Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 01, January -2018

Comparative Analysis on Energy Efficient Carry Select Adders Using BDC and BEC Converters

Damisetti Ramakrishna[1], Battula Vijay Kiran[2]

[1]*PG Scholar, V LSI Design, IIITM Gwalior*
[2]*M. Tech in Systems and Signals, JNTUK- UCEV*

Abstract —*Today there are many techniques for Carry select adders. In data-processing processors the Square Root Carry Select Adder (SQRT CSLA) is using for fastest calculations due to its speed of operations. In this paper presented a comparative analysis on SQRT CSLA. The Carry Select Adder is a fastest adder but it requires more power and area due to its complex architecture. Another SQRT CSLA with BEC is a new technique it reduces the power and area but small delay also included. The SQRT CSLA with BDC, it is also a new adder circuit. This paper explains the merits and demerits of all these adders.*

Keywords-*CSLA, Binary to Excess one Converter, Ripple Carry Adder, Low power, BDC*

I. INTRODUCTION

In VLSI system design the most important areas are area and power efficient architectures with high speed of operation. The Digital adders are most important building blocks in Digital logic circuits. The speed of operation of these adders are depends on the carry generation and its propagation. In elementary adders generally the sum for each bit position is generated sequentially only after the privies bit position has been summed and a carry propagated into the next position.

In [1] the CSLA is used in many computational systems for evaluating the carry propagation delay problem. In this technique multiple carries are generated for generating sum. But this technique is using two pair of Ripple Carry Adders (RCAs) for generating the sum with “carry = 0” and the sum with “carry = 1”. After that choosing the sum and carry values from MUXs. Here the select line for this MUX is the carry bit, which is generated from the privies bits.

In [2] this technique proposed an architecture called “Binary to Excess one Converter” for replacing the “Ripple Carry Adder with carry = 1”. This technique reduced the area and power when compared with the conventional SQRT CSLA by replacing with BEC. The main idea of work is to use BDC instead of RCA with carry = 0, in regular CSLA for reducing the area and power consumption. The main advantage of the BDC is it requires less area and power when compared to the BEC.

The SQRT CSLA with BDC [3] converter is a new technique for adder. This technique is using Binary to Decrement one Converter (BDC) instead of Ripple Carry Adder (RCA) with Carry = 1. This technique also reduced the area and power with slight increment in delay.

The details of SQRT CSLA are discussed in Section II. The details of SQRT CSLA with BEC is discussed Section III. The details of SQRT CSLA with BDC is discussed Section IV. The comparison results for all these adders are discussed in Section V. Finally the paper is concluded in Section VI.

II. CONVENTIONAL SQRT CSLA

The SQRT CLSA circuit with Binary to Excess one Converter (BEC) is shown in figure 1. In this adder the input two bits are adding by considering Carry = 0 in privies stage. The Sum output is an input for BEC circuit. The BEC will increase that output by 1. This result is for the Sum of input two numbers with Carry = 1. Now the Sum and Carry are generated for Carry = 0 and Carry = 1. The Carry bit which is generated from privies stage will be given as a selected line foe MUX. The MUX will propagate the appropriate Sum and Carry bits to the next stage. This technique reduced the area and power by replacing the second RCA with Carry = 1 by BEC.

