Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 89 (2016) 640 – 650

Procedia Computer Science

Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016)

# Area Efficient VLSI Architecture for Square Root Carry Select Adder using Zero Finding Logic

Bala Sindhuri Kandula[a,*], K. Padma Vasavi[b] and I. Santi Prabha[a]

[a] *Jawaharlal Nehru Technological University, Kakinada*
[b] *Shri Vishnu Engineering College for Women, Bhimavaram 534 202*

**Abstract**

Binary addition is one of the primitive operations in computer arithmetic. High performance VLSI integer adders are critical elements in general purpose and digital-signal processing processors since they are employed in the design of Arithmetic-Logic Units, in floating-point arithmetic data paths and in address generation units. Speed, delay and area are the performance parameters for any adder. Speed can be achieved by means of Square Root Carry Select Adder (SQRT CSLA). Tradeoff between those parameters plays the major role in designing new architecture. From the structure of SQRT CSLA, there is a scope to reduce area by using Zero Finding Logic (ZFC) technique. By using ZFC technique in SQRT CSLA, 16bit architecture has been developed. The modified architecture has reduced area and power when compared to SQRT CSLA Adder. The adder is implemented on Spartan 3E FPGA and is compared with SQRT CSLA. Result analysis Show that the proposed adder gives reduced memory when compared to SQRT CSLA using ZFC.


Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016

*Keywords:* And Or Inverter (AOI); Binary to Excess-one Converter (BEC); Carry Select Adder(CSLA); FPGA (Field Programmable Gate Array); Half Adder (HA); Look up Table (LUT); Ripple Carry Adder(RCA); Square Root Carry Select Adder (SQRT CSLA); Very Large Scale Integrated Circuits(VLSI); Zero Finding Logic (ZFC).

## 1. Introduction

Low power, area efficient and high performance VLSI system designs play important role in advanced digital Processors. In digital adders[1], speed of addition is limited by carry which plays the major role in computations. The sum for each bit position in a basic digital adder is generated sequentially only after the previous bit position has been summed and a carry is propagated into the next position which limits the speed of addition. N-bit Ripple Carry adder is constructed by means of N Single bit Full Adders .The computation speed of N.bit RCA is slow because output of each full adder is obtained whenever the previous carry is available.

Regular Linear Carry Select Adder is used in many data computations to alleviate the problem of delay generated by carry. It is designed by portioning the architecture into groups with Ripple Carry Adders and Multiplexers. Computations based on the input carry equal to ‘zero’ and ‘one’ for Ripple Carry adders are performed in advance and

*Corresponding author. Tel.: +91 9441343726.
*E-mail address:* k.b.sindhuri@gmail.com


Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016
doi:10.1016/j.procs.2016.06.028

the final sum and carry are selected by means of multiplexer with the carry obtained from previous group as selection line. The speed of the Regular Linear Carry Select adder is high when compared to Serial bit Adder but the area is very much high. The speed can be further improved by means of Square Root Carry Select adder[2]. SQRT CSLA architecture for 16 bit is designed by portioning it into 5 groups with different sizes of Ripple carry adders. Even though Speed is improved by using Square Root Carry Select Adder, area is high when compared to N-bit Ripple Carry Adder.

In order to overcome this problem Square Root Carry Select Adder with Binary to Excess one converter is designed[3] in which area is drastically reduced. Ripple carry Adder with input carry is equal to one is replaced by means of Binary to Excess-one converter in order to achieve low area.

Kim and Kim used a model in which area is further reduced by using an add-one circuit by replacing RCA with input carry is equal to 'one'[4] in CSLA. Binary to Excess[5,6] one is also known as add-one circuit in which RCA with input is equal to "one" is replaced by means of BEC. By using Add-one circuit the area can be drastically reduced with slight increase in area.

Parallel Prefix adders like Koggestone, Brent-Kung and Ling adders are fast when compared to Conventional and SQRT CSLA architectures. Parallel prefix adders consist of three main parts. Pre processing, carry Look ahead network and post processing are the three main parts in parallel Prefix adders. By using preprocessing stage, carry propagation speed is improved when compared to other architectures.

Koggestone adders[7–9] are generally used in high performance processors The basic blocks of this adder are the Gray Cells and the Black Cells which compute the Group Generate and Group carry Signals. This adder is very fast when compared to Linear CSLA

Till now various VLSI architectures for adders are presented. Area is very much high in many of the architectures. In order to achieve low area Square Root Carry Select Adder with Zero Finding Logic is proposed. The basic idea of this work is to use Zero Finding Logic instead of Ripple carry Adder with input Carry is equal to one and multiplexer in the Square Root Carry Select Adder to achieve low area and power consumption.

This brief is structured as follows. Section 2 deals with the proposed methodology of the basic adder blocks. Section 3 presents the Results. Comparisons of the regular and modified SQRT CSLA are presented in Section 4. Finally, the work is concluded in Section 5.

