An Improved Architecture of 256 Bit CSLA for Reduced Area Applications

E. Lavanya[1]
[1]Assistant Professor, Sreenidhi Institute of Science & Technology, Hyderabad, Telangana, India.
Email: eadaladalavanya@gmail.com

Abstract: In the design of Integrated Circuits, area occupancy plays a vital role because of increasing the necessity of portable systems and to reduce power consumption. In designing a adder carry generation is critical way to reduce number of transistors and to reduce power consumption. Carry Select Adder (CSLA) is one of the fastest adders used and there is further scope of reducing the area in CSLA structure. The proposed design is implemented by sharing the Common Boolean Logic term (CBL) to develop an area-efficient CSLA. one XOR and one inverter in each summation operation as well as one AND gate and one inverter in each carry-out operation and Through the multiplexer, the correct output is selected according to the logic states of the carry in signal. This paper proposes efficient SQRT CSLA (CBL CSLA) architecture in terms of area. 8-bit, 16-bit, 32-bit, 64-bit, 128 bit and 256 bit architectures of CBL based CSLA are designed and compared with regular CSLA and BEC based CSLA.

Keywords: Adder, Binary to Excess-1 Converter (BEC), Carry Select Adder (CSLA), Common Boolean Logic (CBL), Exclusive or (XOR).

I. Introduction

As the scale of integration keeps growing, more and more sophisticated electronic systems are being implemented on a VLSI chip. Adders are most widely used in multipliers, DSP to execute various algorithms like FFT, FIR and IIR. As we know millions of instructions per second are performed in microprocessors. So, speed of operation is the most important constraint to be considered while designing multipliers. Due to device portability miniaturization of device should be high and power consumption should be low. Devices like Mobile, Laptops etc. require more battery backup.

So, a VLSI designer has to optimize these three parameters in a design. These constraints are very difficult to achieve so depending on demand or application some compromise between constraints has to be made. Ripple carry adders exhibits the most compact design but the slowest in speed. Whereas carry look ahead is the fastest one but consumes more area. Carry select adders act as a compromise between the two adders. In 2002, a new concept of hybrid adders is presented to speed up addition process by Wang *et al.* that gives hybrid carry look-ahead / carry select adders design. In 2008, low power multipliers based on new hybrid full adders.

Design of area- and power-efficient high-speed data path logics systems are one of the most substantial areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.

The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input ‘0’ and ‘1’, then the final sum and carry are selected by the multiplexers (MUX).

The existing modified SQRT CSLA is to use Binary to Excess-I Converter (BEC) instead of RCA with Cin=1 in the regular CSLA to achieve lower area and power consumption with slightly increase in the delay. The basic idea of the proposed architecture is that which replaces the BEC by Common Boolean Logic.

II. Basic Structure of Regular SQRT Carry Select Adder

The carry select adder comes in the category of conditional sum adder. Conditional sum adder works on some condition. Sum and carry are calculated by assuming input carry as 1 and 0 prior the input carry comes. When actual carry input arrives, the actual calculated values of sum and carry are selected using a multiplexer. The conventional carry select adder consists of K/2 bit adder for the lower half of the bits i.e. least significant bits and for the upper half i.e. most significant bits (MSB's) two K bit adders. In MSB adders one adder assumes carry input as one for performing addition and another assumes carry input as zero. The carry out calculated from the last stage i.e. least significant bit stage is used to select the actual calculated values of output carry and sum. The selection is done by using a multiplexer. This technique of dividing adder in to two stages increases the area utilization but addition operation fastens. N bit Square root CSLA with P stages, first stage adds M bits second M+1 and so on. Hence number of stages is proportional to √N and delay is proportional to √N.

