American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-5, Issue-7, pp-146-155
www.ajer.org
Research Paper Open Access

Implementation of Cmos Adder for Area & Energy Efficient Arithmetic Applications

Prachi B. Deotale[1], Umesh W. Kaware[2], Chetan G. Thote[3]

[1]*(Dept.of Electronics & Telecommunication Engg.,Dr.Bhausaheb Nandurkar College of Engg & Tech., Yavatmal, Maharashtra, India)*

[2]*(Dept.of Electronics & Telecommunication Engg.,Jawaharlal Darda Institute of Engg & Tech.,Yavatmal, Maharashtra, India)*

[3]*(Dept.of Electronics & Telecommunication Engg.,Dr.Bhausaheb Nandurkar College of Engg & Tech., Yavatmal, Maharashtra, India)*

ABSTRACT: *The most fundamental arithmetic operation is addition which is used in a digital data path logic system. Arithmetic and logic units , Microprocessors ,etc. are some examples where we need to use arithmetic operations for processing data, for calculating addresses respectively .There are different architectures for building adder circuit .For example: 1)carry look ahead adder(CLA), 2)carry propagate adder(CPA), 3)carry save adder(CSA), & 4)carry select adder(CSLA) . Among these different architectures CSLA is a particular way of implementing adder that performs addition rapidly and are used for faster addition in many data processing processors .From observation of the carry select adder architecture we can see that there is scope for modification in order to significantly minimize the area and power consumed by the circuit. In this work we are going to propose simple and efficient modification at gate-level structure in CSLA. Based on this 16-, 32-bit square root CSLA (SQRT CSLA) have been developed & compared with regular structure. The proposed architecture design has reduced area & power consumption compared to regular structure with slight increase in delay. The evaluation of the proposed design is done based on delay, area & power performance metrics. The results show that proposed CSLA design is better than regular SQRT CSLA.*

Keywords: *Area and energy efficient, Arithmetic operations, CSLA, Data path logic systems, SQRT CSLA.*

I. INTRODUCTION

An addition is the most fundamental arithmetic operation. Adders are the electronics circuits that perform the addition of numbers. In many data path logic systems, computers and processors adders are used for ALU, address calculation, increment and decrement operators, table indices calculation and for implementation of other arithmetic operations such as subtraction, multiplication and division etc. With the increase in chip density for implementation of more and more logical functions on a single chip the problem of area and power consumption is becoming more serious & it is the most dealt one by designers. For efficient & faster operation of VLSI systems there is a lot of research on design techniques is going on [1].

In this work we have proposed gate level modification in the architecture of carry select adder (CSLA). The prime factor which hinders the faster operation of adder is the time taken in propagation of carry and this can be alleviated by using "Carry select adder". Because carry select adder generates multiple carries & then select carry to generate the SUM [8]. CSLA is faster but it is not area efficient as multiple ripple carry adder (RCA) pairs are required to generate partial SUM & CARRY by considering carry input Cin 1& Cin 0, after that multiplexer is used to select final outputs that is SUM & CARRY. Basic idea of this work is to use binary to excess-1 converter (BEC) in place of RCA with input Cin=1 in the regular CSLA structure, so that lower area and power consumption can be achieved [2]-[4]. The prime advantage of using this BEC logic at the place of RCA with Cin 1 comes from the use of lesser number of logic gates than the number of logic gates required in n-bit full adder structure .The more elaboration about BEC logic is mentioned in following sections. For the purpose of explanation we have used Square root carry select adder (SQRT CSLA) [5]-[6] in this paper.

