High Speed Carry Select Adder for ALU Blocks

J.Rama Krishna Reddy[1], G.Rakesh Chowdary[2] and Dr. T venkata Rama Krishna[3]
[1]M.Tech VLSI, Dept of ECE, j.ramakrishnareddy@hotmail.com .
[2]Assistant Professor, Dept of ECE, KL University, rakeshchowdaryg@kluniversity.in .
[3] Professor, Dept of ECE, KL University, tottempudi@kluniversity.in

Abstract— The regular SQRT CSLA consists of two RCA blocks with carry input as 0 and 1.The Final sum will be selected from multiplexers (Mux) by the carry out generated by the pervious block. This paper, proposes an area and delay efficient carry select adder with logical reduction of excess redundant hardware. In the proposed architecture, we had implemented the RCA with carry input as 1, only with Mux, Or gate and And gate. For 16-bit regular SQRT CSLA there is a reduction of basic logic gates from 434 to 323.The delay is reduced by replacing Full-adder with half-adder in first bit of every RCA in the proposed architecture. This will reduces the number of Iterations required to get the final sum. The proposed architecture shows that there is reduction of area and delay. Based on this architecture, we designed 4-bit,8-bit,16-bit and 32-bit Square-root CSLA (SQRT CSLA) and compared with the regular SQRT CSLA. In this work, we evaluated the performance of the proposed design in 90-ηm CMOS Technology in Cadence Tools. The result analysis shows that, the proposed SQRT CSLA of 4-bit, 8-bit 16-bit and 32-bit has a reduction of 31.74%, 30.13%, 21.92% and 21.76 % respectively compared with regular SQRT CSLA in area. The delay of Proposed SQRT CSLA of 4-bit,8-bit 16-bit and 32-bit are reduce by 27.47%, 17.23%, 14.32% and 11.63% respectively.

Keywords—Application-specific integrated circuit (ASIC), Carry Select Adder (CSLA), logic reduction, redundant hardware. Ripple Carry Adder (RCA), Arithmetic Logic Unit (ALU).

I. INTRODUCTION

In Present generation in VLSI system are more concentrating in the reduction of area and power and increasing the speed of operation of the circuit. 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 RCA. One time with assumption of carry input as 0 and another time with carry as 1.After the results are calculated, the final sum and the final carry is selected is selected with the multiplexer by the previously generated carry out. The speed in the SQRT CSLA is dependent on the carry generation of the previous cascaded RCA. The sum of the each bit is generated sequentially one after the previous bit position has been summed and a carry propagated into the next position.

The CSLA is used in many electronic applications to alleviate the problem of carry propagation delay by independently generating the multiple carries and then select the carry to generate the sum[2]. However, the CSLA is not area efficient because it contains multiple Paris of cascaded RCA with input Cin=0 and Cin=1, to generate partial sum and carry, then the final sum and carry are selected by multiplexers.

In this work, we proposed an area and delay efficient carry select adder by logical reduction of excess redundant hardware[7] and reducing the number of basic logic gates. The summation of the first bit of every RCA consists of full-adder with Cin=0 and Cin=1.The Full-adder with Cin=0, is replaced with Half-Adder (XOR gate and AND gate to generate sum and carry respectively). The Full-adder with Cin=1 is replaced with compliment of the Half-Adder. The complement of Half adder is implemented by connecting NOT gate to the XOR gate of Half-adder and the carry is generated with OR gate as show in the fig1.

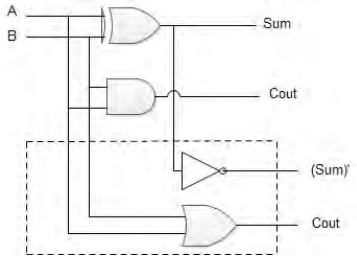


fig1. Half-adder and its complement

The summation of every second bit in the RCA consists of the Full-adders. The second block (Cin=1) of every RCA is implemented with logical reduction of the excess redundant hardware. The full-adder is replaced with Mux, OR gate, and AND gate. The Sum generated by Mux and carry is generated by AND gate and OR

gate as show in the fig 2.This is used to lower area and power consumption [5][6].