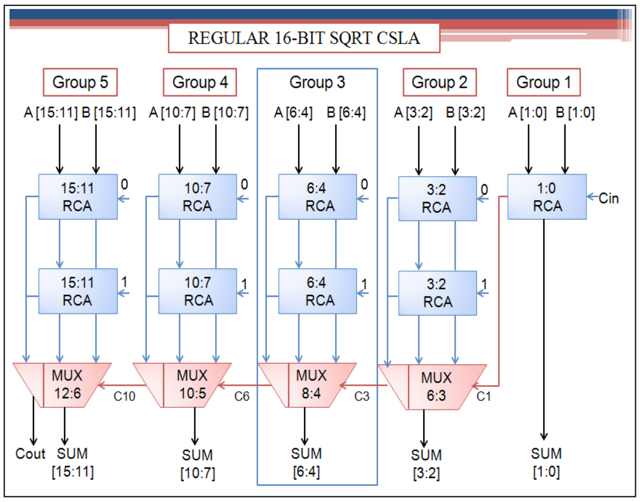


Fig. 1: Block Diagram of Regular 16-b SQRT CSLA with Groups Shown

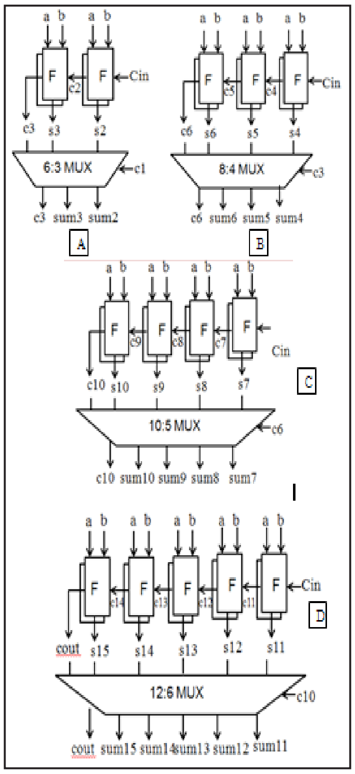


Fig. 2: Area Evaluation Methodology of 16 bit SQRT CSLA

The main disadvantage of regular CSLA is the large area due to the multiple pairs of ripple carry adder. The regular 16-bit Carry select adder is shown in Fig. 1 [7]. It is divided into five groups with different bit size RCA. From the structure of Regular CSLA, there is scope for reducing area and power consumption. The carry out calculated from the last stage i.e. least significant bit stage is used to select the actual calculated values of the output carry and sum. The selection is done by using a multiplexer.

Table I: Delay and Area Count of the Basic Blocks of CSLA

BLOCK	DELAY	AREA
XOR	3	5
2:1 MUX	3	4
HALF ADDER	3	6
FULL ADDER	6	13

Detailed internal structure of each group from group 2 to group 5 and the estimated maximum delay and area of the other groups in the regular SQRT CSLA are evaluated as shown in Fig. 2, Table I and listed in Table II and Table III.

Table II: Area Count Calculation of Individual Groups of 16 Bit SQRT CSLA

GROUP	AREA CALCULATION	AREA in GATES
Group 1	2 bit RCA = 2FA = 2*13 gates	26
Group 2	2 bit RCA(c=0) + 2 bit RCA(c=1) + 3 bit MUX = 1FA + 1HA + 2FA + 3(2:1 MUX) = 1*13 + 1*6 + 2*13 + 3*4 gates	57
Group 3	3 bit RCA(c=0) + 3 bit RCA(c=1) + 4 bit MUX = 2FA + 1HA + 3FA + 4(2:1 MUX) = 2*13 + 1*6 + 3*13 + 4*4 gates	87
Group 4	4 bit RCA(c=0) + 4 bit RCA(c=1) + 5 bit MUX = 3FA + 1HA + 4FA + 5(2:1 MUX) = 3*13 +1*6 + 4*13 + 5*4 gates	117

Table III: Area Count and Delay of 16 Bit SQRT CSLA Groups

GROUP	AREA COUNT	DELAY
Group 1	26	6
Group 2	57	11
Group 3	87	13
Group 4	117	16
Group 5	147	19

III. Structure of 16 Bit Modified SQRT CSLA using BEC

In 16 bit SQRT CSLA with BEC, we use 3 bit BEC, 4 bit BEC, 5 bit BEC, 6 bit BEC in second row replacing RCAs with Cin=1. In each group for Cin=0 addition is perform using RCA. For Cin=1 addition is perform using BEC logic. From both the generated result only one output is selected based on carry in signal (actual Cin) from the previous group. Fig. 3. shows the block diagram of modified SQRT CSLA.