II. METHODOLOGY FOR AREA & DELAY CALCULATION

There are three basic building blocks of CSLA

A) RCA
B) Multiplexer
C) BEC-1

1. X-or Gate:

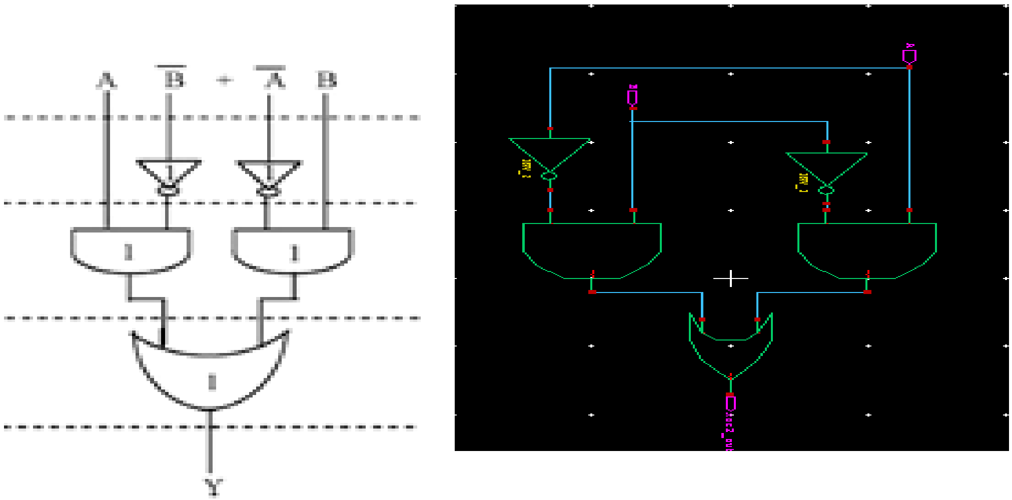


Figure 1: XOR gate implementation using AOI gates and schematic of the same is drawn using Tanner EDA Tool.

Fig 1. Shows the implementation of AND, OR, INVERTER (AOI) implementation of XOR gate. The gates shown under dotted lines perform their operation in parallel. The numeric representation of each gate indicates the delay incorporated by that logic gate.

In this method of calculation we have considered following points:
1. All gates used in implementation are AND, OR, INVERTER.
2. Each gate has gate delay equal to 1 unit.
3. We calculate delay by counting number of logic gates in the longest path of logic block & this delay contributes to maximum delay.
4. Area calculation is carried out by counting number of AOI gates required for implementation of each logic block.

Based on this methodology areas & delays of (2:1) MUX, XOR, HA & FA are calculated and results are tabulated in following table 1.

- AREA: - Total number of AOI required implementing XOR logic block are 2INV, 2AND, 1OR gates. Hence (2+2+1=5) area count is 5.
- DELAY: - Longest logic path in XOR logic block consist of 1INV, 1AND, & 1OR gate. Hence (1+1+1=3) delay is 3.

2. Binary to Excess-1 Converter:

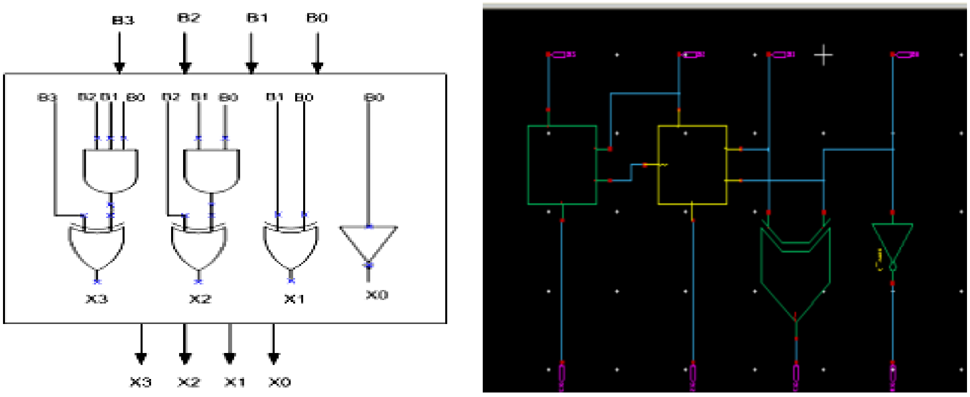


Figure 2: 4-bit BEC-1 and schematic of the same have drawn using Tanner EDA Tool.

Fig 2. Shows the implementation of BEC-1 using XOR, AND & INV gates .The operation of the 4 bi BEC-1 can be represented by using function equations (1)-(4) given as below. (LEGENDS: ~NOT, &AND, ^OR)

X0=~B0 (1)
X1=B0^B1 (2)
X2=B2^ (B0&B1) (3)
X3 =B3^ (B0&B1&B2) (4)

1. AREA :There are total 12 AOI gates are required for implementation of 4 bit BEC-1 (2AND, 3XOR (2INV, 2OR, 2AND), 1INV)(2*1+3*5+1*1=18).Hence area count is 18.

2. DELAY: There are 4 gates in the longest path of BEC-1 logic block (1XOR, 1AND) (1*3+1*1=4). Hence delay is 4.

Similarly using above method the area & delay count for different logic blocks that are used in a carry select adder structure are calculated & values are noted in Table I. below.

Table I: Delay and area of different logic blocks used in CSLA

LOGIC BLOCK	DELAY	AREA
XOR	**3**	**5**
2:1MUX	**3**	**4**
HA	**3**	**6**
FA	**6**	**13**
BEC1(4 BIT)	**4**	**12**

Based on the values derived in the previous section we can obtain the values for area count and area count of different groups of 32-bit CSLA block.

III. 32-BIT REGULAR SQRT CSLA BLOCK
IV.

1. Structure of 32 bit regular SQRT CSLA

The CSLA is used in many digital system designs to overcome the problem of carry propagation delay by independently performing addition operation by considering carry inputs (Cin) as 1 and 0.