**Nomenclature**

A cin, c1, c3, c6, cout refers to carry
B Sum (1:0), sum (3:2), sum (6:4), Sum (10:7), Sum (15:7) refers to Final Sum

## 2. Proposed Methodology

### *2.1 Modified SQRT CSLA architecture using zero finding logic*

The Structure of the proposed Square Root Carry Select Adder for 16 Bit using Zero Finding Logic for Ripple Carry Adder for input Carry =1 and multiplexer to optimize the area and power is shown in Fig. 1. The architecture is partitioned into five groups.

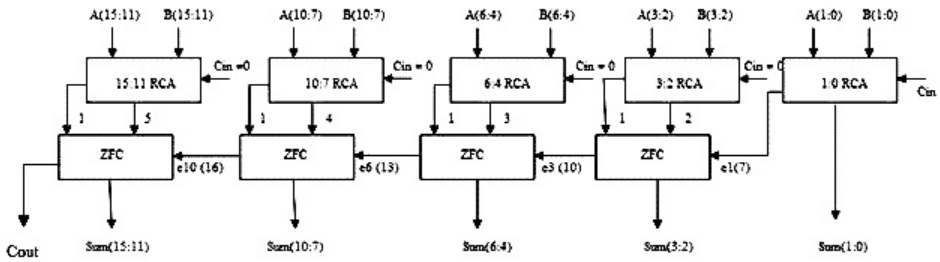


Fig. 1. SQRT CSLA Architecture using ZFC.

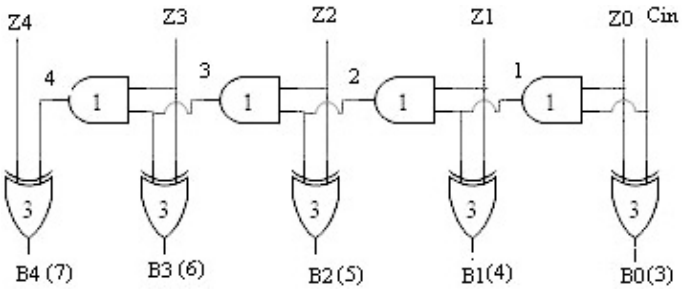


Fig. 2. Zero Finding Logic.

Group1 consists of Ripple Carry Adder for two Bit. It consists of two Full Adders. Inputs for Ripple Carry Adder are a (1:0), b (1:0), Cin. Outputs obtained from Group1 are sum (1:0) and c1. Group2 consists of Ripple Carry Adder for two bit with input carry is equal to 'zero' and Zero Finding Logic. It consists of Half Adder and Full Adder. Inputs for Ripple Carry Adder are a (3:2), b (3:2).Outputs obtained from Ripple Carry Adder are s(3:2) and c3 which are given as inputs to Zero Finding Logic .Outputs obtained from ZFC are sum(3:2) and carry3. Group3 consists of Ripple Carry Adder for three bit with input carry is equal to 'zero' and Zero Finding Logic. It consists of Half Adders and two Full Adders. Inputs for Ripple Carry Adder are a (6:4), b (6:4).Outputs obtained from Ripple Carry Adder are s(6:4) and c6 which are given as inputs to Zero Finding Logic. Outputs obtained from ZFC are sum(6:4) and carry6. Group4 consists of Ripple Carry Adder for four bit with input carry is equal to 'zero' and Zero Finding Logic. It consists of Half Adders and two Full Adders. Inputs for Ripple Carry Adder are a (10:7), b (10:7). Outputs obtained from Ripple Carry Adder are s(10:7) and c7 which are given as inputs to Zero Finding Logic. Outputs obtained from ZFC are sum(10:7) and carry7. Group5 consists of Ripple Carry Adder for five bit with input carry is equal to 'zero' and Zero Finding Logic. It consists of Half Adders and two Full Adders. Inputs for Ripple Carry Adder are a (15:11), b (15:11). Outputs obtained from Ripple Carry Adder are s(15:11) and c11 which are given as inputs to Zero Finding Logic. Outputs obtained from ZFC are sum(15:11) and Cout. Ripple

Carry Adders of different sizes with input carry is equal to 'zero' is designed by means of half adder and full adders. If '$n$' bit RCA is used in the design then a Half Adder and '$n-1$' Full Adders are used. Ripple carry adders consumes more area when compared to Zero Finding Logic and multiplexers.

### *2.2 Zero finding logic*

Zero Finding Logic is developed by means of half adders in cascaded form. The output from XOR gate is the final output where as the Carry from Half Adder is given as input for the next stage. The final XOR gate output is the carry. In the proposed Architecture zero finding logic[5–8] is used to achieve lower area instead of RCA with input carry is equal to 'one' and multiplexer in the regular CSLA. The main advantage of zero finding logic is that it uses lesser number of logic gates than the $n$-bit Full Adder (FA) structure. The structure of a 5-bit a first zero finding logic is shown in Fig. 2. By using AOI Logic, the gate count for RCA for 5 bit is sixty five units where as for multiplexer is twenty four units of area. So, the total count for this method is eight nine units of area. Similarly the total area count for Zero finding Logic is thirty five units of area.