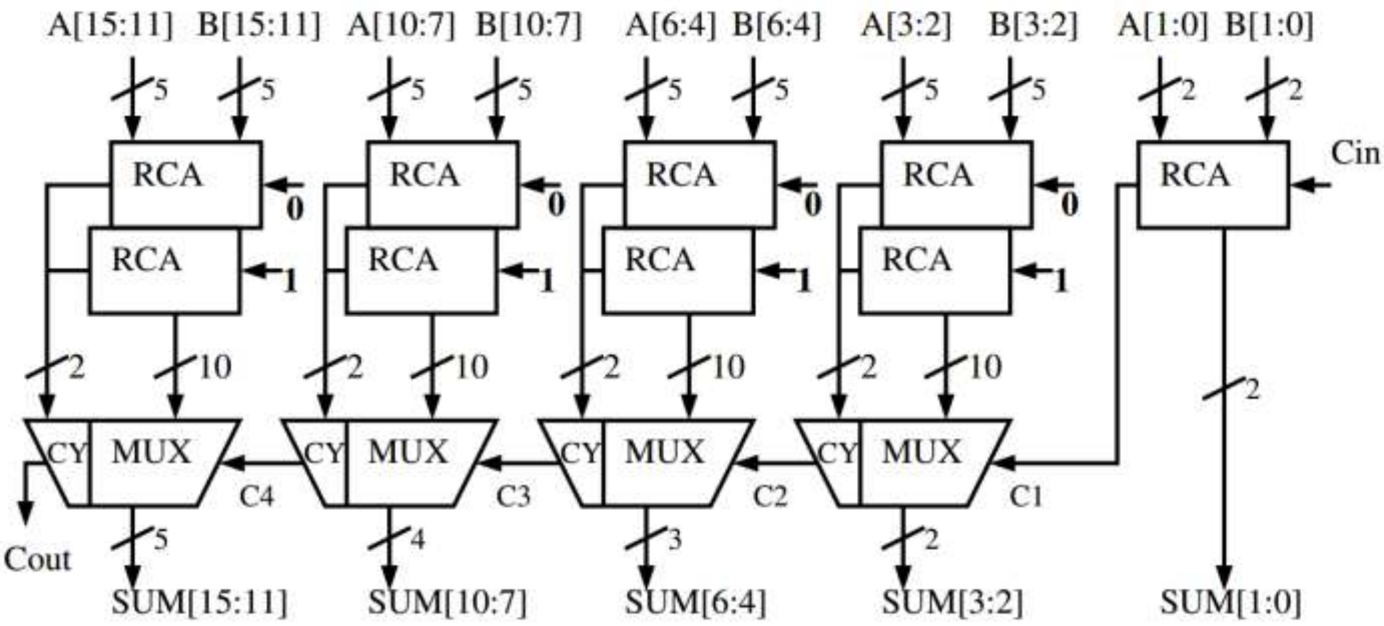

A[15:11] B[15:11] A[10:7] B[10:7] A[6:4] B[6:4] A[3:2] B[3:2] A[1:0] B[1:0]
5
2
RCA
0
1
Cin
10
CY
MUX
C4
C3
C2
C1
Cout
4
3
SUM[15:11]
SUM[10:7]
SUM[6:4]
SUM[3:2]
SUM[1:0]

Figure 1. 16-b Conventional SQRT CSLA

III. SQRT CSLA WITH BEC

The SQRT CLSA circuit with Binary to Excess one Converter (BEC) is shown in figure 2. In this adder the input two bits are adding by considering Carry = 0 in privies stage. The Sum output is an input for BEC circuit. The BEC will increase that output by 1. This result is for the Sum of input two numbers with Carry = 1. Now the Sum and Carry are generated for Carry = 0 and Carry = 1. The Carry bit which is generated from privies stage will be given as a selected line foe MUX. The MUX will propagate the appropriate Sum and Carry bits to the next stage. This technique reduced the area and power by replacing the second RCA with Carry = 1 by BEC.