Fig. 3 shows a 32-bit Regular 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.

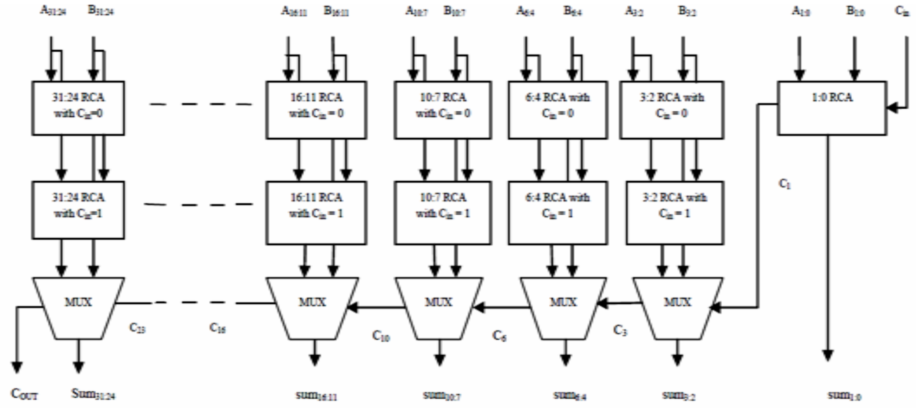

$A_{31:24}$ $B_{31:24}$
$A_{16:11}$ $B_{16:11}$
$A_{10:7}$ $B_{10:7}$
$A_{6:4}$ $B_{6:4}$
$A_{3:2}$ $B_{3:2}$
$A_{1:0}$ $B_{1:0}$ C_{in}
31:24 RCA with C_{in}=0
16:11 RCA with C_{in} = 0
10:7 RCA with C_{in} = 0
6:4 RCA with C_{in} = 0
3:2 RCA with C_{in} = 0
1:0 RCA
31:24 RCA with C_{in}=1
16:11 RCA with C_{in} = 1
10:7 RCA with C_{in} = 1
6:4 RCA with C_{in} = 1
3:2 RCA with C_{in} = 1
C_1
MUX
MUX
MUX
MUX
MUX
C_{23}
C_{16}
C_{10}
C_6
C_3
C_{OUT}
$Sum_{31:24}$
$sum_{16:11}$
$sum_{10:7}$
$sum_{6:4}$
$sum_{3:2}$
$sum_{1:0}$

Figure 3: 32 bit regular SQRT CSLA block diagram.

Based on the values of area count of different CSLA logic blocks calculated in previous section we can obtain the values of area count for each CSLA group as explained in following subsections.

2. Calculation of area count for regular 32 bit SQRT CSLA

2.1 Group 1:

First group of 32 bit regular CSLA consist of 1 2-bit RCA. Fig. 4 Shows internal Structure of 2-bit RCA. It requires 1FA & 1HA.The area count of HA is 6 and that of FA is 13.Therefore the total area count for 2-bit RCA is 19.

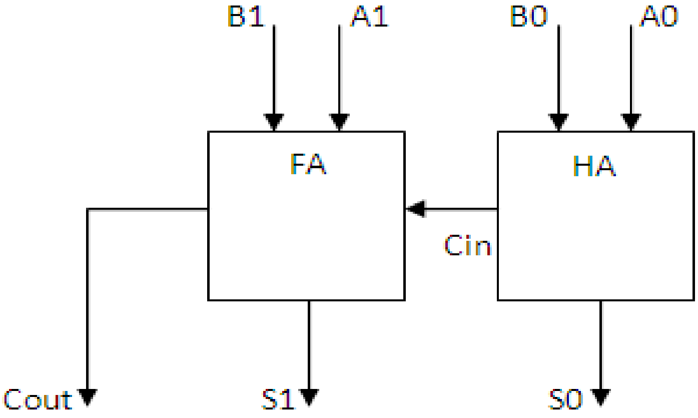


Figure 4: Group 1 of 32-bit regular SQRT CSLA (Internal structure of 2-bit RCA).

Group 1=1FA+1HA
1FA=13(1*13)
1HA=6(1*6)
AREA =19(13+6)

2.2 Group 2:

As shown in fig. 5 Second group of CSLA consists of two 2-bit RCAs (for cin=0 & cin=1) & 1(6:3) MUX. One set of RCA for cin=1 consists of 2FA and other set of RCA for cin=0 consists of 1FA and 1HA.Based on area calculation method we obtain area count for group 2 is 57.