The proposed VLSI architecture is tested for several 16 bit inputs and the obtained results are presented in the next section.

## 3. Results

The proposed architecture is implemented using Xilinx ISIM tool for simulation on a INTEL core2 (TM) Duo I2 processor, 32 bit operating System, RAM 2 GB with 2.93 GHZ clock frequency. Initially two 4 bit inputs are taken into consideration & the results are presented in Fig. 3. Modified Square Root Carry Select Adder with Zero Finding Logic for 4 bit, 8 bit, 16 bit are Simulated on Xilinx ISE 12.2.

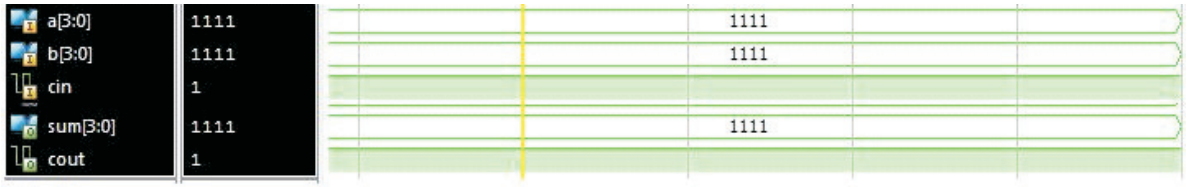


Fig. 3. Simulation Results for 4 bit SQRT CSLA using ZFC.

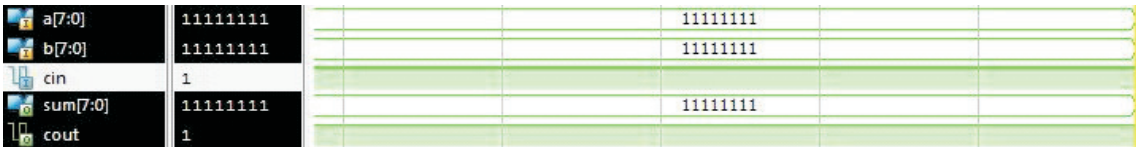


Fig. 4. Simulation Results for 8 bit SQRT CSLA using ZFC.

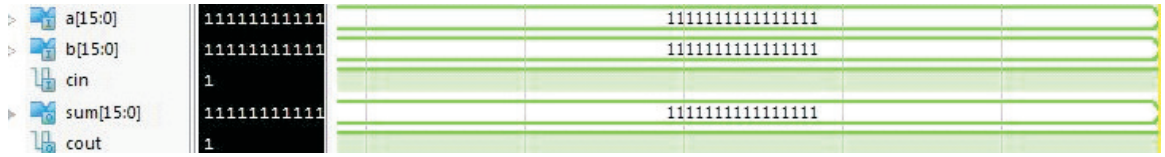


Fig. 5. Simulation Results for 16 bit SQRT CSLA using ZFC.

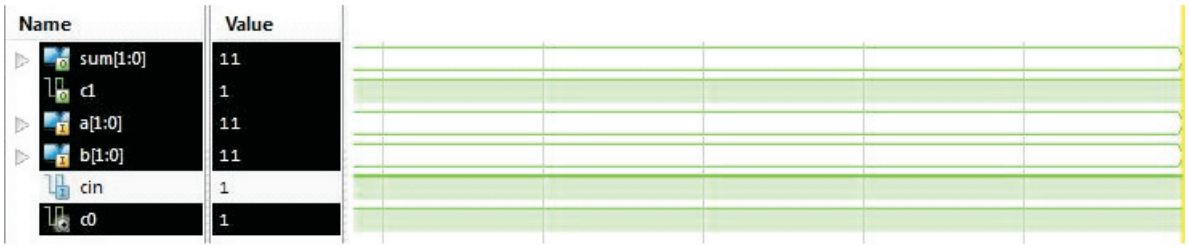


Fig. 6. Simulation Results for 2 bit Ripple Carry Adder.

The inputs for modified SQRT CSLA a(3:0) ,b(3:0) ,Cin for 8 bit are taken as a “1111”, “1000”, ‘1’ and the obtained outputs are “1000” and ‘1’.

The inputs for modified SQRT CSLA for 8 bit a(7:0) ,b(7:0) ,Cin are taken as a “11111111”, “11111111”, ‘1’ and the obtained outputs are “11111111” and ‘1’.

The inputs for modified SQRT CSLA for 16 bit a(15:0) ,b(15:0) ,Cin are taken as a “1111111111111111”, “1111111111111111”, ‘1’ and the obtained outputs are “1111111111111111” and ‘1’.