The process is explained as below. The number of iteration required to get output and number of logic gates required to implement the logic of the basic adder blocks and the logical reduction of excess redundant hardware is given below. The architecture of regular SQRT CSLA[1] has been chosen for comparison with the proposed architecture as it has less delay, low power and less area[3][4]. The number of iteration required to get the output and the number of logic gates required to implement the logic of regular SQRT CSLA and proposed SQRT CSLA are presented below.

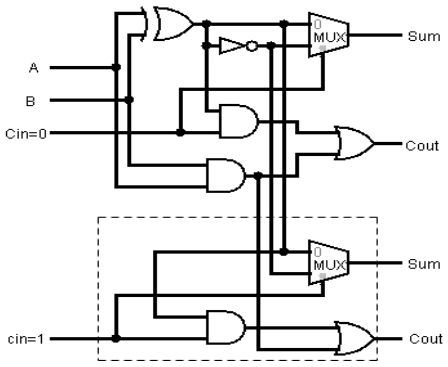


Fig2.Implementation of Modified Reduced Logic Block (MRLB)

II. IMPLEMENTATION OF BASIC ADDER BLOCKS

The SQRT CSLA consists of basic blocks such as Mux, Xor, Half-adder and Full-adder. These basic gates are used to implement the whole operation of the SQRT CSLA. The implementation of the 2:1 mux is shown in the fig3.The gates in the dotted line are operated parallel and the numeric representation of each gate indicate the area contributed by that gate. we consider all the basic gates consists of 1 unit delay and 1 unit in area. The Table I gives the details of all other basic gates.

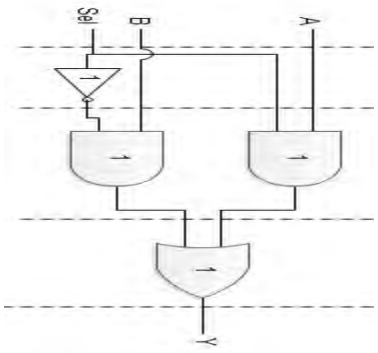


Fig.3 Calculation of required number of iterations and logic gates.

To calculate the total number of iteration required to get the output, add the number of gates in the critical path and the area is calculated by counting total number of gates for each logic block. Based on this we have calculated for XOR, Half-adder & Full-adder are listed in table I.

Table I. Number of iterations and gates required for Basic Blocks

Adder Blocks	Iterations	Basic Gates
Xor	3	5
2:1 mux	3	4
Half adder	3	6
Full adder	6	13

III. ARCHITECTURE OF REGULAR 16-BIT SQRT CSLA

The architecture of the regular SQRT CSLA is shown in the fig.4.It consists of 5 groups of different sizes of RCA.The area of each group is calculated as, the total number of gates required to implement the regular SQRT CSLA of particular group. The Table II gives the details of area of 5 groups of regular SQRT CSLA. The delay and area estimation of each group is given in fig.5. The calculation of delay is given below.

1) The Group1 has one set of 2-bit RCA. Based on consideration of delays. The carry output (c1) of the 2nd RCA arrives at [t=7] because 1st RCA generates carry at [t=5] and XOR will operate parallel with 1st RCA and 2nd RCA the sum sum1 and sum2 be generated at t=6 and t=8 respectively as shown in the fig.5(a).

2) The Group2 has two sets of 2-bit RCA. One with cin=0 and another with cin=1.The arrival of the carry output (c1) from the 1st group is [t=7].The c1 of the mux 6:3 is earlier than s3[t=8] and later than s2[t=6]. Thus, the sum3[t=11] is summation of s3 and mux[t=3], sum2[t=10] is summation of c1 and mux and carry c3[t=10] is summation of internal carry and mux as show in the fig.5(b).

