FPGA Implementation of Area Efficient and Delay Optimized 32-Bit SQRT CSLA with First Addition Logic

Sumeet D. Gandhe
Research Scholar
Department of EE
JDCOEM Nagpur-441501,India

Venkatesh Giripunje
Department of ECE
PCOE
Nagpur-440019, India

ABSTRACT
Binary addition is a fundamental operation in most digital circuits. There are a variety of adders, each has certain performance. Parallel adders are used for fast binary addition which plays a crucial impersonation in majority of digital circuit designs including digital signal processors (DSP) and microprocessor data path unit. In this project we implement the parallel adder 32-bit SQRT CSLA(square root carry select adder) with BEC logic and compare them to the simple ripple carry adder(RCA),carry look ahead adder(CLA) & carry select adder(CSLA).

To reduce the area and delay of n-bit CSLA, n-bit SQRT CSLA with BEC-1 logic was designed. Since CSLA is area consuming because it consists of dual RCA in structure. From the structure of the CSLA, it is clear that there is scope for reducing the area and power consumption in the CSLA. So as to achieve the same an efficient gate-level modification would be implemented to vitally reduce the delay of the CSLA. 32-bit SQRT CSLA with BEC-1logic is already designed which reduces the area of corresponding CSLA. But while performing area reduction and power reduction it is found that there is an increase in the delay. So in this project we put forward the structure of square root CSLA (SQRT CSLA) using "first addition logic circuit(FAL)" for 32-bit instead of BEC-1 logic circuit. The performance in terms of area, delay and utilization of power are calculated for SQRT CSLA with FAL and are compared with existing CSLA, SQRT CSLA and SQRT CSLA with BEC-1 logic.

Keywords
Field programmable gate array (FPGA), First addition logic (FAL), Ripple carry Adder (RCA).

1. INTRODUCTION
Reduced area and high speed data path logic systems are the main areas of research in VLSI system design. High-performance processors and systems always demands swift addition and multiplication. Time required to propagate the carry through an adder is the basic limitation in the pullup of speed of digital adder. Each sum bit generates only after the previous bit position has been summed also depends on carry generation of lower significant bit position. Ripple Carry Adder, Carry Look Ahead Adder, Carry Save Adder, Carry Skip Adder which have its own advantages and disadvantages. 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 immensely.

Basic gates like AND, OR, NOR, NAND etc. are used in digital computers to perform basic arithmetic operations like addition, multiplication, division. Among all the arithmetic operations if we can implement addition then it is easy to perform multiplication, subtraction or division.

There are many types of digital adders are available for designing a digital circuits and arithmetic units in a processor. The performance of digital adders is limited by the speed of addition. The reason behind this limitation is the time taken to propagate the carry. There are many ways to design an Adder. The Ripple Carry Adder (RCA) structure is chain of Full adders which is easy to design, but takes longer time to perform addition operation due to the delay in propagation of carry from one adder to another. The delay due to carry propagation in RCA is proportional to the number of input bits (N) to RCA. For large values of N, the delay of the RCA also increases. To overcome delay problem, a new adder structure is designed called Carry Look-Ahead Adder (CLA).

CLA is designed using two blocks namely propagate and generate block. As the number of input bits increases, the size of propagate and generate blocks also increases which causes increase in area and also introduces delay again. So CLA avoids the delay problem for less number of input bits, but not suitable for large size input. The CSLA provides a Compromise between RCA and CLA.

The CSLA (carry select adder) is implemented in many computational systems to mitigate the problem of carry propagation delay by liberally generating numerous carries and then select a carry to generate the sum. In CSLA, there are numerous pairs of ripple carry adder(RCA) to generate partial sum and carry by assuming carry input Cin=0 and Cin=1,afterwards the multiplexers are used for the selection of final sum and carry owing to which it is area inefficient. Delay of 16-bit RCA=80 input gate delay. Delay of 16-bit CLA=42 input gate delay. Delay of 16-bit CSLA=29 input gate delay.(uniform sized)Usually a variable sized CSLA is used to reduce the input gate of first RCA block. Delay of 16-bit SQRT CSLA=22 input gate delay (variable sized).