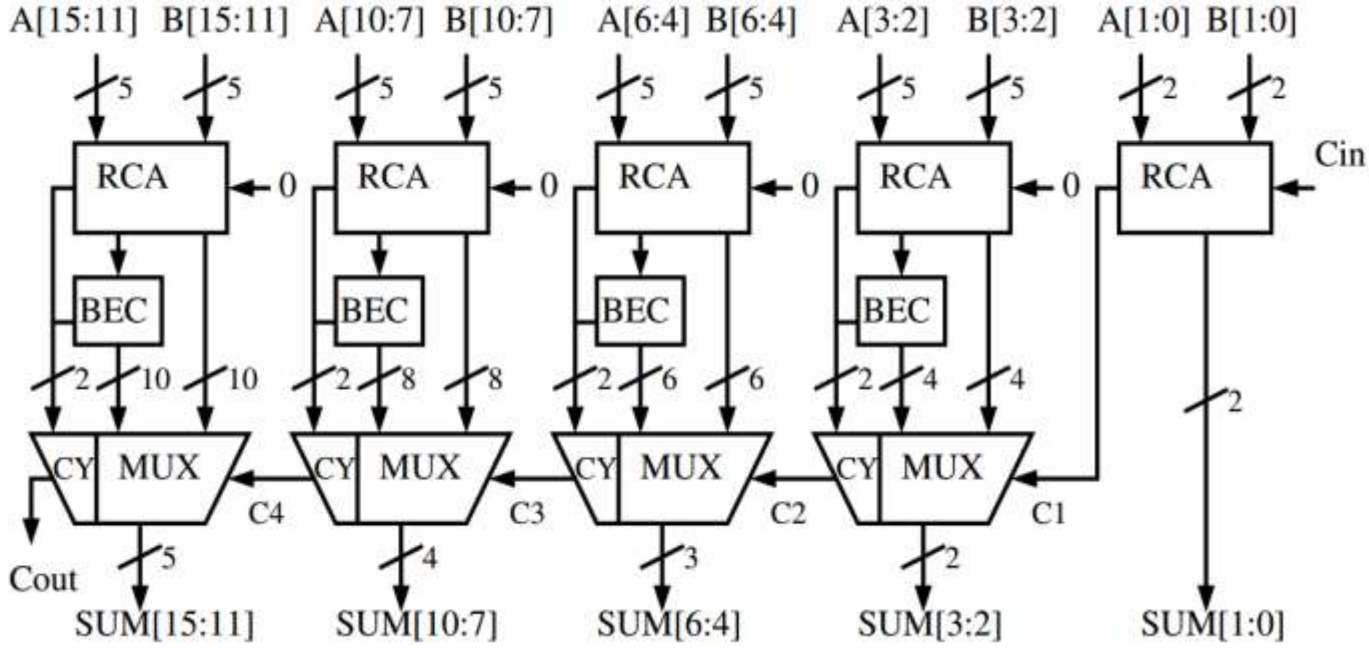


Figure 2. 16-b SQRT CSLA the parallel RCA with Carry = 1 is replaced with BEC

2.1. Binary to Excess one Converter (BEC)

It is a Digital circuit for converting the input binary number to excess one number. This figure is shown in figure 3. The 3-bit BEC requires “N – 1” number of XOR gates, “N - 2” number of AND gates and a single NOT gate. Here “N” represents the number of bits.