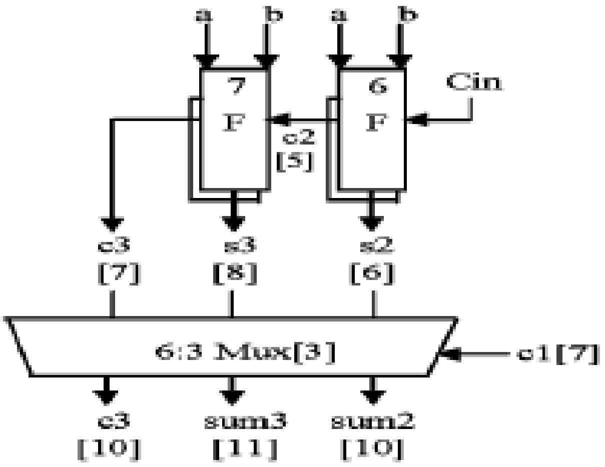


Figure 5: Group 2 of 32-bit regular SQRT CSLA.

Group 2=3FA+1HA+1(6:3) MUX
3FA=39(3*13)
1HA=6(1*6)
1(6:3) MUX=12(3*4)
AREA =57(39+6+12)

2.3 Group 3:

As shown in fig 6. The third group of CSLA consists of two 3-bit RCAs (for cin=0 & cin=1) & 1(8:4) MUX. One set of RCA for cin=0 consists of 1HA & 2FA and other RAC for cin=1 consists of 3FA.The area of Group 3 is 87.

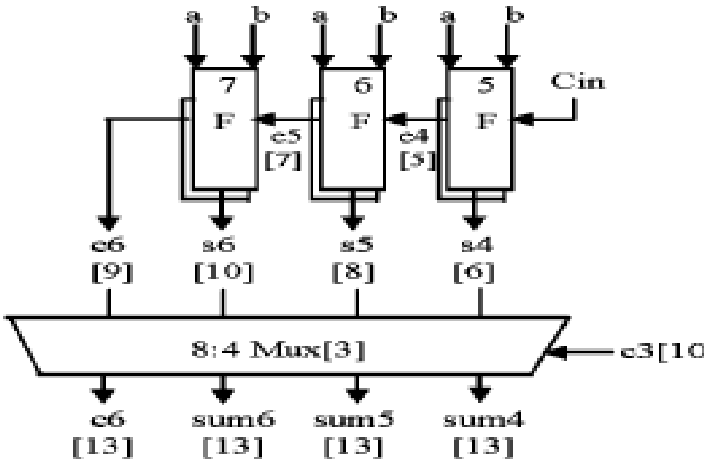


Figure 6: Group 3 of 32-bit regular SQRT CSLA.

Group 3=1HA+5FA+1(8:4) MUX
1HA=6(1*6)
5FA=65(5*13)
1(8:4) MUX=16(4*4)
AREA =87(6+65+16)

2.4 Group 4:

As shown in fig 7. The fourth group of CSLA consists of two 4-bit RCAs (for cin=0 & cin=1) & 1(10:5) MUX. One set of RCA for cin=1 consists of 4FA and other set of RCA for cin=0 consists of 1HA and 3HA.Hence area count for Group 4 is 117.

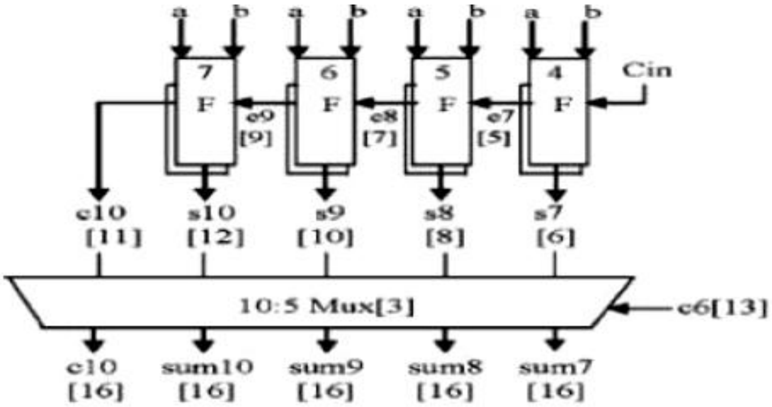


Figure 7: Group 4 of 32-bit regular SQRT CSLA.

Group 4=1HA+7FA+1(10:5) MUX
1HA=6(1*6)
7FA=91(7*13)
1(10:5) MUX=20(5*4)
AREA =117(6+91+20)

Similarly, area count for all 8 groups of 32 bit Regular CSLA is calculated and the resulting values are mentioned in the table 2. Due to the use of two RCAs there is large requirement of the area, since RCA with cin=1 requires 'n' number of FAs for n-bit addition. Because of this reason , regular CSLA is faster but it is not area efficient .This lack of area efficiency can be alleviated by using BEC-1 at the place of RCA with cin=1.