The inputs for 2bit RCA for it a(1:0) ,b(1:0) ,Cin are taken as a “11”, “11”, ‘1’ and the obtained outputs are “11” and ‘1’

After Simulation HDL Synthesis is performed, RTL and Technological Schematics for Modified SQRT CSLA Architecture for 16 bit are shown in Fig. 15

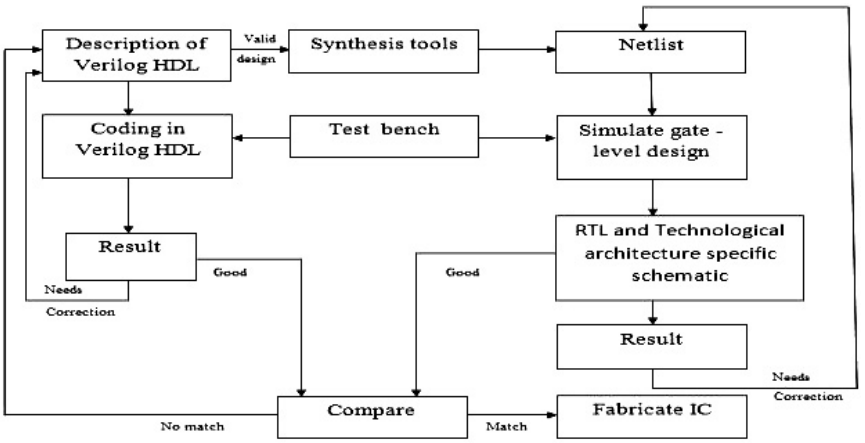


Fig. 7. Design Flow for Verifying the VLSI Architecture for 16 bit Modified SQRT CSLA.

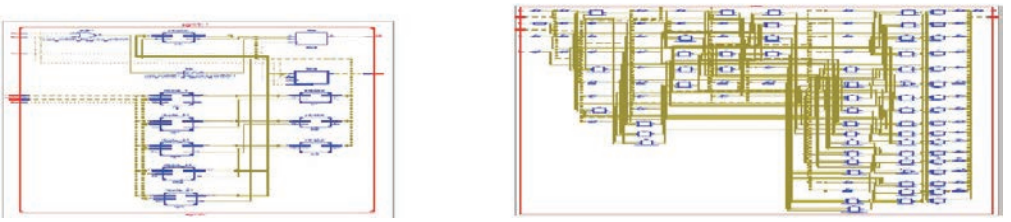

Fig. 8. RTL Schematic and Technological Schematics for Modified SQRT CSLA with ZFC Architectures.

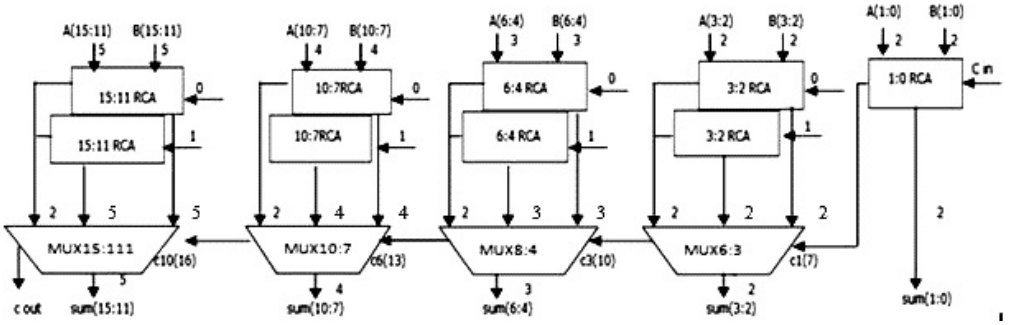


Fig. 9. Architecture for SQRT CSLA.

## 4. Comparisons

### *4.1 Delay and area evaluation methodology for SQRT CSLA for 16 bit*

1. Group1 consists of Ripple Carry Adder for two bit with input carry Cin. Arrival Time for sum0 is six units of time where as c0 is five units of time. Depending on the c0, Arrival time for sum1 is eight units of time and c1 is seven units of time.

   By using AOI Logic[5], the gate count for Group1 is Twenty Six units.

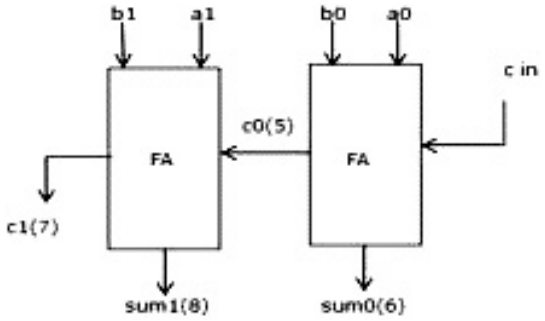


Fig. 10. Delay Evaluations for Group1.

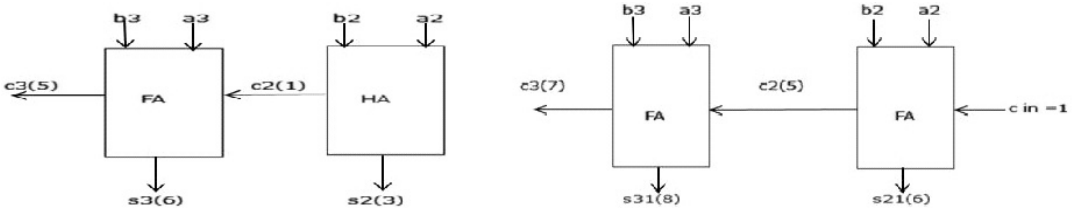


Fig. 11. Delay Evaluations for Ripple Carry Adders in Group2 with Input Carry = 1 and Carry = 0.