2. SQUARE ROOT CARRY SELECT ADDER

2.1 Structure and working of SQRT CSLA
Figure 1 is the basic building block of a carry-select adder, where the block size is 4. Two 4-bit ripple carry adders are multiplexed together, where the resulting carry and sum bits are selected by the carry-in. Since one ripple carry adder assumes a carry-in of 0, and the other assumes a carry-in of 1, selecting which adder had the correct assumption via the actual carry-in yields the desired result.

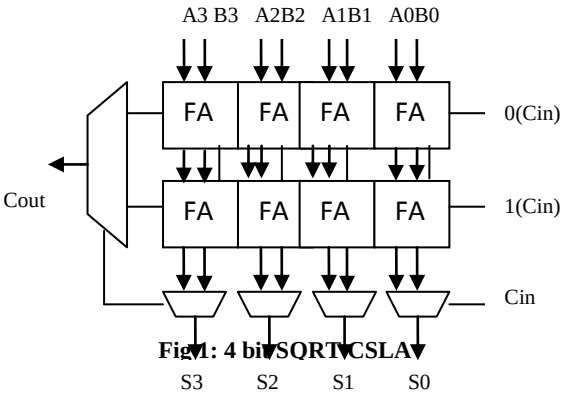


Fig 1: 4 bit SQRT CSLA

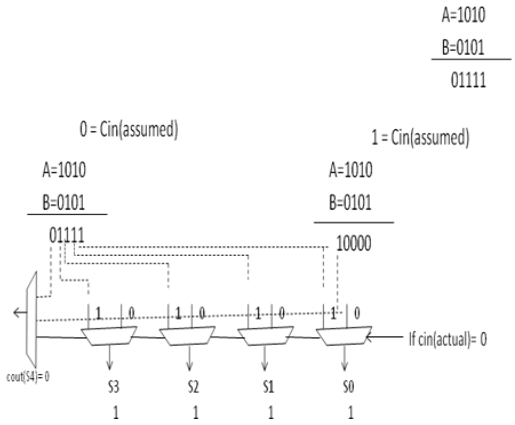


Fig 2: Illustration of 4 bit SQRT CSLA

There are two types of Carry Select Adders:

1. Uniform sized CSLA
2. Variable sized CSLA

In uniform Carry Select Adder each block size is fixed in all stages, but in variable Carry Select Adder block size is variable. The delay at Cin input stage can be reduced using variable type of CSLA.

3. VARIABLE SIZED 32-BIT SQRT CSLA

3.1 Structure and working of 32 bit SQRT CSLA

A 32-bit carry-select adder with a variable block size of 8,7,6,4,3,2,2 can be created with a n-bit ripple carry adder. Owing to the knowledge of carry-in at the initiation of computation, a carry select block is not needed for the first two bits.

A 32-bit CSLA with variable size can reduce the input gate delay of first RCA block. Here we show an adder with block sizes of 2-2-3-4-6-7-8. When the full-adder delay is equivalent to the multiplexer (MUX) delay the concept of breaking compact 32-bit CSLA module into blocks is helpful. The CSLA is used in many digital systems design to overcome the problem of carry propagation delay by independently performing addition operation by considering carry inputs (Cin) as 1 and 0.

Figure 3 shows a 32-bit SQRT CSLA. The SQRT CSLA is divided into m=√2m carry select stages (CSS), where m is number of input bits. The 32 bit SQRT CSLA consists of 7 CSS. The CSS consists of two ripple carry adders one with carry in 0 and other with carry in 1. It also consists of a multiplexer which is used to select the sum and carry values from the two RCAs by using the control signal to it. The control signal to multiplexer is nothing but the carry out of the previous CSS. If the control signal is 1 then sum and carry out of RCA with Cin=1 is selected by the multiplexer and if control signal is 0 then sum and carry out of RCA with Cin=0 is selected by the multiplexer.