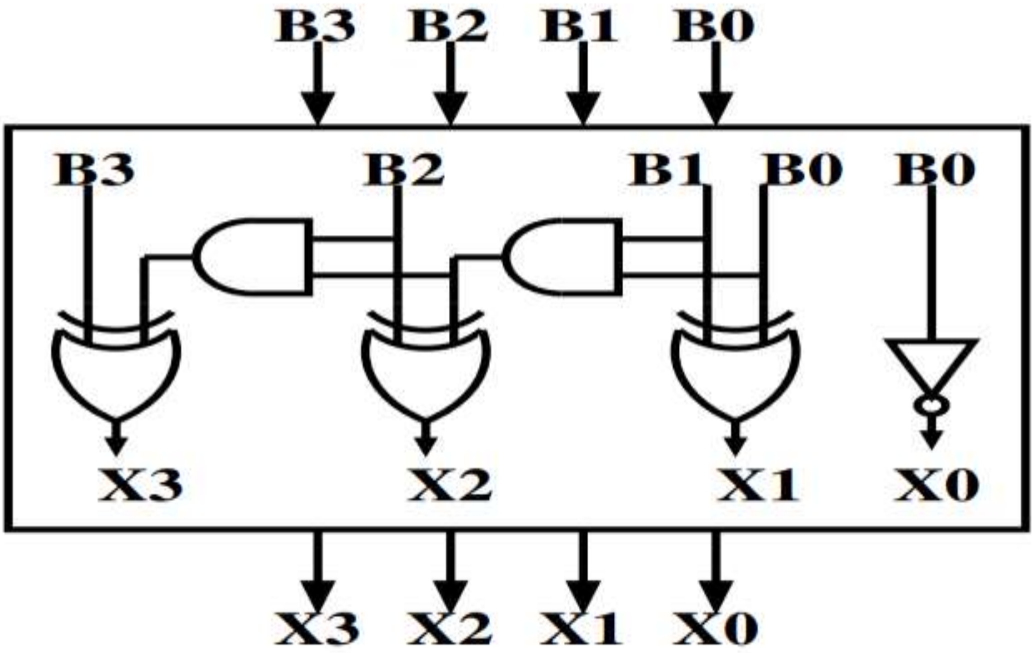


Figure 3. Binary to excess one converter

IV. SQRT CSLA WITH BDC

The SQRT CLSA circuit with Binary to Decrement one Converter (BDC) is shown in figure 4. In this adder initially the input two bits are adding by considering Carry = 1 in privies stage. The output Sum is an input for BDC circuit. The BDC will decrease that output by 1. This result is for the Sum of input two numbers with Carry = 0. Now the Sum and Carry are generated for Carry = 0 and Carry = 1. The Carry bit which is generated from privies stage will be given as a selected line foe MUX. The MUX will propagate the appropriate Sum and Carry bits to the next stage. This technique reduced the area and power by replacing the second BEC by BDC.

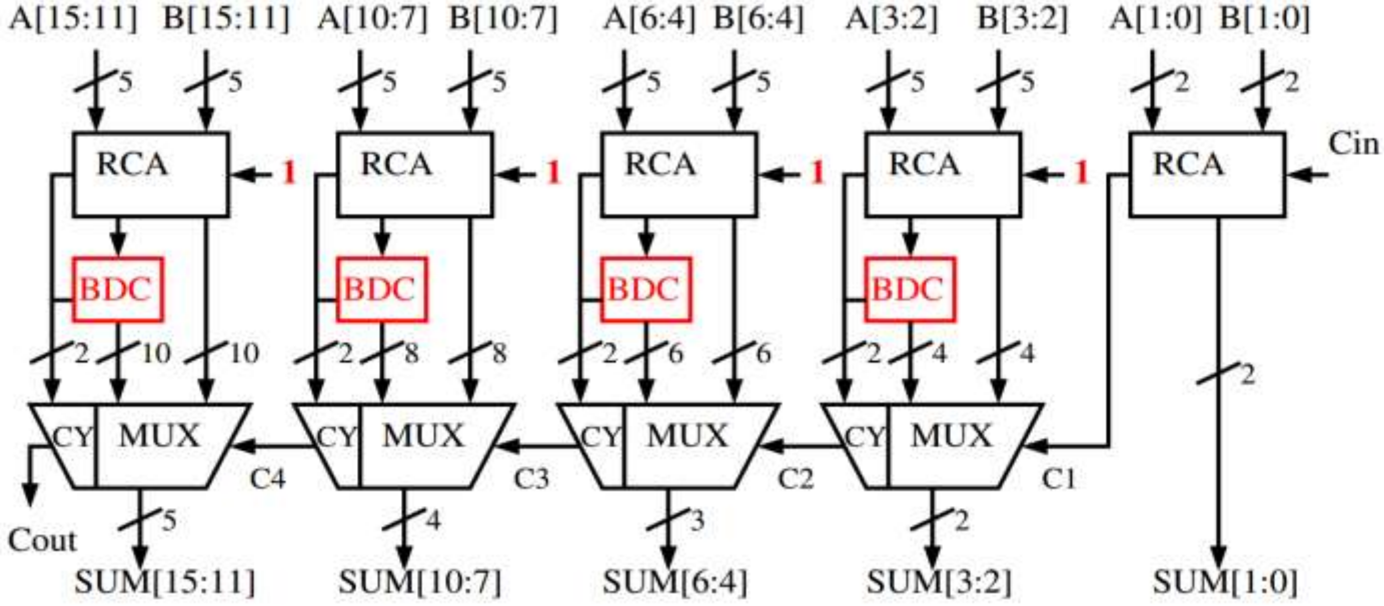


Figure 4. 16-b SQRT CSLA the parallel RCA with Carry = 0 is replaced with BDC

3.1. Binary to Decrement one Converter (BDC)

It is a Digital circuit for decreasing the input binary number by one number. This figure is shown in figure 5. The 3-bit BDC requires "N – 2" number of XOR and NOR gates, one XNOR and a single NOT gate. Here "N" represents the number of bits.