V. 32-BIT MODIFIED SQRT CSLA BLOCK

1. Structure of 32 bit modified SQRT CSLA

The fig 8.below shows the structure of 32-bit Modified SQRT CSLA .Comparing this modified structure with regular CSLA structure we can see that RCA with cin=1 is replaced by BEC-1 in the modified structure. For replacing n bit RCA we require n+1 bit BEC-1 logic block. Let us see how it affects the values of area count for the different groups of Modified 32 bit CSLA structure.

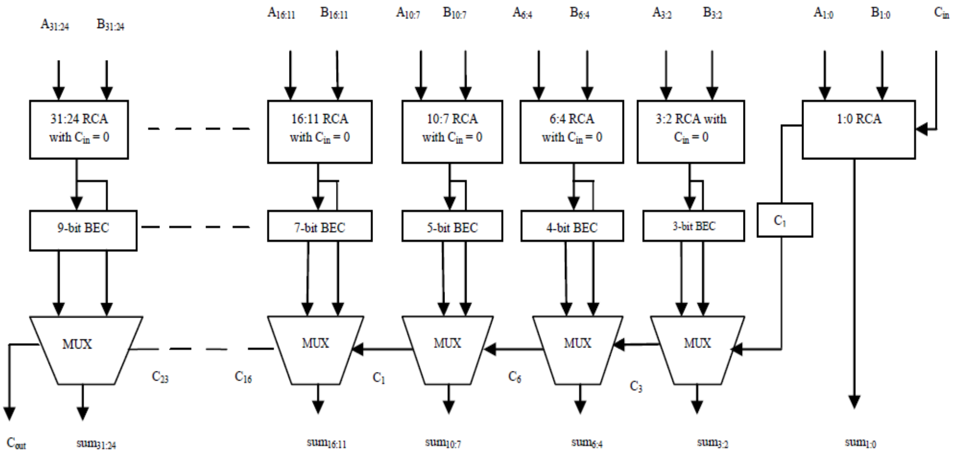


Figure 8: 32 bit modified SQRT CSLA block diagram.

2. Calculation of area count for modified 32 bit SQRT CSLA

2.1 Group 2

As shown in fig 9. Group 2 consists of 1 2-bit RCA with cin=0, 1 3-bit BEC-1, and 1 (6:3) MUX. The structure of 3-bit BEC-1 is built by using 2XOR gates, 1AND and 1NOT gate. The area count for group 2 is 43 as calculated below.

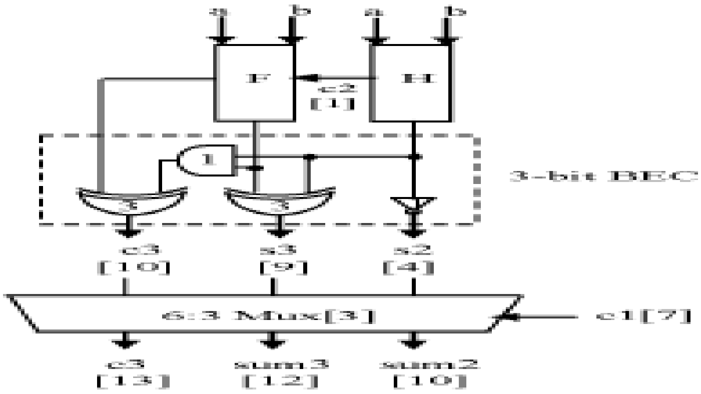


Figure 9: Group 2 of 32 bit modified SQRT CSLA.

Group 2=1FA+1HA+2XOR+1AND+1NOT+1(6:3) MUX
1FA=13(1*13)
1HA=6(1*6)
2XOR=10(2*5)
1AND=1(1*1)
1NOT=1(1*1)
1(6:3) MUX=12(3*4)
AREA =43(13+6+10+1+1+12).

2.2 Group 3:

As shown in fig 10. Group 3 consists of 1 (3-bit) RCA with cin=0, 1 (4-bit) BEC-1, 1(8:4) MUX. The structure of 4-bit BEC-1 is built by using 3XOR, 2AND & 1NOT gates. The area count for group 3 is 66 as calculated below.

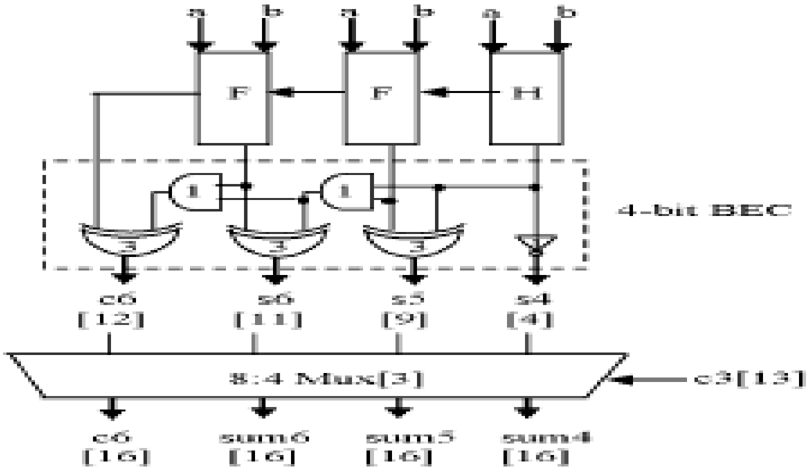


Figure 10: Group 3 of 32 bit modified SQRT CSLA.