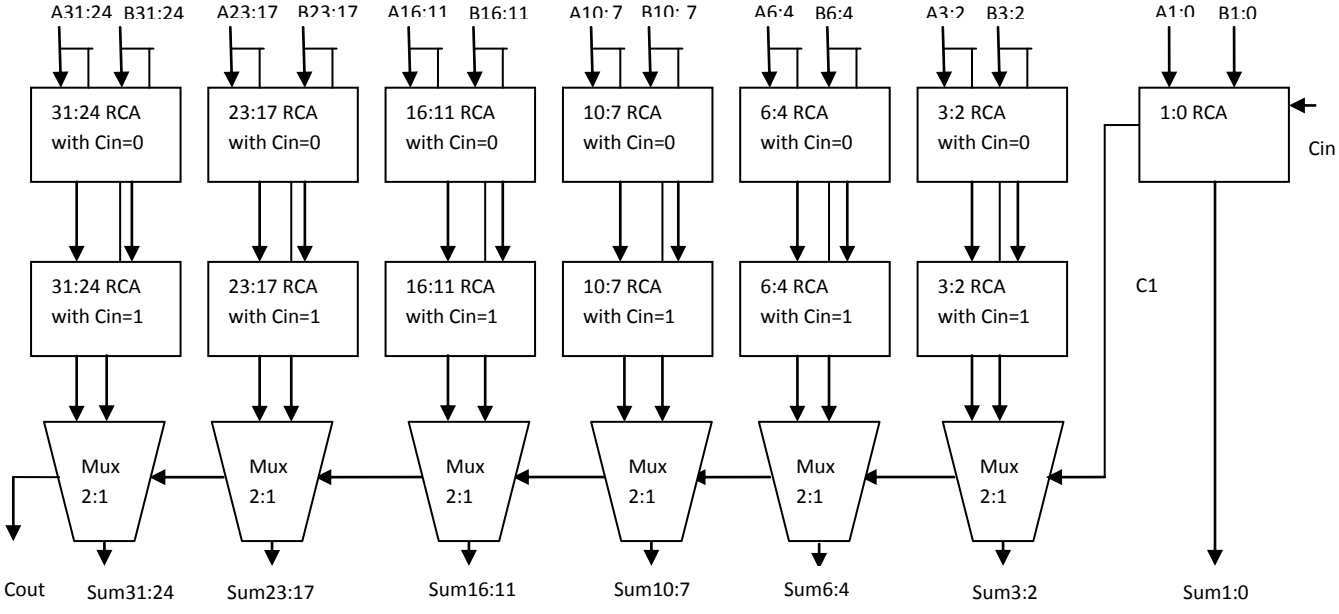


Fig 3: Variable sized 32 bit SQRT CSLA

3.2 Delay & area evaluation of basic modules in CSLA

Basic modules in CSLA: RCA (i.e. HA & FA) & MUX. Consider all gates to be made up of And OR INVERTER (AOI logic). AND OR & INVERTER each having a delay of 1.

Table 1. Delay and area evaluation of basic modules of CSLA.

Module	Delay	Area
XOR	3	5
2:1 MUX	3	4
HA	3	6
FA	6	13

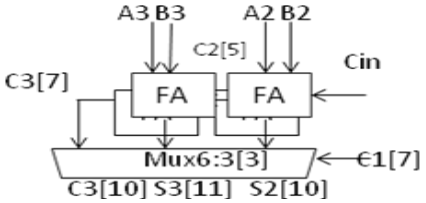


Fig 4: Group 2

As shown in fig.4, group2 consists of a 2-bit RCA with Cin=0, a 2-bit RCA with Cin=1 and also a 6:3 MUX. The 2-bit RCA with Cin=0 consists of one FA and one HA where as 2-bit RCA with Cin=1 consists of two FAs. Based on the numerals in column of area as in table-1, the total number of gates present in group2 is:

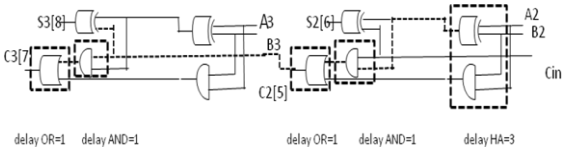


Fig 5: Group 2 in AOI logic

Delay of C3 [7] = delay of c3 +mux delay=7+3=10

Delay of sum2 [10] = delay of c1 +mux delay=7+3=10

Delay of sum3 [11] = delay of s3 +mux delay=8+3=11

The total number of gates present in the remaining groups can be calculated in the same way as for group 2. The maximum delay can be calculated by adding the delays of gates in the longest path in the architecture that contributes the maximum delay. In this adder, dual RCAs are used which occupies more area which in turn increases the power consumption. So a new adder is proposed is SQRT CSLA with Binary to Excess-1 Converter (BEC).

