. Obtained B.Tech in Electronics and Communication Engineering from SVEW in 2013 currently pursuing Master degree in VLSI Design in VEMU institute of Technology, P.Kothakota. Interested in low power VLSI design and Digital circuits.

{2}N KIRANKUMAR: born in Chittoor, India. Working as Assistant Professor in VEMU institute of Technology, P.Kothakota. Obtained B.Tech from SVCET, Chittoor in 2006, Master degree in Embedded systems from Narayana Engineering College, Nellore in 2011. Area of interests Embedded systems and Communications.

{3} H CHANDRASEKHAR: working as Associate Professor and Coordinator in VEMU institute of Technology, P.Kothakota.

{4}L RAMAMURTHY: working as Principal and HOD for Dept. of Electronics and Communication Engineering in VEMU institute of Technology, P.Kothakota

.