3) Except for Group2, the arrival time of mux selection inputs 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, sum[6:4]} = c3[t=10]+mux

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

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

4) The one set of 2-b RCA in group2 has 2 FA for Cin =1 and the other set has 1 FA and 1 HA for Cin=0.Based on the area count of table I, the total number of gate count in group2 is determined as follows:

Gate count = 57(FA+HA+Mux)

FA = 39(3*13)

HA = 6(1*6)

Mux = 12(3*4)

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

Table II. Number of iterations and gates required for Regular SQRT CSLA

Group	Iterations	Basic Gates
Group1	7	26
Group2	11	57
Group3	13	87
Group4	16	117
Group5	19	147

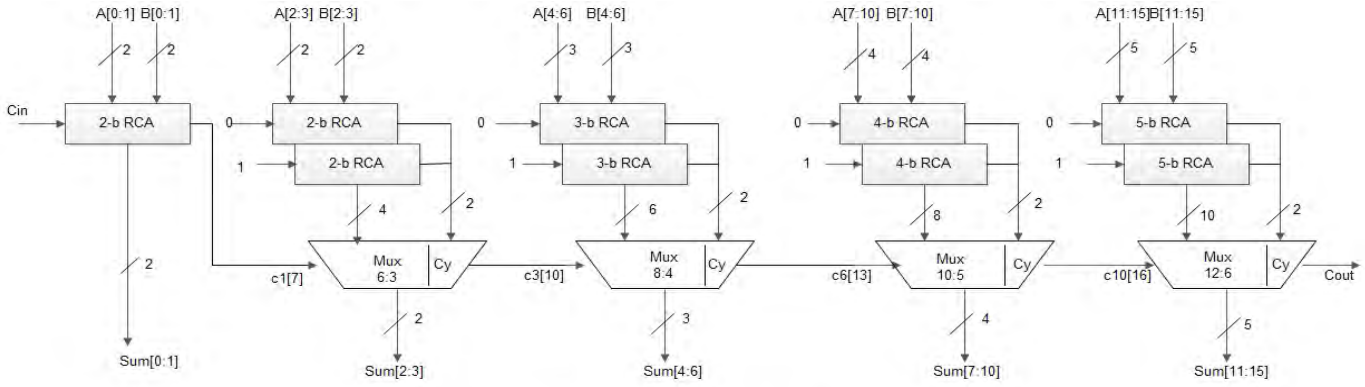


Fig4. Architecture of 16-b Regular SQRT CSLA

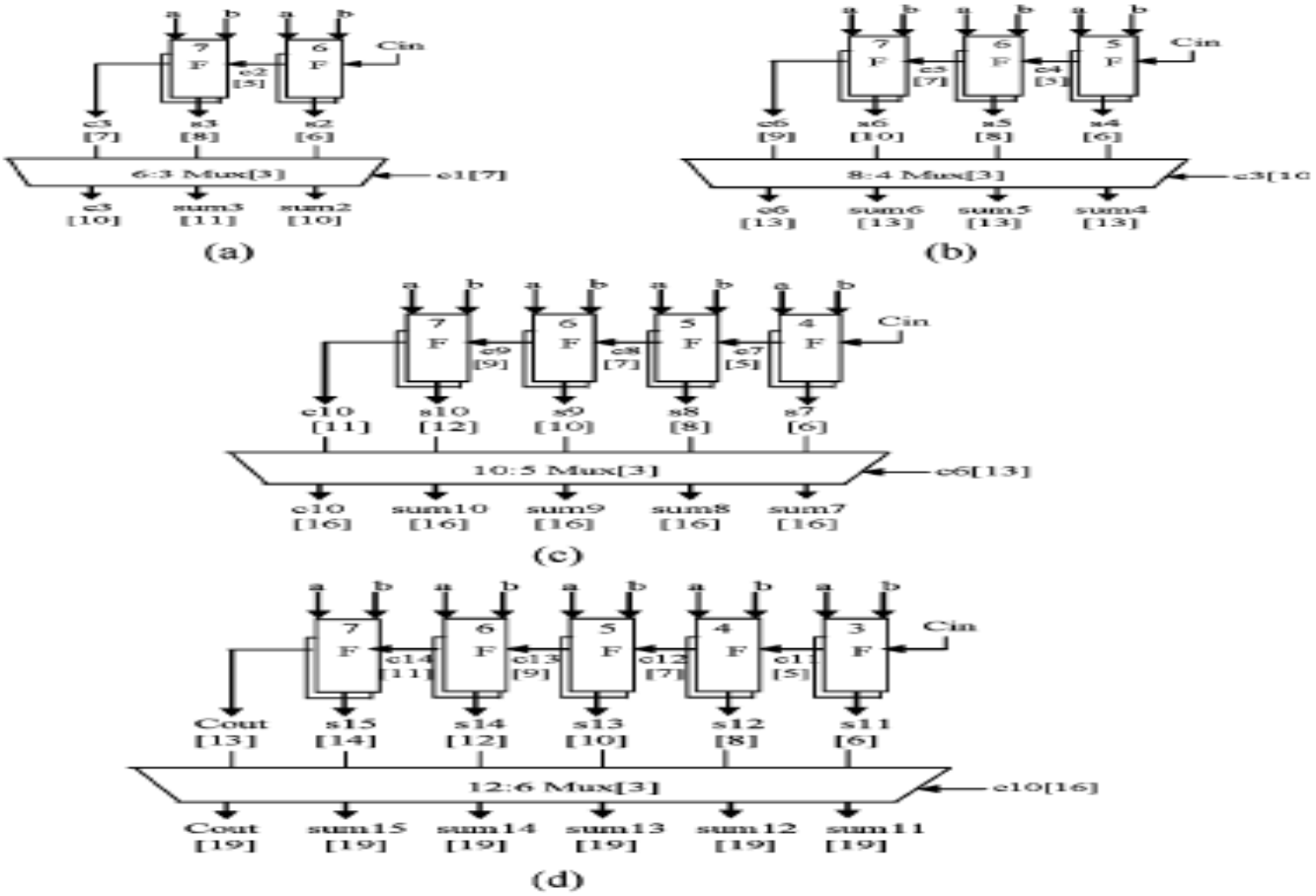


Fig.5 Calculation of required no. of iterations and logic gates of regular SQRT CSLA (a) Group2, (b) Group3, (c) Group4, (d) Group5

IV. ARCHITECTURE OF PROPOSED 16-BIT SQRT CSLA

The architecture of the proposed 16-bit SQRT CSLA using logic redundancy is show in the fig.6. It consists of five different groups of RCA. The area of the each group is calculated as, the total number of gates required to implement the each group in SQRT CSLA. The Table III gives the details of area of the 5 groups of proposed SQRT CSLA. The delay and area estimation of each group of proposed SQRT CSLA is given in fig.7. The calculation of delay is given below.