Group3=2FA+1HA+3XOR+2AND+1NOT+1(8:4) MUX
2FA=26(2*13)
1HA=6(1*6)
3XOR=15(3*5)
2AND=2(2*1)
1NOT=1(1*1)
1(8:4) MUX=16(4*4)
AREA =66(26+6+15+2+1+16)

Similarly, area count for remaining groups of Modified 32-bit CSLA is calculated and the resulting values are noted in Table II below. As mentioned earlier we have proposed the use of BEC-1 in modified structure of 32-bit CSLA at the place of RCA with cin=1 in the regular structure.

Table II: Area count for different groups of 32-bit SQRT CSLA

GROUPS	REGULAR STRUCTURE	MODIFIED STRUCTURE
GROUP 1	19	19
GROUP 2	57	43
GROUP 3	87	66
GROUP 4	117	89
GROUP 5	147	112
GROUP 6	177	135
GROUP 7	207	158
GROUP 8	87	66
TOTAL	**898**	**688**

2.3 Calculation of reduction in area count:

898-688=**210**. Percentage of reduction in area count= 210/898*100= **23.38%.**

On comparing area count for regular and modified structure of 32-bit SQRT CSLA, it is evident that the area count of CSLA is reduced by 210 i.e. around 23.38% of area occupancy could be reduced by using modified structure for 32-bit CSLA implementation.

VI. SIMULATION RESULTS AND COMPARISON

For the simulation of 32-bit modified SQRT CSLA architecture we have used **Tanner EDA Tool (v.13)**, design is carried out using 0.25μm technology. The simulation results for area and power of regular 32-bit CSLA are taken as a reference from previous research papers [7].

Fig 11. Shows the resulting schematic of 32-bit Modified SQRT CSLA with inputs A= FFFF_FFFF & B=FFFF_FFFF.

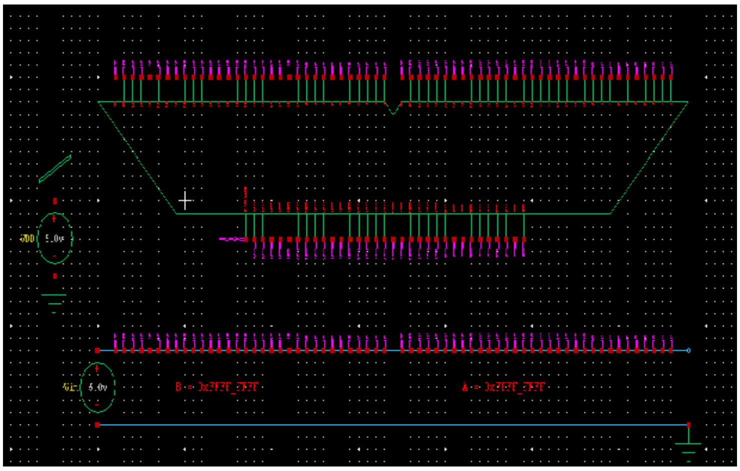


Figure 11: Schematic for 32 bit modified SQRT CSLA. (Inputs: A=FFFF_FFFF & B=FFFF_FFFF)

Fig 12. Shows schematic of the internal structure of 32-bit modified SQRT CSLA. We can see different groups of CSLA containing RCAs, BEC-1s & MUXs. Here RCA with Cin=1 is replaced by BEC-1.

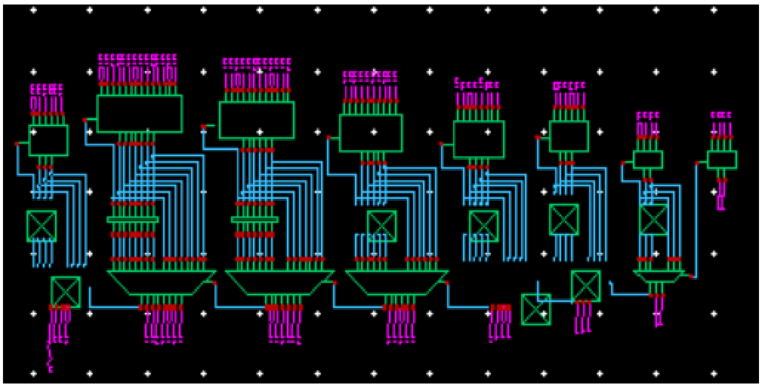


Figure 12: Schematic for Internal structure of 32-bit CSLA (RCA with cin=1 replaced by BEC-1)

Fig 13. Shows the output waveforms of 32-bit modified SQRT CSLA.