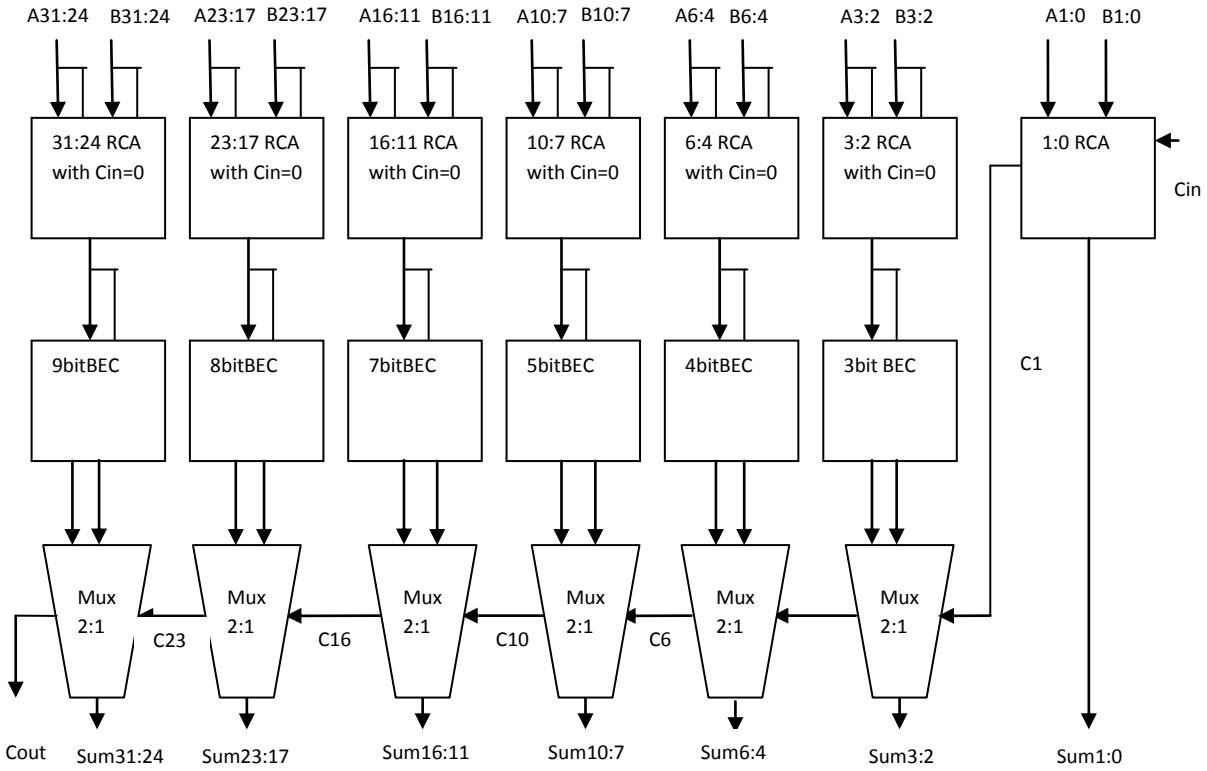


Fig 6: Variable sized 32 bit SQRT CSLA with BEC

4. VARIABLE SIZED 32-BIT SQRT CSLA WITH BEC

4.1 Structure and working of 32 bit SQRT CSLA with BEC

For the reduction in area and power consumption of the traditional CSLA, RCA with Cin =1 is substituted with BEC (Binary to Excess-1 Converter) as shown in figure 6. An n-bit RCA can be replaced with a n+1-bit BEC. Figure 7, explains the basic function of the CSLA by using the 6-bit BEC together with the multiplexer. A set of six bits (6-bit input) and the other set of 6- bits (6-bit BEC output) were given as input to the 12:6 multiplexer. Depending on the control signal Cin, multiplexer selects either the BEC output or the 6-bit input. The advantage of the BEC logic in SQRT CSLA is that, as the number of input bits is increased the requirement of area is progressively decreased. Figure 6, shows the structure of a 32-bit SQRT CSLA with BEC.