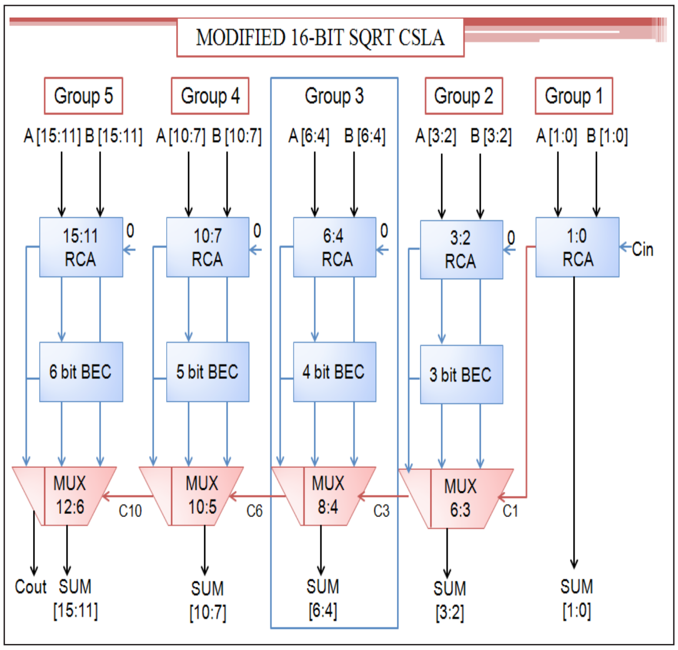


Fig. 3: Modified SQRT Carry Select Adder Parallel RCA with Cin=1 is Replaced with BEC

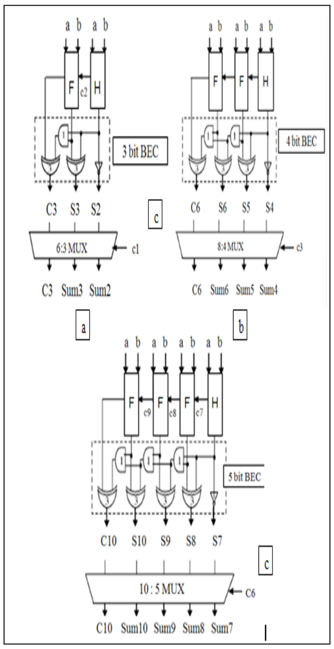


Fig. 4: Delay and Area Evaluation of Modified 16 Bit SQRT CSLA: (a) Group 2, (b) Group 3 and (c) Group 4. H is a Half Adder F is Full Adder

The delay and area estimation of each group are shown in Fig. 4 above and Fig. 5 below. The steps leading to the evaluation are given here.

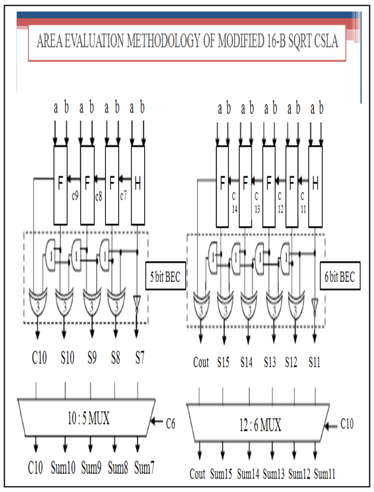


Fig. 5: Delay and Area Evaluation of Modified 16 Bit SQRT CSLA: (d) Group 5H is a Half Adder F is Full Adder

The Group 2 has one 2-b RCA which has 1 FA and 1 HA for Cin=0 Instead of another 2-b RCA with Cin=1 a 3-b BEC is used which adds one to the output from 2-b RCA. Based on the consideration of delay values of Table I, the arrival time of selection input C1 [t=7] of 6:3 MUX is earlier than the S3 [t=9] and C3 [t=10] and later than the S2 [t=4]. Thus, the Sum 3 and final C3 (output from MUX) are depending on S3 and MUX and partial C3 (input to MUX) and MUX, respectively. The Sum2 depends on C1 and MUX.

The area count of group 2 is determined as shown:

Gate count = 43 (FA + HA + MUX + BEC)

FA = 13 (1*13)

HA = 6 (1 * 6)

MUX = 12 (3*4)

BEC (3-BIT): NOT = 1, AND =1, XOR = 10 (2 * 5)

Similarly, the estimated maximum delay and area of the other groups of the modified SQRT CSLA are evaluated in Table IV and Table V.