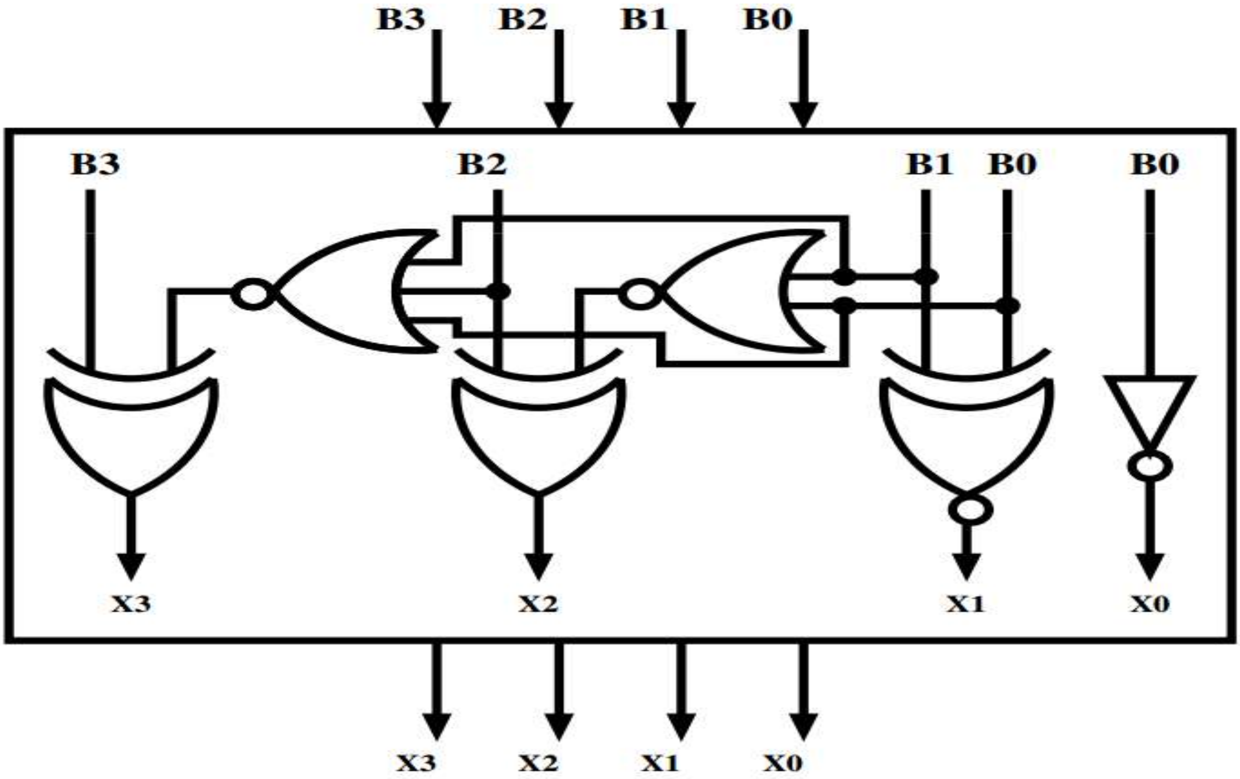


Figure 5. Binary to decrement one converter (BDC)

IV. CAMPARATIVE ANALYSIS ON SQRT CSLA ADDERS

The regular SQRT CSLA requires two ripple carry adders at each stage. One ripple carry adder requires four full adders if it is four bit RCA. One full adder requires 2 XOR gates and 2 AND gates and one OR gate. Therefore for 4-bit RCA, it requires 8 XOR gates 8 AND gates and 4 OR gates, i.e. for N-bit RCA, it requires “2N” XOR gates “2N” AND gates and “N” number of OR gates. The number of gates required for SQRT CSLA with BEC and SQRT CSLA with BDC and regular SQRT CSLA is given in Table 1. Among these three techniques regular SQRT CSLA requires more number of gates. Therefore the area and power consumption is more for regular SQRT CSLA.