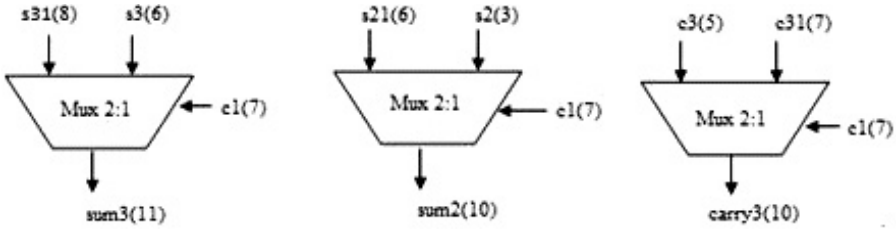


Fig. 12. Delay Evaluations for Final Sum and Carry for Group2.

2. Group2 consists of 2 sets of 2-bit RCA with Cin=0, Cin=1 and Multiplexer.
   a. Outputs from 1st set of RCA with Cin=0 are s3, s2 and c3. Arrival time for s2 is three units of time where as c(2) is one unit of time. Depending on the carry c(2), arrival time for s(3) is six units of time and c(3) is five units of time.
   b. Outputs from 2nd set of RCA with Cin=1 are S (3:2) and C(3). Arrival time for S(2) is six units of time where as C(2) is five units of time. Depending on he carry C(2), arrival time for S(3) is eight units of time and C(3) is seven units of time.
   c. Outputs from 1st set and 2nd set are given as inputs to multiplexer with input selection line as carry(1). Depending on arrival time for carry (1), Arrival times for sum(2), sum(3) and carry(3) are evaluated as ten, eleven ,ten units of time respectively.

      By using AOI Logic[5], The gate count for Group2 is 'Fifty Seven' units.

      Similarly the maximum delay for Group3 ,Group4, Group5 for SQRTCSLA[5] are 'thirteen', 'Sixteen' units , 'Nineteen' units of time, and gate Count is 'Eighty Seven', 'one hundred and fourteen', 'one hundred and forty seven' units of area.

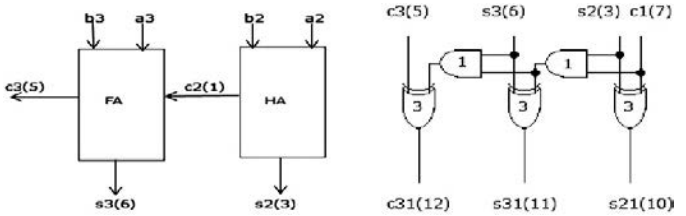


Fig. 13. Delay Evaluations for Group2.

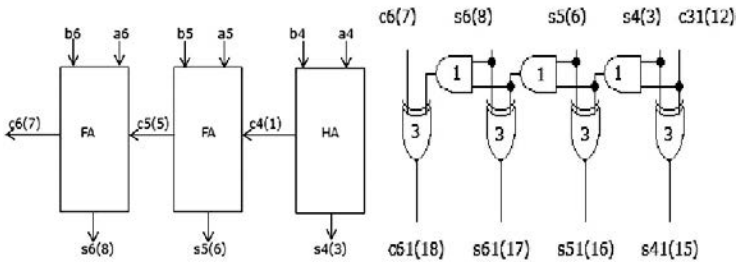


Fig. 14. Delay Evaluations for Group3.

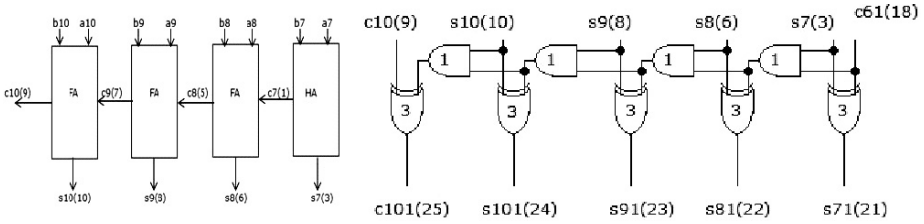


Fig. 15. Delay Evaluations for Group4.

### *4.2 Delay and area evaluation methodology for SQRT CSLA with ZFC for 16 bit*

1. The delay evaluation methodology for Group1 for SQRT CSLA using ZFC is same as SQRT CSLA.
2. Group2 consists of 1st set of 2-bit RCA with cin=0 and 2nd set of Zero finding logic
   a. Outputs from 1st set of RCA with cin=0 are s(3:2) and c(3). Arrival time for s(2) is three units of time where as c(2) is one unit of time. Depending on the carry c(2), arrival time for s(3) is six units of time and c(3) is five units of time.
   b. Outputs from 2nd set using ZFC are S(3:2) and C(3). Arrival time for S(2) is ten units of time where as C(2) is eight units of time. Depending on the carry C(2) ,arrival time for S(3) is eleven units of time and final carry C(3) is twelve units of time. The delay evaluations are shown in Fig. 13.
      By using AOI Logic[5], the area count for Group2 is 'thirty five' units
3. Group3 consists of 1st set of 3-bit RCA with cin=0 and 2*nd* set of Zero finding logic. The delay evaluations are shown in Fig. 14.
   By using AOI Logic[5], the area count for Group3 is 'fifty five' units.
4. Group4 consists of 1st set of 4-bit RCA with cin=0, and 2nd set of zero finding logic. The delay evaluations are shown in Fig. 15.
   By using AOI Logic[5], the area count for Group4 is 'Seventy four' units.