Table IV: Area Count and Delay of Modified 16 Bit SQRT CSLA (BEC)

GROUP	AREA COUNT	DELAY
Group 1	26	11
Group 2	43	13
Group 3	66	16
Group 4	89	19
Group 5	113	22

Table V: Area Count Calculation of Individual Groups of 16 Bit Modified SQRT CSLA Using BEC

GROUP	AREA CALCULATION	AREA in GATES
Group 1	2 bit RCA = 2FA = 2*13 gates	26
Group 2	2 bit RCA(c=0) + 3 bit BEC + 3 bit MUX = 1FA + 1HA + 1 NOT + 1AND + 2XOR + 3(2:1 MUX) =1*13 + 1*6 + 1*1 + 1*1 + 2*5 + 3*4 gates	43
Group 3	3 bit RCA(c=0) + 4 bit BEC + 4 bit MUX = 2FA + 1HA + 1 NOT + 2AND + 3XOR + 4(2:1 MUX) =2*13 + 1*6 + 1*1 + 2*1 + 3*5 + 4*4 gates	66
Group 4	4 bit RCA(c=0) + 5 bit BEC + 5 bit MUX = 3FA + 1HA + 1 NOT + 3AND + 4XOR + 5(2:1 MUX) =3*13 +1*6 + 1*1 + 3*1 + 4*5 + 5*4 gates	89
Group 5	5 bit RCA(c=0) + 6 bit BEC + 6 bit MUX = 4FA + 1HA + 1 NOT + 4AND + 5XOR + 6(2:1 MUX) =4*13 + 1*6 + 1*1 + 4*1 + 5*5 + 6*4 gates	113

IV. Common Boolean Logic

To remove the duplicate adder cells in the conventional CSLA, an area efficient SQRT CSLA is proposed by sharing Common Boolean Logic (CBL) term. While analysing the truth table of single bit full adder, results show that the output of summation signal as carry-in signal is logic "0" is inverse signal of itself as carry-in signal is logic "1". It is illustrated by red circles in Fig. 6. To share the Common Boolean Logic term, we only need to implement a XOR gate and one INV gate to generate the summation pair. And to generate the carry pair, we need to implement one OR gate and one AND gate. In this way, the summation and carry circuits can be kept parallel.

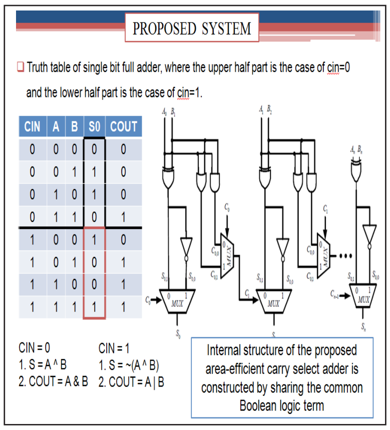


Fig. 6: Truth Table of Single Bit Full Adder

Where the upper half part is the case of Cin=0 and the lower half part is the case of Cin=1. So result of lower part is inverse of S0 result of lower part i.e~(A ^ B). Whereas Cout of upper part can be implemented using AND gate and lower part with OR gate.

V. Structure of 16 Bit Proposed CSLA Using CBL

This method replaces the Binary to Excess-1 converter add one circuit by common Boolean logic. As compared with modified SQRT CSLA, the proposed structure is little bit faster. Internal structure of proposed CBL logic is shown in Fig. 7.

In the proposed SQRT CSLA, the transistor count is trade-off with the speed in order to achieve lower power delay product. Thus the proposed SQRT CSLA using CBL is better than all the other designed adders. Fig. 8 shows the block diagram of Proposed SQRT CSLA.

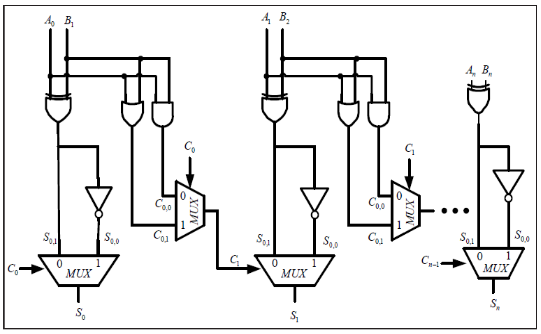


Fig. 7: Internal Structure of the Proposed Area-Efficient Carry Select Adder is Constructed by Sharing the Common Boolean Logic Term

In 16 bit SQRT CSLA with CBL, we use 2 bit BEC, 3 bit BEC, 4 bit BEC, 5 bit BEC in second row replacing RCAs with Cin=1. In each group for Cin=0 addition is perform using RCA. For Cin=1 addition is perform using CBL logic. From both the generated result only one output is selected based on carry in signal (actual Cin) from the previous group.