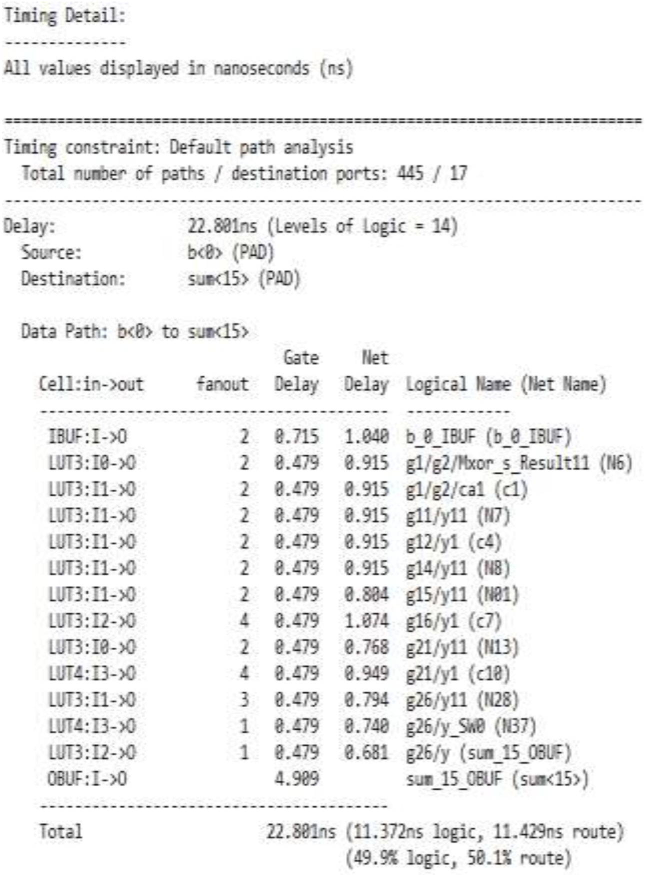
Table 1. Number of gates required for each N-bit design

Design name	XOR	XNOR	AND	OR	NOR	NOT
RCA	2N	0	2N	N	0	0
BEC	N-1	0	N-2	0	0	1
BDC	N-2	1	0	0	N-2	1

In SQRT CSLA with BEC technique it is using BEC instead of RCA with Carry = 1. Therefore the BEC need to wait until the operation completed by RCA with Carry = 0. In SQRT CSLA with BDC technique it is using BDC instead of RCA with Carry = 1. Therefore here also the BDC need to wait until the operation completed by RCA with Carry = 1. This type of waiting is not there in regular SQRT CSLA. Therefore the delay is less in regular SQRT CSLA when compared to other two techniques.

The BEC and BDC both techniques require same number of gates but the BEC circuit is using AND gates where as BDC circuit is using NOR gates. To implement the AND gate in CMOS, it requires 6 transistors but the NOR gate required only 4 transistors. Therefore the BDC circuit reduces the area and power consumption by reducing the number of transistors in the design.

The SQRT CSLA with BDC and the SQRT CSLA with BEC both techniques are implemented in Xilinx ISE Design. The figure 5 and 6 shows the timing details of SQRT CSLA with BEC and SQRT CSLA with BDC respectively. The results proved the SQRT CSLA with BDC technique requires low area and low power when compared to the SQRT CSLA with BEC.