Table 1. Delay, Area Theoretical Evaluations for SQRT CSLA and SQRT CSLA with ZFC.

| | SQRTCSLA | | | SQRTCSLA with ZFC | | |
|---|---|---|---|---|---|---|
| Group | Area | Maximum Delay | | Area | Maximum Delay | |
| | | Sum | Carry | | Sum | Carry |
| Group1 | 26 | 8 | 7 | 26 | 8 | 7 |
| Group2 | 57 | 11 | 10 | 35 | 11 | 12 |
| Group3 | 87 | 13 | 13 | 55 | 17 | 18 |
| Group4 | 117 | 16 | 16 | 74 | 24 | 25 |
| Group5 | 147 | 19 | 16 | 93 | 32 | 33 |

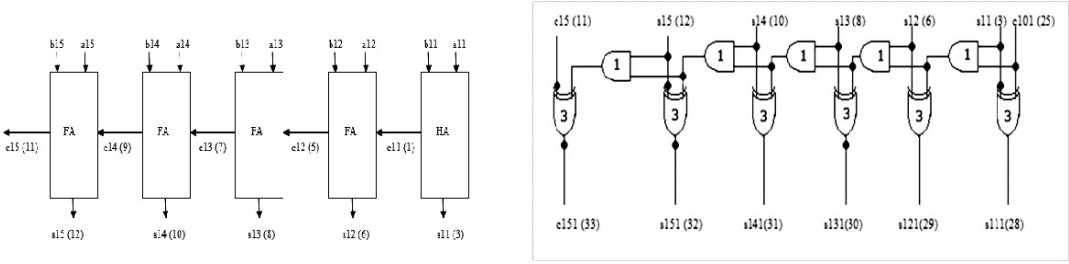


Fig. 16. Delay Evaluations for Group5.

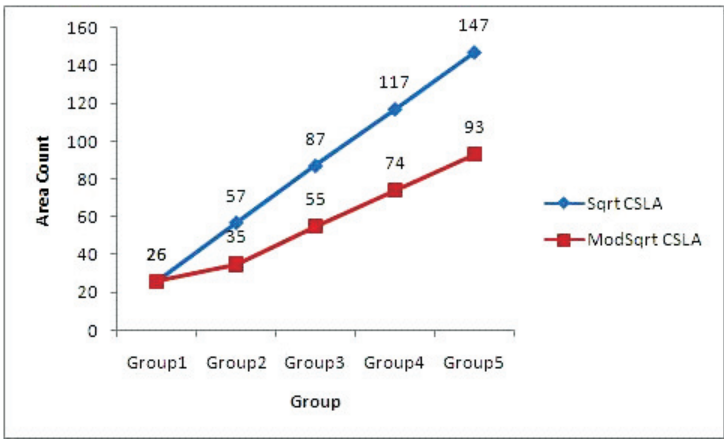


Fig. 17. Comparison of Area Count for Conventional and Modified SQRTCSLA.

5. Group5 consists of 1st set of 5-bit RCA with cin=0 and 2nd set of Zero finding logic. The delay evaluations are shown in Fig. 16.

   By using AOI Logic[5], the area count for Group4 is 'Seventy four' units.

From the Table 1, Area count for Group2 is thirty five units where as for conventional CSLA the area count is fifty seven units. It is clearly observed that the area count is less by twenty two units. Area count for Group3 is fifty five units where as for conventional CSLA the area count is eight seven units. It is clearly observed that the area count is less by thirty two units. Area count for Group4 is seventy four units where as for conventional CSLA the area count is one hundred and seventeen units. It is clearly observed that the area count is less by forty three units. Area count for Group5 is ninety three units where as for conventional CSLA the area count is one hundred and forty seven units. It is

Table 2. Comparison of Parameters for SQRT CSLA and SQRT CSLA with ZFC.

| Word Size | Adder | Memory (KB) | Frequency (MHZ) | Power (mW) | | |
|---|---|---|---|---|---|---|
| | | | | Total | Dynamic | Quiescent |
| 4 Bit | SQRT CSLA | 186124 | 90.547 | 82.79 | 1.79 | 81.00 |
| | Modified SQRT CSLA | 185612 | 80.873 | 82.95 | 1.95 | 81.01 |
| 8 Bit | SQRT CSLA | 186316 | 85.063 | 82.93 | 1.92 | 81.01 |
| | Modified SQRT CSLA | 185932 | 79.643 | 83.03 | 2.02 | 81.01 |
| 16 Bit | SQRT CSLA | 190732 | 79.891 | 83.54 | 2.53 | 81.01 |
| | Modified SQRT CSLA | 187660 | 77.688 | 83.27 | 2.26 | 81.01 |