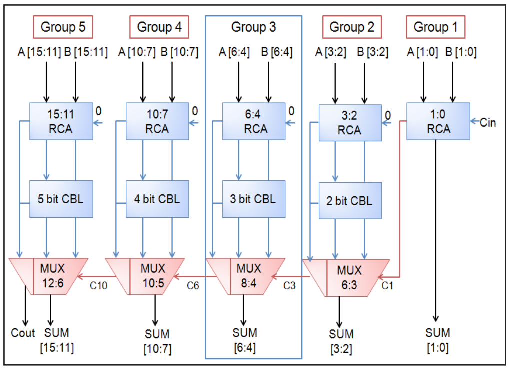


Fig. 8: 16-Bit Proposed SQRT CSLA Using Common Boolean Logic

The structure of the 16-bit Proposed SQRT CSLA is shown in Fig. 8. It has 5 groups of different size RCA and CBL. Each group contains one RCA, one CBL and MUX. In the Proposed SQRT CSLA, The group3 performed a three bit addition which are A[4] with B[4], A[5] with B[5] and A[6] with B[6]. This is done by one half adders (HA) and two full adders (FA). The CBL block has a 4:2 multiplexer to select the appropriate carryout and summation signal for carry-in signal "1". Through 2:1 multiplexer the carry signal is propagate to the next adder cell. The 6:3 MUX and 4:2 MUX is the combination of 2:1 MUX.

If the C3 = 0, The MUX select RCA output otherwise it select CBL output. The output of Group 3 are Sum [6:4] and carryout, C6. Then the area count of Group 3 is determined as shown.

Gate count = 54 (FA + HA + MUX + CBL)

FA = 26 (2*13)

HA = 6 (1 * 6)

MUX =16 (4*4)

CBL (3-BIT) = 3NOT + 3OR = 6

Similarly the estimated area of the other groups in the modified SQRT CSLA shown in Fig. 9 and Fig. 10 are evaluated and listed in Table VI.

Table VI: Area Count and Delay of Proposed 16 Bit SQRT CSLA

GROUP	AREA COUNT
Group 1	26
Group 2	35
Group 3	54
Group 4	73
Group 5	92

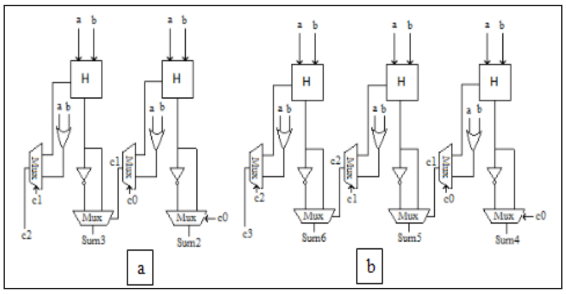


Fig. 9: Area Evaluation of Proposed SQRT CSLA Using CBL: (a) Group 2 (b) Group 3. H is a Half Adder

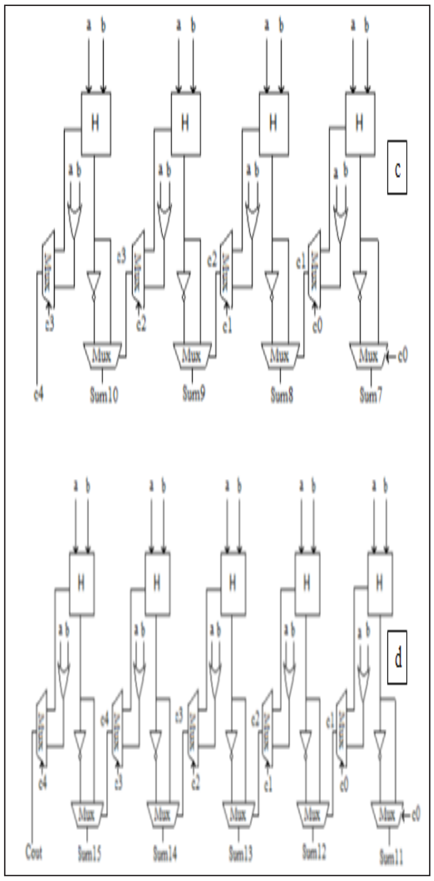


Fig. 10: Area Evaluation of Proposed SQRT CSLA Using CBL: (c) Group 4 and (d) Group 5. H is a Half Adder