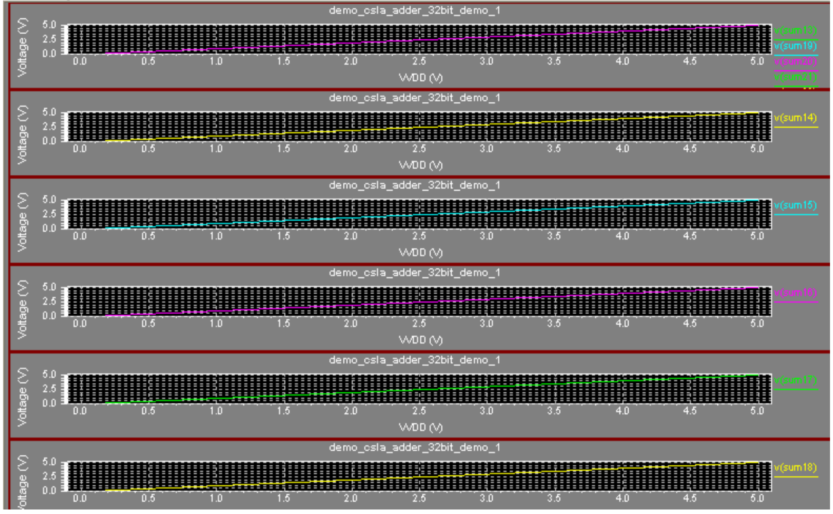
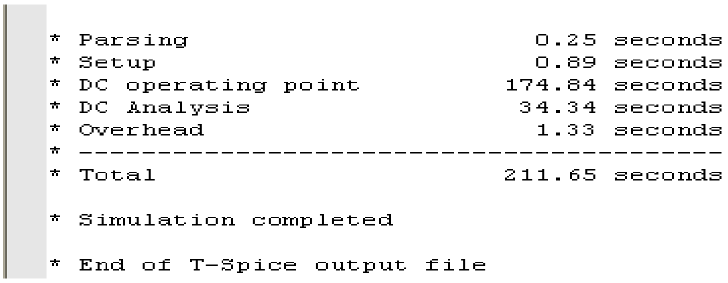


Figure 13: Output waveform of 32-bit modified SQRT CSLA (Expanded view).

Fig14. & Fig15.shows the resulting simulation status (T_spice) of 32-bit modified SQRT CSLA.The resulting parameters are tabulated in Table III & Table IV.

(Units: nA- Nano Ampere; nW-Nano Watts)

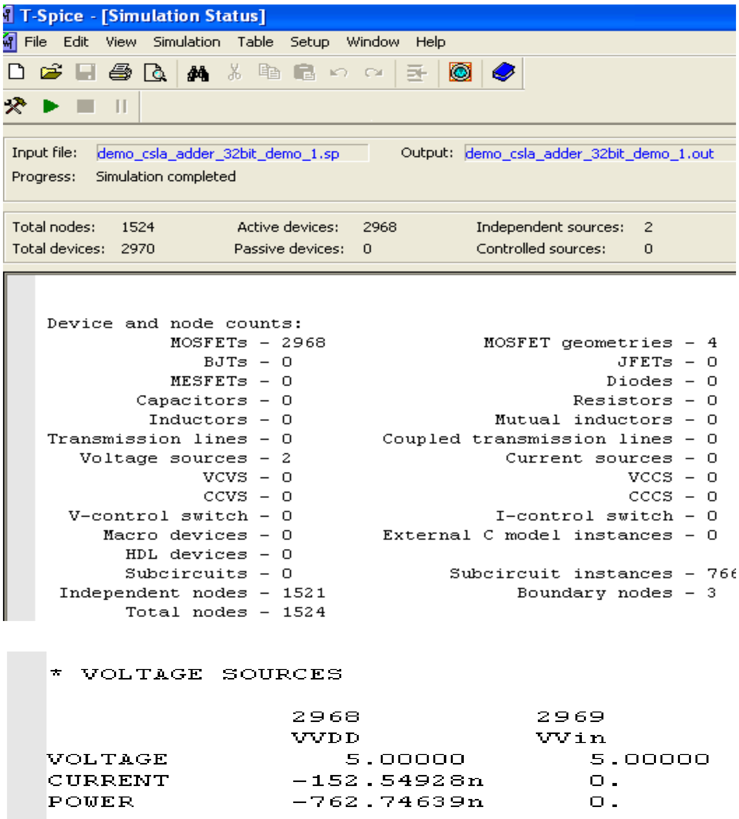


```
* Parsing                                 0.25 seconds
* Setup                                   0.89 seconds
* DC operating point                    174.84 seconds
* DC Analysis                            34.34 seconds
* Overhead                                1.33 seconds
* -----------------------------------------------------
* Total                                 211.65 seconds

* Simulation completed

* End of T-Spice output file
```