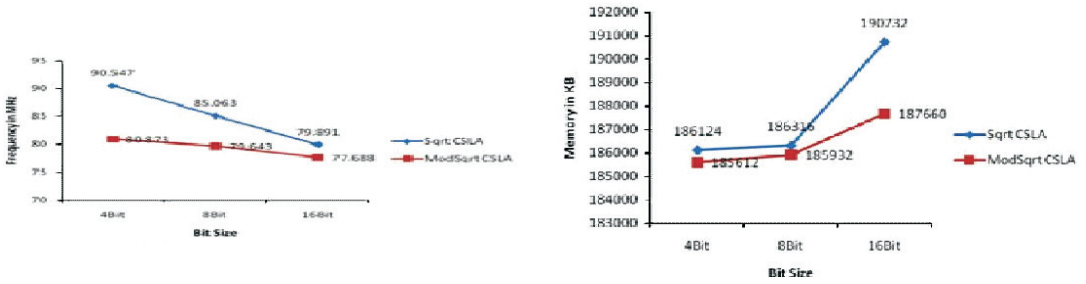


Fig. 18. Comparison of Frequency and Memory for SQRT CSLA and SQRT CSLA using ZFC.

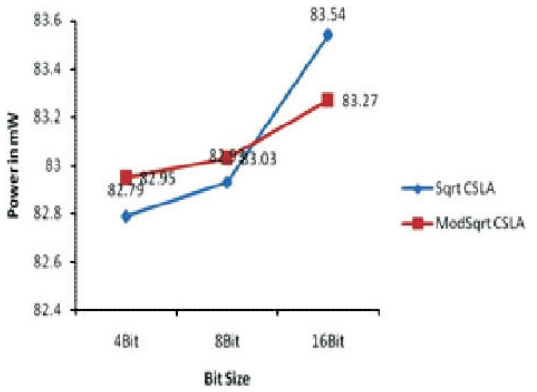


Fig. 19. Comparison of Power for SQRT CSLA and SQRT CSLA using ZFC.

clearly observed that the area count is decreased by fifty four units. As the group Size increases, the difference in area count for modified & Conventional CSLA Architectures is also increased.

From the Fig. 17, it is observed that as the Group Size increases, the difference in area count is increasing. As the bit size increases the steepness of the curve is also very sharp and the efficiency in terms of area count is also very high.

From the Fig. 18, it is observed that as the Bit Size increases, the steepness of the curve for modified SQRTCSLA is increased when compared to conventional SQRTCSLA. Similarly, as the Bit Size increases, the frequency of the modified architecture is also decreasing when compared to conventional SQRTCSLA.

From the Fig. 19, it is observed that as the Bit Size increases, the steepness of the curve for the power consumed by the modified SQRTCSLA is increased when compared to conventional SQRTCSLA.

| Device Utilization Summary (estimated values) | | |
|---|---|---|
| **Logic Utilization** | **Used** | **Available** |
| Number of Slices | 5 | 4656 |
| Number of 4 input LUTs | 9 | 9312 |
| Number of bonded IOBs | 17 | 232 |
| Number of GCLKs | 1 | 24 |

Fig. 20. Device Utilization Summary for Conventional SQRT CSLA for 4 bit.

| Device Utilization Summary (estimated values) | | |
|---|---|---|
| **Logic Utilization** | **Used** | **Available** |
| Number of Slices | 4 | 4656 |
| Number of 4 input LUTs | 8 | 9312 |
| Number of bonded IOBs | 17 | 232 |
| Number of GCLKs | 1 | 24 |

Fig. 21. Device Utilization Summary for SQRT CSLA using ZFC for 4 bit.

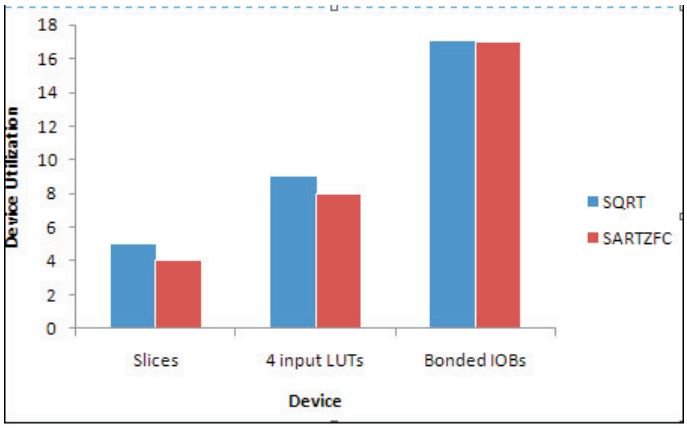


Fig. 22. Comparison of Device Utilization Summary for Conventional SQRT CSLA & SQRT CSLA using ZFC for 4 bit.