1) The Group1 consists of 1FA and 1HA. First bit of adder is implemented using half-adder (Cin=0 for addition). The Second bit is implemented using full-adder as shown in the fig.7 (a). The carry out c1 will be generated at t = 3.

2) The Group2 consists of the 2 sets of 2-bit RCA. Based on the consideration of delay. The arrival time of selection input c1[t=3] of 6:3 mux is earlier than the s2[t=4] and s3[t=6]. Thus, sum2 [t=6] is the summation of c1 and mux and sum3 [t=8] is the summation of s3 and mux (if selection input arrives earlier then the delay will be 2).The c3 [t=6] is the summation of cout[t=3] of RCA and mux is show in fig.7 (b).

3) Except for group2, the arrival time of the selection input is always greater than the output of the RCA's of different groups. Thus the delay of group3 and group5 is determined, respectively as follows:

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

$$\{c10,sum[10:7]\} = c6[t=9]+mux$$

$$\{cout,sum[15:11]\} = c10[t=12]+mux.$$

From the above calculation, for proposed SQRT CSLA we got the final carry out at t=15 and for the regular SQRT CSLA it is at t=19.which gives the reduction of 4 units in delay.

4)The one set of the 2 bit RCA in group2 has 1H and 1FA for cin = 0.For cin = 1 is implemented using the logical reduction of excess redundant hardware as sharing the hardware to implement the same operation. Based on this the area count of the proposed SQRT CSLA is reduce .The calculation of area of group2 is given below. The total gates required to implement the group2 are 39 gates.