VI. Results

This work has been developed using Xilinx 9.2i tool. Table VIII shows the comparison between the various adders like conventional regular SQRT CSLA, modified SQRT CSLA (BEC) and proposed SQRT CSLA (CBL) for 8-bit, 16-bit, 32-bit, 64-bit, 128-bit and 256-bit. Comparison depicts that the proposed SQRT CSLA has less number of gates and hence less area. It is clear that area, power of proposed SQRT CSLA is reduced as compared to other adders.

In following simulation results a, b are N bit data, Cin is input carry signal, Cout is carry out signal and 's' is N bit summation result of a and b.

Inputs:

a[255:0] = 256'h5f52356566489afdcd456489456,

b[255:0] = 256'haaaa58791234bc4587d986ca123564647,
cin = 0

Outputs:

s[255:0] = 256'haaaa58d8646a21abd07484975799eda9d,
cout = 0

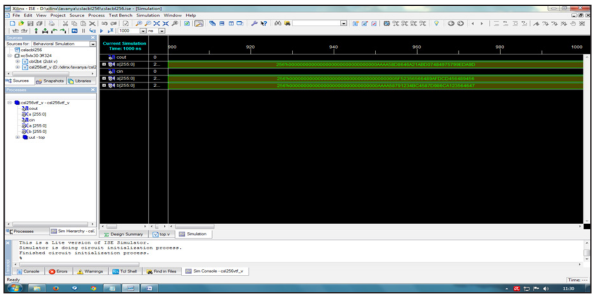


Fig. 11: Simulation Result of 256 Bit Regular, BEC, CBL SQRT CSLA

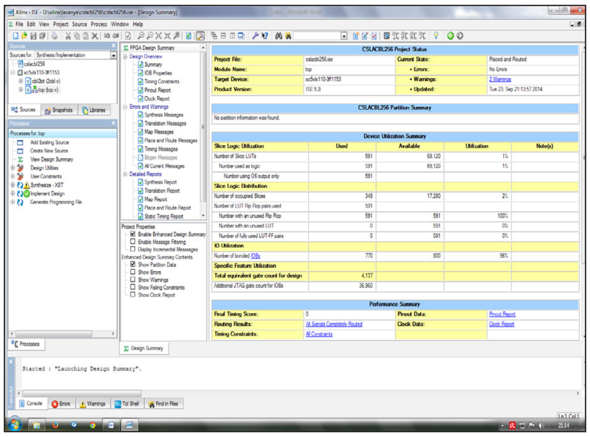
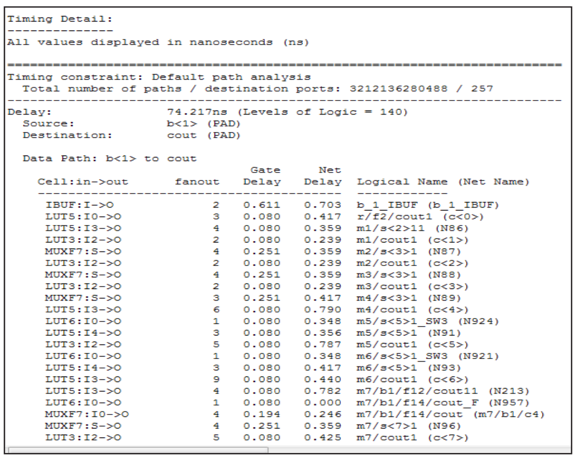