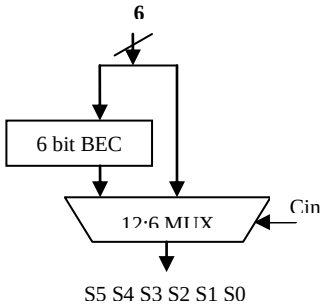


Fig 7: 6-bit BEC

In 32bit SQRT CSLA with BEC, we use 3bit BEC, 4bit BEC, 5bit BEC, 7bit BEC, 8-bit BEC & 9bit BEC so we first study how 6bit BEC works. We need 6bit BEC where we want to replaced a 5bit RCA with Cin=1. In each group the addition is perform for Cin=0 this addition is perform using RCA .Cin=1 this addition is perform using 6bit BEC logic (i.e replacing the RCA for Cin=1).From both the generated result only one o/p is selected based on carry in signal (actual Cin) from the previous group.

Addition performs by 5 bit RCA with Cin=0:
0= Cin (assumed.)
10101 + 01010 = 011111
Addition performs by 5 bit RCA with Cin=1:
1= Cin (assumed.)
10101 + 01010 = 100000
So our 6bit BEC should give the o/p 100000 when its fed by appropriate input .so from where should we fed this 6 bit input to BEC? We use the output of upper 5 bit RCA with Cin=0 as an input to 6bit BEC.

4.2 Delay and area evaluation of SQRT CSLA with BEC

As shown in fig. 8, group2 consists of a 2 bit RCA with C_{in}=0, a 3-bit BEC and a 6:3 Mux. The 2-bit RCA with C_{in}=0 consists of a FA and HA. Based on the area count of table-1, the total number of gates present in group2 is:

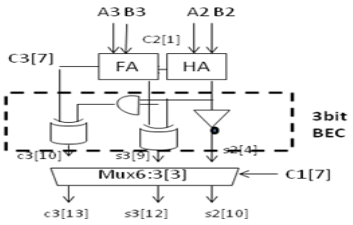


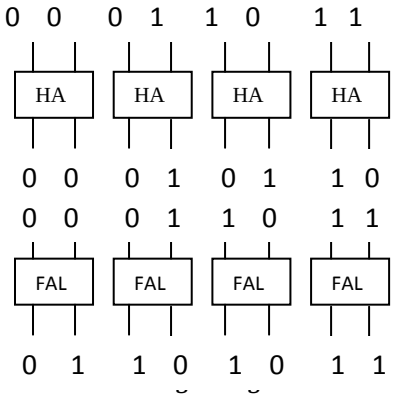
Fig 8: Group 2

Gate count of group2 = FA+HA+MUX+BEC
= 1FA+1HA+3MUX+(1NOT+2XOR+1AND)
= (1*13) + (1*6)+ (3*4)+ (2*5)+1+1=13+6+12+10+1+1 = 43
The total number of gates present in the remaining groups can be calculated in the same way as for group 2. The maximum delay can be calculated by adding the delays of gates in the longest path in the architecture that contributes the maximum delay. But delay is increased in this architecture. To reduce this delay, a new architecture of SQRT CSLA using first addition logic circuit is proposed.