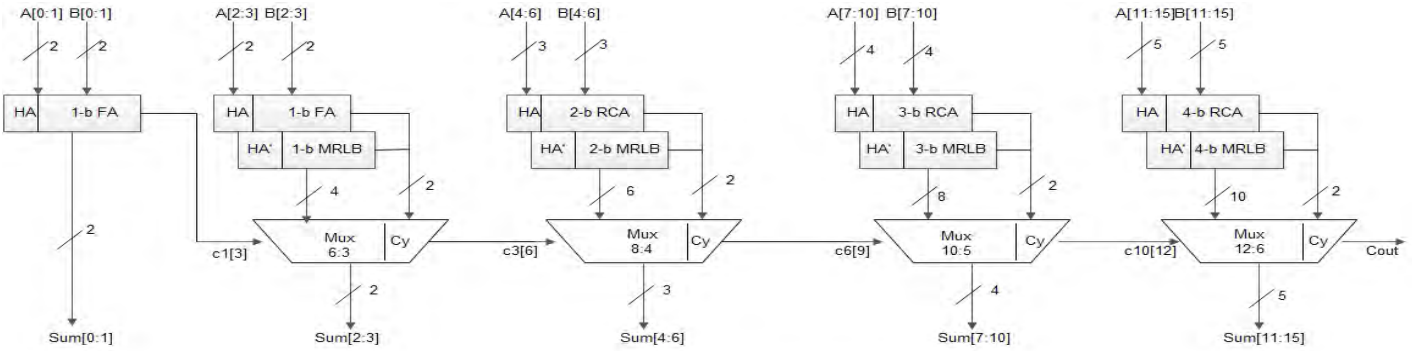


Fig6. Architecture Proposed 16-b SQRT CSLA

Gate count = 39 (Mux + XOR +INV+ AND+OR)

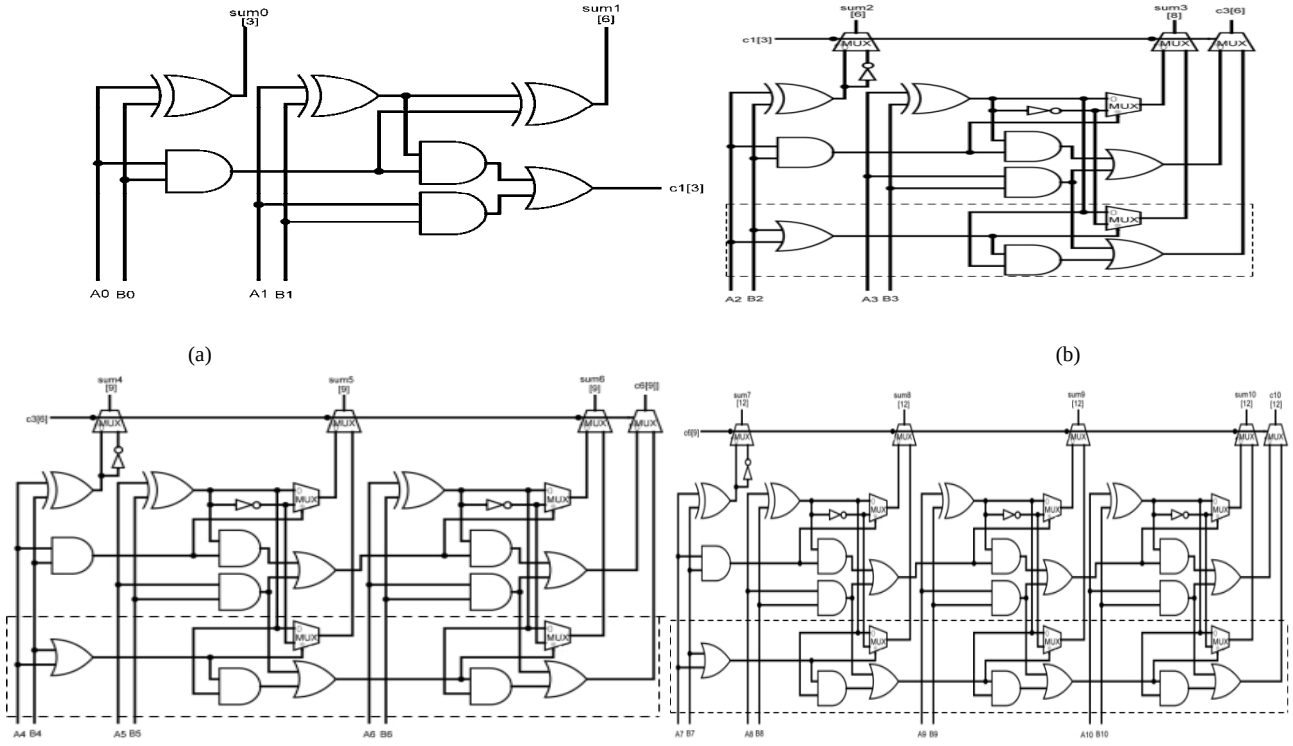
Mux = 20(5*4)