Fig. 12: Device Utilization Summary of 256 Bit CBL SQRT CSLA

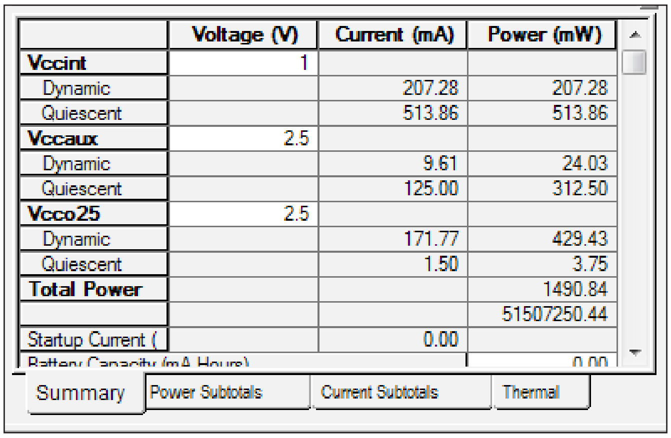


```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 3212136280488 / 257
-------------------------------------------------------------------------
Delay:               74.217ns (Levels of Logic = 140)
  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   0.611   0.703  b_1_IBUF (b_1_IBUF)
     LUT5:I0->O            3   0.080   0.417  r/f2/cout1 (c<0>)
     LUT5:I3->O            4   0.080   0.359  m1/s<2>11 (N86)
     LUT3:I2->O            2   0.080   0.239  m1/cout1 (c<1>)
     MUXF7:S->O            4   0.251   0.359  m2/s<3>1 (N87)
     LUT3:I2->O            2   0.080   0.239  m2/cout1 (c<2>)
     MUXF7:S->O            4   0.251   0.359  m3/s<3>1 (N88)
     LUT3:I2->O            2   0.080   0.239  m3/cout1 (c<3>)
     MUXF7:S->O            3   0.251   0.417  m4/s<3>1 (N89)
     LUT5:I3->O            6   0.080   0.790  m4/cout1 (c<4>)
     LUT6:I0->O            1   0.080   0.348  m5/s<5>1_SW3 (N924)
     LUT5:I4->O            3   0.080   0.356  m5/s<5>1 (N91)
     LUT3:I2->O            5   0.080   0.787  m5/cout1 (c<5>)
     LUT6:I0->O            1   0.080   0.348  m6/s<5>1_SW3 (N921)
     LUT5:I4->O            3   0.080   0.417  m6/s<5>1 (N93)
     LUT5:I3->O            9   0.080   0.440  m6/cout1 (c<6>)
     LUT5:I3->O            4   0.080   0.782  m7/b1/f12/cout11 (N213)
     LUT6:I0->O            1   0.080   0.000  m7/b1/f14/cout_F (N957)
     MUXF7:I0->O           4   0.194   0.246  m7/b1/f14/cout (m7/b1/c4)
     MUXF7:S->O            4   0.251   0.359  m7/s<7>1 (N96)
     LUT3:I2->O            5   0.080   0.425  m7/cout1 (c<7>)
```

Fig. 13: Timing Details of 256 Bit CBL SQRT CSLA



	Voltage (V)	Current (mA)	Power (mW)
Vccint	1		
Dynamic		207.28	207.28
Quiescent		513.86	513.86
Vccaux	2.5		
Dynamic		9.61	24.03
Quiescent		125.00	312.50
Vcco25	2.5		
Dynamic		171.77	429.43
Quiescent		1.50	3.75
Total Power			1490.84
			51507250.44
Startup Current (		0.00	
Battery Capacity (mA Hours)			0.00

Fig. 14: Power Analyser Report of 256 Bit CBL SQRT CSLA

VII. Comparisons

This work has been developed using Xilinx 9.2i tool. Table VII shows the comparison between the various adders like conventional regular SQRT CSLA, modified SQRT CSLA (BEC) and proposed SQRT CSLA (CBL) for 8-bit, 16-bit, 32-bit, 64-bit, 128-bit and 256-bit. Each of the CSLA implementation is done individually using higher bit BEC and CBL logic. Inputs are directly given to the adder to complete the sum and carry.