5. VARIABLE SIZED 32-BIT SQRT CSLA WITH FIRST ADDITION LOGIC

5.1 Structure and working of 32 bit SQRT CSLA with FAL

The structure of the proposed 32-bit SQRT CSLA using FAL needed that the addition operation over 32 bit operand is carried out by decomposing the design into seven groups. For the sake of convenience we named them as group1 to 7.Group1 is simple 2-bit RCA .The carry output of group 1 is fed to the multiplexer of the next higher group. In our circuit, the LSB position i.e. the output of the first addition is always apposite of output sum bit of HA in group2. In the HA any out of the 4 input combination can appears and accordingly it generate sum and carry bit.



Now if the Cout from previous lower FA2 is 0 then this generated bits becomes final sum bit. If the cout from previous lower FA2 is 1 then 1 should be added to the sum bit generated (as the case in our normal addition). As we are not using FA at this LSB position to add this 3rd bit (i.e. cout =1 of prev. lower FA).So we need to take sum bit and NOT of sum bit .and these both are feed as i/p of 2:1 mux whose select line (control signal) is given by cout of prev. lower FA2.This is as shown in figure 10.

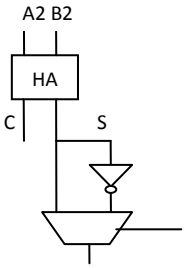


Fig 10: Selection of LSB

Our proposed design as shown in figure 11 mitigates the resource overhead of CSL by replacing lower copy of RCA by FAL. In the logic, we need to find out the zero at the LSB position. To accomplish this And gate is implemented. The output of the AND gate is used as select input to the mux which is used to select between normal sum and its complement. In other words, the carry out signal for the FAL is one if and if only if all the sum outputs from the n bit block are one. As all sum equal one, the circuit generates one at the final node. For all the other cases, it generates zero carry out.

6. SIMULATION RESULTS

The adders design proposed in this paper has been developed using VHDL and all the simulations are carried out using ISim simulator. Successful implementation of SQRT CSLA with FAL is carried out using AOI logic gates. The designs are synthesized in Leonardo spectrum to get the area (number of gates) and delay (ns).the area and delay of proposed design are calculated and compare with normal and BEC equivalent corresponding modules.

```
--------------------------------------------------------------------------------
Delay:               12.855ns (Levels of Logic = 19)
  Source:            a<3> (PAD)
  Destination:       cout (PAD)
  Data Path: a<3> to cout
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             6   0.694   0.630  a_3_IBUF (a_3_IBUF)
     LUT4:I0->O            7   0.086   0.459  fa3_c1 (fa3_c)
     LUT5:I3->O            3   0.086   0.444  fa5_c1 (fa5_c)
     LUT3:I1->O            6   0.086   0.455  fa6_c1 (fa6_c)
     LUT5:I3->O            5   0.086   0.251  fa8_c1 (fa8_c)
     MUXF7:S->O            4   0.281   0.447  fa11_c_f7 (fa11_c)
     LUT5:I3->O            5   0.086   0.452  fa13_c1 (fa13_c)
     LUT5:I3->O            4   0.086   0.447  fa15_c1 (fa15_c)
     LUT3:I1->O            4   0.086   0.447  fa16_c1 (fa16_c)
     LUT5:I3->O            5   0.086   0.452  fa18_c1 (fa18_c)
     LUT5:I3->O            5   0.086   0.452  fa20_c1 (fa20_c)
     LUT5:I3->O            5   0.086   0.452  fa22_c1 (fa22_c)
     LUT5:I3->O            6   0.086   0.455  fa24_c1 (fa24_c)
     LUT5:I3->O            5   0.086   0.452  fa26_c1 (fa26_c)
     LUT3:I1->O            2   0.086   0.615  Mxor_fa28_s_xo<0>11 (N11)
     LUT5:I1->O            3   0.086   0.444  fa29_c1 (fa29_c)
     LUT3:I1->O            2   0.086   0.772  fa30_c1 (fa30_c)
     LUT5:I0->O            1   0.086   0.235  mux30_c_y1 (cout_OBUF)
     OBUF:I->O                 2.144          cout_OBUF (cout)
    ----------------------------------------
    Total                     12.855ns (4.495ns logic, 8.360ns route)
                                       (35.0% logic, 65.0% route)
================================================================================
```