Xor = 10(2*5)

AND = 4

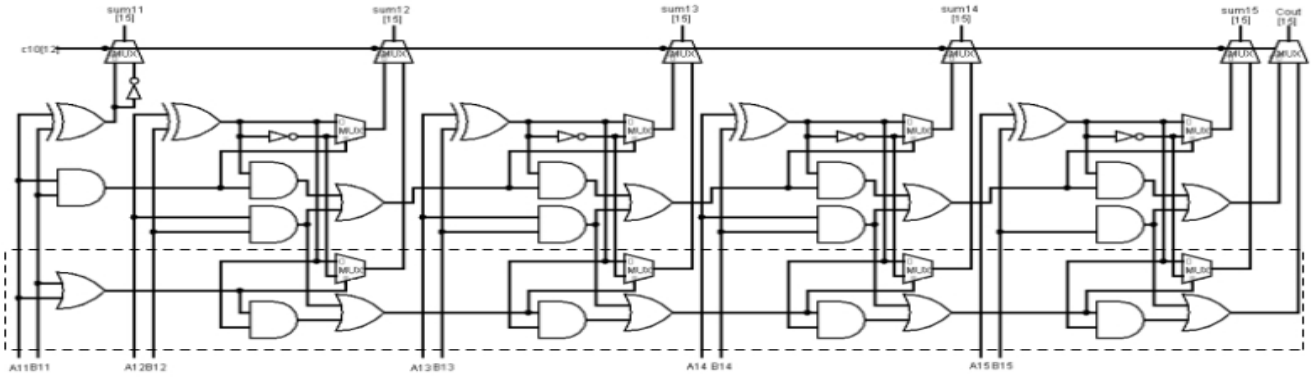
OR = 3

INV = 2.



(a) (b)

(c) (d)



(e)

Fig.7 Calculation of required no. of iterations and logic gates of regular SQRT CSLA (a) Group1, (b) Group2, (c) Group3, (d) Group4 and (e) Group5

5) Similarly, the estimated maximum delay and area of the other groups in the proposed SQRT CSLA are evaluated

and listed in the Table III.

Table III. Number of iterations and gates required for Proposed SQRT CSLA

Group	Iterations	Basic Gates
Group1	3	19
Group2	6	39
Group3	9	62
Group4	12	85
Group5	15	108

Comparing Table II and Table III, it is clear that the proposed SQRT CSLA saves 121 basic gates of areas and also 6 units decreases in gate delay than the regular SQRT CSLA. To calculate the performance of the proposed SQRT CSLA we have used Cadence, ASIC implementation and simulation.

V. ASIC IMPLEMANTION RESULTS

The design proposed in this paper has been developed using Verilog-HDL and synthesized in Cadence RTL Compiler using typical libraries of TSMC 90ηm Technology. The Sy-nthesized Verilog netlist and their respective design constrain-ts file (*.SDC*) are imported to Cadence SoC Encounter and are used to generate automated layout from standard cells and placement and routing[8].