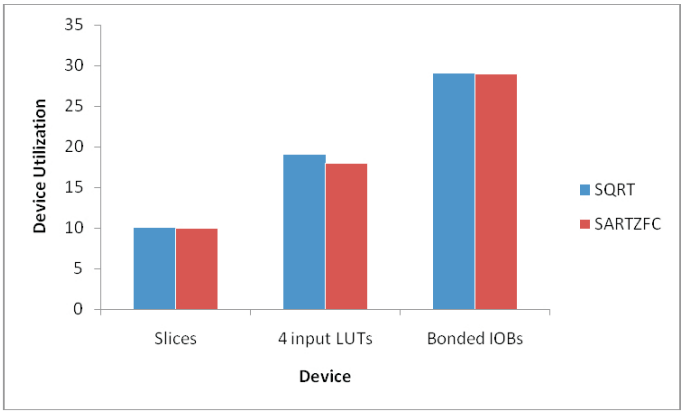


Fig. 23. Comparison of Device Utilization Summary for Conventional SQRT CSLA & SQRT CSLA using ZFC for 8 bit.

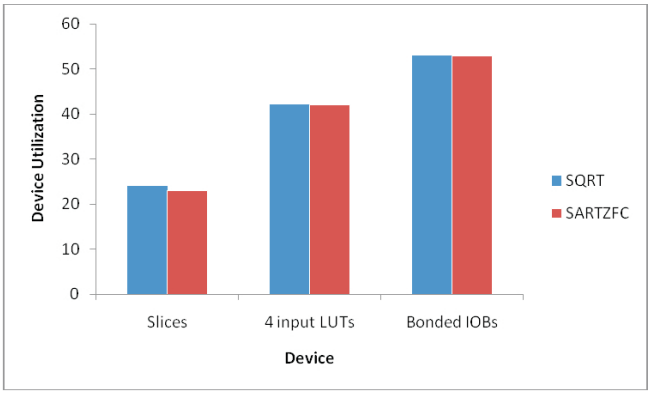


Fig. 24. Comparison of Device Utilization Summary for Conventional SQRT CSLA & SQRT CSLA using ZFC for 16 bit.

From the Fig. 22, it is observed that no. of slices are 4 for modified architecture where as no. of slices are five for conventional CSLA.Similarly it is observed that no. of bonded IOBS are 8 for modified architecture where as no. of bonded IOBS are 9 for conventional CSLA.

From the Fig. 23 & Fig. 24, it is observed that no. of slices are less for modified architecture when compared to conventional SQRT CSLA. The work is carried on Xilinx ISE12.2.

## 5. Conclusions

In this paper, Modified SQRT CSLA with ZFC architecture is designed to reduce the memory. The results show that the memory in the architecture of modified SQRTCSLA is less by 1.61% than the conventional SQRTCSLA as word size is increasing from 4 bit to 16 bit. The power consumed by the proposed architecture is also less by 0.32% when compared to Conventional SQRT CSLA. The frequency for the proposed architecture is also less by 2.75% when compared to Conventional CSLA. The no. of LUTS are also less for modified architecture when compared to conventional SQRT CSLA. The modified and conventional adders are simulated by using Verilog HDL and implemented on Spartan XC3S500E FPGA device.

## References

[1] O. J. Bedrij, Carry Select Adder, *IRE Transactions on Electronics Computer*, pp. 340–344, (1962).
[2] B. Ramkumar, H. M. Kittur and P. M. Kannan, ASIC Implementation of Modified Faster Carry Save Adder, *European Journal on Scientific Research*, 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. Letter*, vol. 37, no. 10, pp. 614–615, May (2001).
[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] B. Ramkumar and Harish M. Kittur, Low-Power and Area-Efficient Carry Select Adder, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 371–375, vol. 20, no. 2, February (2012).
[6] Basant Kumar Mohanty and Sujit Kumar Patel, Area–Delay-Power Efficient Carry Select Adder, *IEEE Transactions on Circuits and Systems-II:Express Briefs*, vol. 61, no. 6, June (2014).
[7] U. Sajesh Kumar, K. Mohamed Salih and K. Sajith, Deisgn and Implementation of Carry Select Adder without using Mutilpexers, 1$^{st}$ *International Conference on Emerging Technology in Electronics, Communication and Networking*, (2012)
[8] K. Bala Sindhuri, K. Padma Vasavi, I. Santi Prabha and N. Udaya Kumar, VLSI Architecture for Linear Carry Select Adder with Zero Finding Logic, 6$^{th}$ *International Advanced Computing Conference IACC 2016*, pp. 31, February (2016).
[9] B. Tapasvi, K. Bala Sindhuri, I. Chaitanya Varma and N. Udaya Kumar, Implementation of 64 Bit KoggeStone Carry Select Adder with ZFC For Efficient Area, *2015 IEEE International Conference on Electrical, Computer & Communication Technology*, SVS College of Engineerng, Coimbatore, 5–7 March (2015),