Figure 12.Timing report of proposed design

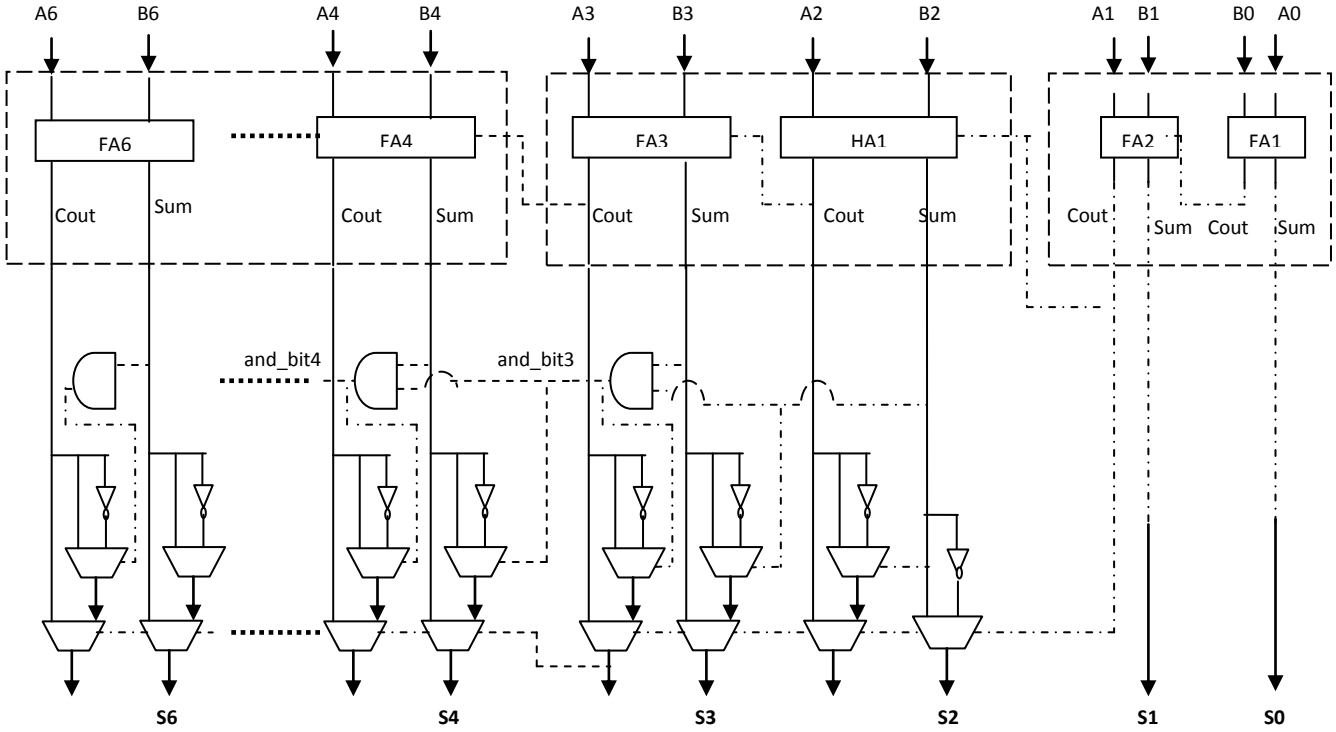


Fig 11: A group1, 2and 3 of 32-bit SQRT CSLA with proposed design.

Table 2.comparision of area and delay in between the two adders

Word size (bits)	Adder	Area(number of gates)	Delay(ns)
8	SQRT CSLA normal	318	5.892
	SQRT CSLA with BEC	280	6.268
	SQRT CSLA with first addition logic	262	5.463
32	SQRT CSLA normal	1496	9.788
	SQRT CSLA with BEC	1285	18.927
	SQRT CSLA with first addition logic	1245	12.855