Parasitic extraction is performed using Encounter‘s Native *RC* extraction tool and the extracted parasitic *RC* (*SPEF* format) is back annotated to Common Timing Engine in Encounter platform for static timing analysis. for each word size of the adder ,the same value changed dump(VCD) file is generated for all possible input conditions and imported the same to Cadence Encounter Power analysis to perform the power simulations. The similar design flow is followed for both the regular and proposed SQRT CSLA.

Table IV. Comparisons between Regular and Proposed SQRT CSLA

Word size	Adder	Delay (ns)	Area (μm^2)	Total Power(µW)
4 – Bit	Regular	0.364	221	6.4750
	Proposed	0.294	151	4.4270
8 - Bit	Regular	0.586	531	16.770
	Proposed	0.485	371	10.960
16 - Bit	Regular	0.768	1163	38.950
	Proposed	0.658	908	28.390
32 - Bit	Regular	1.212	2430	84.790
	Proposed	1.071	1901	65.070

Table IV, exhibits the simulation results of both the SQRT CSLA architecture in terms of delay, area and power. The area indicates the total cell area of the design and the total Power dissipation is sum of the leakage

power, internal power and switching power. The percentage reduction in the cell area, total power dissipation, delay are shown in fig7,Fig8, Fig9 respectively. it is clear that the area of the 4-bit,8-bit,16-bit.