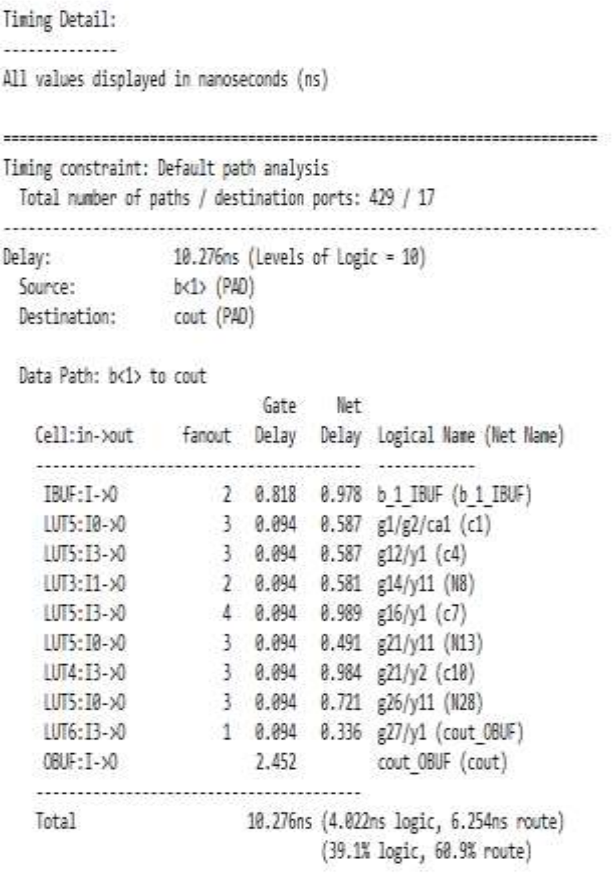


```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 445 / 17
-------------------------------------------------------------------------
Delay:               22.801ns (Levels of Logic = 14)
  Source:            b<0> (PAD)
  Destination:       sum<15> (PAD)

  Data Path: b<0> to sum<15>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   0.715   1.040  b_0_IBUF (b_0_IBUF)
     LUT3:I0->O            2   0.479   0.915  g1/g2/Mxor_s_Result11 (N6)
     LUT3:I1->O            2   0.479   0.915  g1/g2/ca1 (c1)
     LUT3:I1->O            2   0.479   0.915  g11/y11 (N7)
     LUT3:I1->O            2   0.479   0.915  g12/y1 (c4)
     LUT3:I1->O            2   0.479   0.915  g14/y11 (N8)
     LUT3:I1->O            2   0.479   0.804  g15/y11 (N01)
     LUT3:I2->O            4   0.479   1.074  g16/y1 (c7)
     LUT3:I0->O            2   0.479   0.768  g21/y11 (N13)
     LUT4:I3->O            4   0.479   0.949  g21/y1 (c10)
     LUT3:I1->O            3   0.479   0.794  g26/y11 (N28)
     LUT4:I3->O            1   0.479   0.740  g26/y_SW0 (N37)
     LUT3:I2->O            1   0.479   0.681  g26/y (sum_15_OBUF)
     OBUF:I->O                 4.909          sum_15_OBUF (sum<15>)
    ----------------------------------------
    Total                     22.801ns (11.372ns logic, 11.429ns route)
                                       (49.9% logic, 50.1% route)
```

Figure 5. Timing details of SQRT CSLA with BEC



```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 429 / 17
-------------------------------------------------------------------------
Delay:               10.276ns (Levels of Logic = 10)
  Source:            b<1> (PAD)
  Destination:       cout (PAD)

  Data Path: b<1> to cout
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   0.818   0.978  b_1_IBUF (b_1_IBUF)
     LUT5:I0->O            3   0.094   0.587  g1/g2/ca1 (c1)
     LUT5:I3->O            3   0.094   0.587  g12/y1 (c4)
     LUT3:I1->O            2   0.094   0.581  g14/y11 (N8)
     LUT5:I3->O            4   0.094   0.989  g16/y1 (c7)
     LUT5:I0->O            3   0.094   0.491  g21/y11 (N13)
     LUT4:I3->O            3   0.094   0.984  g21/y2 (c10)
     LUT5:I0->O            3   0.094   0.721  g26/y11 (N28)
     LUT6:I3->O            1   0.094   0.336  g27/y1 (cout_OBUF)
     OBUF:I->O                 2.452          cout_OBUF (cout)
    ----------------------------------------
    Total                     10.276ns (4.022ns logic, 6.254ns route)
                                       (39.1% logic, 60.9% route)
```

Figure 6. Timing details of SQRT CSLA with BDC

V. CONCLUSION

This paper compared the area, power and delay for SQRT CSLA adders. Among these three adders the regular SQRT CSLA is fastest adder. The SQRT CSLA with BEC has low area and power when compared to regular SQRT CSLA with slight increment in delay. The SQRT CSLA with BDC has low area and power when compared to both regular SQRT CSLA and SQRT CSLA with BEC. Therefore the SQRT CSLA with BDC is a energy and area efficient adder.

REFERENCES

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

[2] Ramkumar, B., Harish M. Kittur. "Low-power and area-efficient carry select adder." *IEEE transactions on very large scale integration (VLSI) systems* 20.2 (2012): 371-375.

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

[4] Y. Kim, L.-S. Kim, “64-bit carry-select adder with reduced area,”,Electron. Lett., vol. 37, no. 10, pp. 614–615, May 2001.