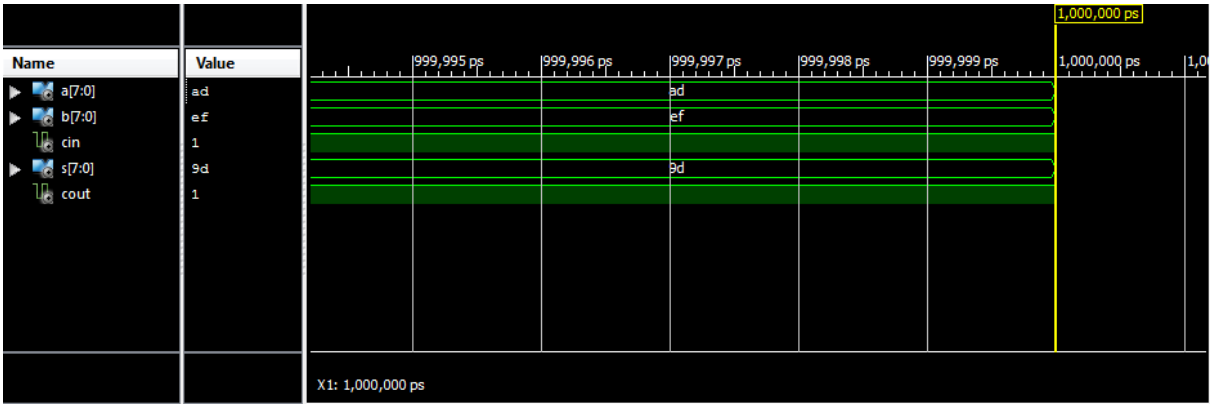


Fig 12: simulations of 8-bit proposed design.

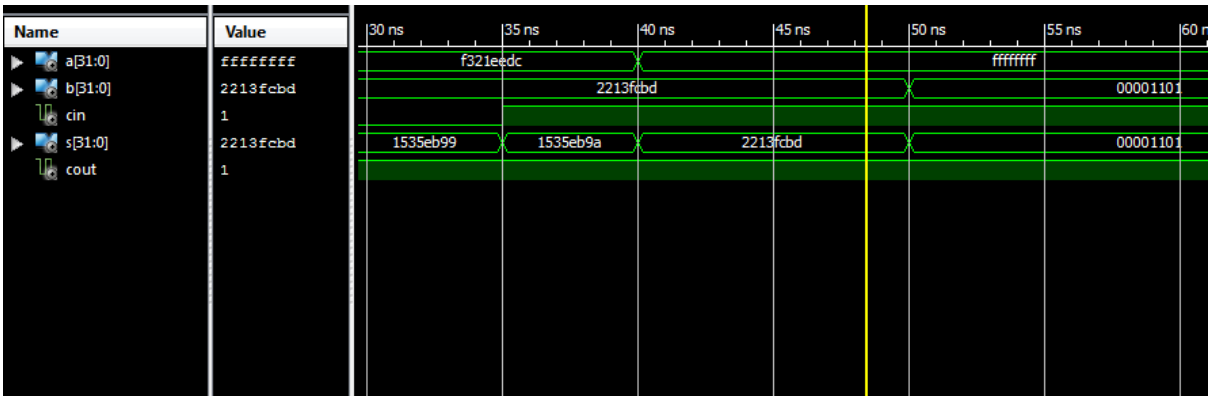


Fig 13: simulations of 32-bit proposed design.

7. CONCLUSION

A simple approach is proposed in this paper to reduce the area and delay of SQRT CSLA architecture. The reduced number of gates of this work offers the great advantage in the reduction of area and also the total delay (Table 2).The modified CSLA reduces the area and delay 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 FAL.

8. REFERENCES

[1] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput., pp.340–344, 1962.

[2] 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.

[3] 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.

[4] 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.

[5] J. M. Rabaey, Digtal 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., 2005, vol. 4, pp. 4082–4085.

[7] 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]

[8] N. Weste and K. Eshragian, Principles of CMOS VLSI Designs: A System Perspective, 2nd ed., Addison-Wesley, 1985-1993.

[9] Morinaka, H., Makino, H., Nakase, Y. et. al, "A 64 bit Carry Look-ahead CMOS adder using Modified Carry Select". Cz/stoin Integrated Circuit Conference, 1995, pages 585-588