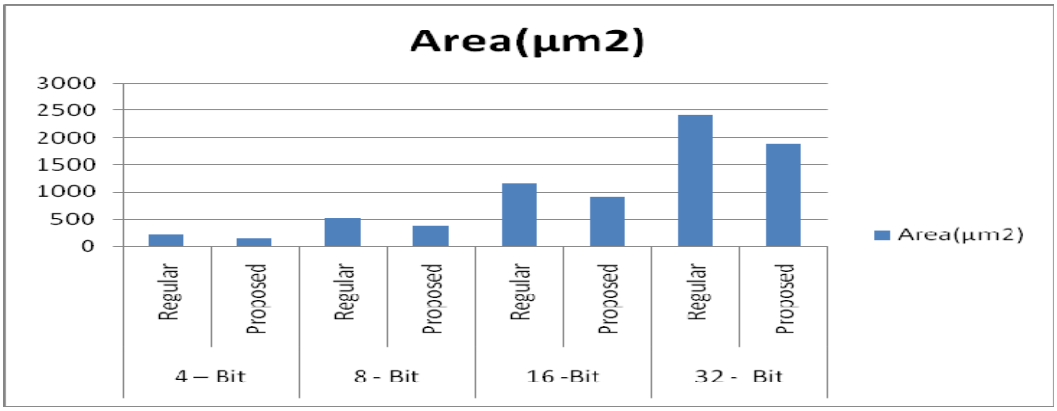


Fig7. Comparison of Area between Regular and Proposed SQRT CSLA

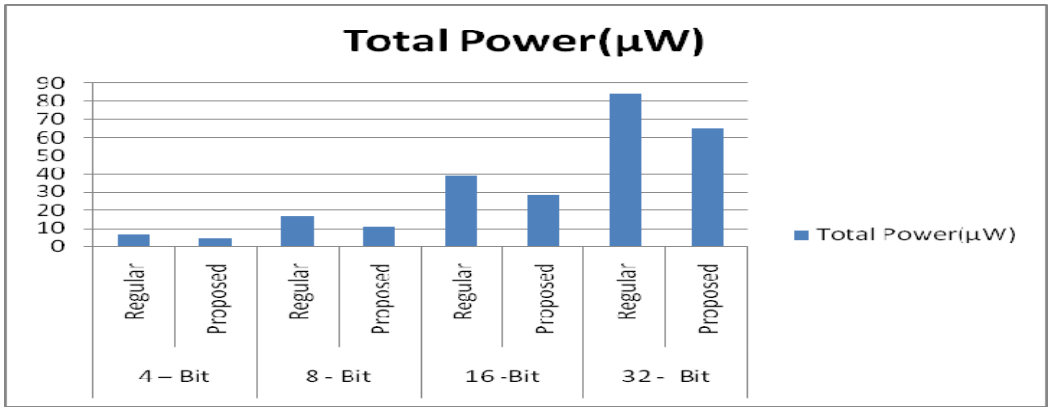


Fig8. Comparison of Power Dissipation between Regular and Proposed SQRT CSLA

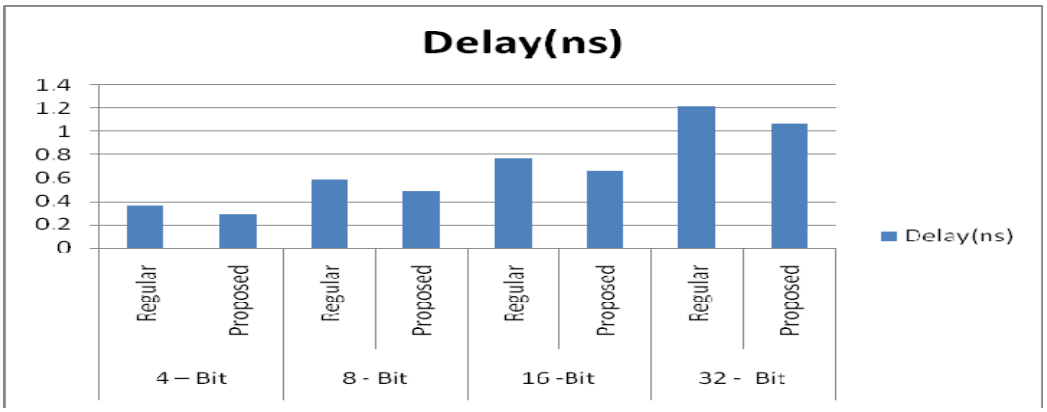


Fig9. Comparison of Delay between Regular and Proposed SQRT CSLA

and 32-bit proposed SQRT CSLA is reduced by 31.74%, 30.13%, 21.92% and 21.76 %. The total power consumed is also reduced as 31.62%, 34.64%, 27.11%, & 23.25%.The delay is reduced by 27.47%, 17.23%, 14.32% and 11.63% respectively.

VI. CONCLUSION

A simple approach is proposed in this paper to reduce the area, delay and the power dissipation of SQRT CSLA architecture .The reduced number of gates of this work offers the advantage in the reduction of area ,delay and also the total power. The compared results of the 32-bit shows that the proposed SQRT CSLA has reduction in the area and power by 21.76% and 27.11% respectively. The proposed CSLA architecture is therefore, low area, low power, less delay, simple and efficient for VLSI hardware implementation.

REFERENCES

[1] B. Ram Kumar, Harish M Kittur, "Low-Power and Area-Efficient Carry Select Adder", *IEEE transaction on very large scale integration (VLSI) systems*, vol.20, no.2, pp.371-375, Feb 2012.

[2] O. J. Bedrij, "carry-select adder", *IRE Trans. Electron. Comput*, pp.340-344, 1962.

[3] J. M. Rabaey, "Digital Integrated Circuits— A Design perspective", *New Jersey, Prentice-Hall, 2001*.

[4] 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*.

[5] 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.*

[6] T. Y. Ceiang and M. J. Hsiao,"Carry-select adder using single ripple carry adder", *Electron. Let., vol. 34, no. 22, pp. 2101-2013, Oct. 1998*.

[7] Barry W. Johnson, James H. Aylor, Haytham H. Hana, "Efficient Use of Time and Hardware Redundancy for Concurrent Error Detection in a 32-bit VLSI Adder", *IEEE journal of solid-state circuits, Vol. 23No.1, Feb 1988.*

[8] Cadence, "Encounter user guide", Version 4.1.5, May 2005.

[9] Prashant Gurjar, Rashmi Solanki, Pooja Kansliwal, and Mahendra Vucha ,"VLSI Implemntation of adders for high speed ALU", *India Conference (INDICON), 2011 Annual IEEE, vol., no., 1,6, 16-18 Dec. 2011.*