Figure 14: Simulation result (T-spice)



Input file: demo_csla_adder_32bit_demo_1.sp Output: demo_csla_adder_32bit_demo_1.out

Progress: Simulation completed

Total nodes: 1524 Active devices: 2968 Independent sources: 2

Total devices: 2970 Passive devices: 0 Controlled sources: 0

```
Device and node counts:
          MOSFETs - 2968              MOSFET geometries - 4
             BJTs - 0                             JFETs - 0
          MESFETs - 0                            Diodes - 0
       Capacitors - 0                         Resistors - 0
        Inductors - 0                  Mutual inductors - 0
Transmission lines - 0      Coupled transmission lines - 0
  Voltage sources - 2                   Current sources - 0
             VCVS - 0                              VCCS - 0
             CCVS - 0                              CCCS - 0
  V-control switch - 0                 I-control switch - 0
    Macro devices - 0        External C model instances - 0
      HDL devices - 0
      Subcircuits - 0             Subcircuit instances - 766
 Independent nodes - 1521                Boundary nodes - 3
      Total nodes - 1524
```

```
* VOLTAGE SOURCES

                    2968              2969
                    VVDD              VVin
VOLTAGE           5.00000           5.00000
CURRENT        -152.54928n          0.
POWER          -762.74639n          0.
```

Figure 15: Simulation result showing values of no. of devices used, voltage, current and power consumed.

Table III: Results in tabular form

Parameters	Resulting values
Total no. of devices	2356(MOSFETs)
Voltage	5 volts
Current	152.5492 nA
Power	762.746nW
Simulation delay	211.65secs ~3 mint.
Temp.	25^0 c.

Table IV: Device count obtained after simulation of 32-bit modified SQRT CSLA

Sr. No	Parameters	Values
1	MOSFETS	2968
2	MOSFET GEOMETRIES	4
3	TOTAL NODES	1524
4	ACTIVE DEVICES	2968
5	INDEPENDENT SOURCES	2
6	TOTAL DEVICES	2970

Table V. shows the comparison of regular & modified 32-bit CSLA based on area and power consumption.

Table V: Comparison table

Parameters	Regular 32-bit CSLA	Modified 32-bit CSLA
Area(gate count)	898	688
Power	1127.3µW	762.746nW

****Reduction in the number of transistors used:**

In the proposed work for the implementation of 32-bit modified Carry select adder for more area and energy efficient operation we require total 2968 number of transistors. There are three main constituents of 32-bit modified Carry select adder structure namely RCAs with cin=0, BEC-1s and MUXs. The distribution of total number of transistors required for implementation of these constituents is as given below:

1. Total number of transistors required to build Ripple carry adders with carry input cin=0 are 1712 transistors.
2. Total number of transistors required to build BEC-1s are 812 transistors.
3. Total number of transistors required to build MUXs are 444 transistors.
4. Total number of transistors required to build Ripple carry adders with carry input cin=1 are 1984 transistors.

When we use regular structure 32-bit Carry select adder with two Ripple carry adders (RCAs) with carry input cin=0 and cin=1, the total number of transistors required for implementation will be 4140 transistors. As shown below,

RCAs (cin=0):		1712
RCAs (cin=1):	+	1984
MUXs	:	444
Total	:	4140

For 32-bit modified structure Carry select adder we are replacing RCAs with carry input cin=1 by BEC-1s. This modification in the 32-bit CSLA structure considerably reduces the total number of transistors required for its implementation to 2968 transistors. As shown below,

RCAs (cin=0):		1712
BEC-1s	: +	812
MUXs	:	444
Total	:	2968

There is significant reduction in the number of transistors required to implement 32-bit modified SQRT CSLA that is 4140-2968= 1172 transistors are reduced.

VII. CONCLUSION

In this work we have used gate level modification i.e. BEC-1 logic is used at the place of RCA with cin=1 in regular CSLA to create modified structure .The resulting implementation shows that the area consumption has reduced by 23% and power consumption is also reduced due to use of less number of transistors (Results in Table 3,4&5). With this technique of adder implementation we can achieve more efficient structures for 64-, 128-bit adders which can give use significant reduction in area and power consumption & therefore more efficient arithmetic operation.

VIII. REFERENCES

[1]. B.Ramkumar and Harish M. Kittur"low –power and area efficient carry select adder" IEEE Trans. VLSI sytems, Vol.20, no.2, Feb 2012.

[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, 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.