Table VIII exhibits the delay; area of regular, modified and proposed CSLA Simulation is carried out using Xilinx 9.2i simulation tool and Vertex 5 as the target device. Comparison depicts that the proposed SQRT CSLA using CBL logic has less number of gates than the regular and BEC based CSLA with slight increase in the delay and hence less area. Reduction in area is 10% to 20% for different number of bits and reduction in area gradually decreasing with higher bits. It is clear that area, power of proposed SQRT CSLA is reduced as compared to other adders. Table VIII shows comparison of Results of proposed design without cascading and results of previous design obtained by cascading Spartan 3e as target device.

Table VII: Comparison of 8, 16, 32, 64 Bits Regular, BEC CBL SQRT CSLAs Using Spartan 3e

No of bits		Total equivalent gate count		Delay (ns)	
		Previous [1]	Proposed	Previous [1]	Proposed
8	Reg	144	117	11.92	12.085
	BEC	132	96	13.69	13.125
	CBL	111	120	11.15	11.143
16	Reg	348	264	16.15	16.710
	BEC	291	261	18.77	15.361
	CBL	276	234	15.48	19.260

32	Reg	698	582	28.97	24.454
	BEC	762	609	34.44	19.969
	CBL	552	495	26.23	30.883
64	Reg	1592	1266	52.82	36.735
	BEC	1498	1275	64.61	27.410
	CBL	1104	1191	47.4	59.255

Table VIII: Area and Delay Comparison of 8 Bit,16 Bit,32bit,64 Bit,128 Bit,256 Bit Regular, BEC, CBL, SQRT CSLAs Using Vertex 5

No of bits		Total equivalent gate count	Delay (ns)
8	Reg	105	4.871
	BEC	91	5.547
	CBL	91	5.727
16	Reg	224	6.379
	BEC	231	7.709
	CBL	196	6.917
32	Reg	490	8.087
	BEC	532	10.118
	CBL	427	10.940
64	Reg	1029	10.628
	BEC	1092	11.219
	CBL	945	18.764
128	Reg	2086	14.689
	BEC	2261	14.723
	CBL	1953	34.048
256	Reg	4200	23.976
	BEC	4907	31.686
	CBL	4137	74.217

VIII. Conclusion

This paper has really given an effective description of a higher bit area efficient carry select adder. This has been achieved by altering the logic blocks of the regular module, by sharing the common Boolean logic (CBL) term, the duplicated adder cells in the conventional carry select adder is removed. A regular CSLA uses two copies of the carry evaluation blocks, one with block carry input is zero and other one with block carry input is one. The Regular SQRT CSLA has the disadvantage of more power consumptions and occupying more chip area. The reduced number of gates of this work offers the great advantage in the reduction of area. This paper proposes a scheme for 128 bit and 256 bit which reduces the area than the regular and modified SQRT CSLA 10% to 20%. Equivalent gate count of proposed SQRT CSLA is reduced having less area and low power which makes it simple and efficient for VLSI hardware implementations.

References

[1] S. Manju, V. Sornagopal, "An efficient SQRT architecture of carry select adder design by common boolean logic," *International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT),* pp. 15, 7-9 January, 2013.

[2] K. Rawat, T. Darwish, and M. Bayoumi, "A low power and reduced area carry select adder," *45th Midwest Symposium on Circuits and Systems,* vol. 1, pp. 467-470, March 2002.

[3] 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.

[4] J. M. Rabaey, "Digtal integrated circuits: A design perspective," Upper Saddle River, NJ: Prentice-Hall, 2001.

[5] S. Singh, and D. Kumar, "Design of area and power efficient modified carry select adder," *International Journal of Computer Applications (0975-8887),* vol. 33, no. 3, Nov. 2011.

[6] R. Priya, and J. S. Kumar, "Enhanced area efficient architecture for 128 bit Modified CSLA," *International Conference on Circuits, Power and Computing Technologies,* 2013.

[7] S. Parmar, and K. P. Singh, "Design of high speed hybrid carry select adder," *IEEE,* 2012.

[8] I. C. Wey, C.-C. Ho, Y.-S. Lin, and C.-C. Peng, "An area efficient carry select adder design by sharing the common boolean logic term," *Proceedings of the International MultiConference of Engineers and Computer Scientist,* IMCES HongKong, vol. 2, 14-16 March 2012.

[9] B. Ramkumar, and H. 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.

[10] S. Manjui, and V. Sornagopae, "An efficient SQRT architecture of carry select adder design by common Boolean logic," *IEEE,* 2013.

[11] www.xilinx.com