An Area and Speed Efficient Square Root Carry Select Adder Using Optimized Logic Units

Dr.P.Bhaskara Reddy, S.V.S. Prasad, K. Ananda Kumar

Abstract—**Adder is an inevitable circuit in any of the VLSI Designs. Since, the arithmetic operations such as subtraction, multiplication and division depends on the operation of addition, adder is dubbed as heart of any Digital Signal Processor (DSP), microprocessor and VLSI Architecture. In this report, the logic formulations in regular carry select adder (CSLA) and binary to excess 1 converter (BEC) based CSLA are analysed and the data dependency, redundant logic operations were scrutinized to improve the Area-Delay Product (ADP). All the redundant logic operations identified were removed and a new logic formulation has been proposed for the CSLA based on data dependency. In the proposed arrangement, the carry select (CS) operation is carried out before calculation of final-sum. Carry input for various selections (corresponding toC_{in}= 0and 1) and fixed C_{in} bits are used for logic optimization of CG and CS units. An efficient CSLA scheme is obtained using optimized logic units using BEC based CSLA Approach. The proposed CSLA design provides significant values of area and delay than the recently proposed CSLAs. Due to a small carry-output delay, the proposed CSLA design is well suited for square-root (SQRT) based variable CSLA Approach. Simulation results shows that the proposed SQRT-CSLA involves nearly 42% less area–delay–product (ADP) than the existing CSLAs. This is best among the existing CSLA designs for different bit-widths (128, 64, 32 and 16). Area in terms of logical elements and the delay in nano-seconds were obtained through Xilinx ISE 13.2 Version.**

Keywords—**ADP, BEC, CSLA, MSUX, SQRT CSLA, *etc.,***

I. INTRODUCTION

Addition is an essential operation crucial in processing the fundamental arithmetic operations [1]. It is widely used in many VLSI designs and is by far the most frequently used operation from a general purpose system to application specific processors. Since, the operations of subtraction, multiplication, division and address calculation usually rely on the operation of addition, it is an essential part of the arithmetic unit. It is dubbed as the heart of any microprocessor, Digital Signal Processing (DSP) system and VLSI Architecture.

In addition, the resulting output bits depends upon the input bits as to be of equal or having less bits. Addition is a very crucial operation because it usually involves a carry ripple step which propagates a carry signal from each bit to its higher bit position. This carry propagation produces some considerable circuit delay. Therefore, the system's overall speed is effectively determined by the adder, which lies in the critical delay path. On the other hand, to reduce the area occupancy of adder, the hardware circuitry has to be as minimum as possible by providing the same functionality [2]. Area and delay along with energy consumption of any VLSI design circuits mainly depends on adder elements and it's time to propagate the carry to the final block in the system.

A regular carry select adder (CSLA) uses pair of RCAs to produce sum words and carry output bits equal to the appropriate input carry, Cin = 0 or 1 and selects exactly one out of each pair for final sum and a final carry output [3]. A regular CSLA has less CPD compared to RCA, but it occupies more area in the design due to the usage of dual RCA [4]. Kim and Kim [5] used one RCA and one additional circuit instead of two RCAs, where the additional circuit is realized using a multiplexer (MUX). He and Chang [6] suggested a square-root (SQRT) CSLA to implement addition of large bits with less delay. Generally in a SQRT CSLA, the CSLAs are connected in a cascading structure in an increasing size of bit widths.

The main aim of this kind of design is to provide a parallel path for carry propagation that reduces the overall adder delay. B.Ramkumar and H.M.Kittur [7] recommended a binary to BEC based CSLA. The BEC based CSLA contains less logic resources than the regular CSLA, but its delay is marginally high. In papers [8] and [9] a CSLA based on common Boolean logic (CBL) is proposed. This approach based on [8] significantly uses less logic resource than the regular CSLA but its CPD is high, almost equal to that of the RCA.

To overcome this problem, a SQRT CSLA based on CBL was proposed in [9]. However, the CBL based SQRT CSLA design of uses more logic resource and delay than the BEC based SQRT CSLA approach of [6]. Here, we can observe that logic optimization largely depends on readiness of redundant operations in the formulation of logic, but adder delay largely depends on data dependence. In the current briefs, the logic is optimized without giving consideration to the data dependence [10].

In this report, analysis is made on logic operations involved in regular and BEC based CSLAs and the data dependency is studied to recognize redundant logic operations. Based on this approach, a new logic formulation for the regular CSLA has

Manuscript received August 25, 2015.

Dr.P.Bhaskara Reddy, Professor & Principal, MLRIT, India
S.V.S. Prasad, Assoc. Prof.& HOD, MLRIT, India
K. Ananda Kumar, PG Student, MLRIT , India

been proposed. The main objective of this report is to make logic formulation based on data dependence and using optimized logic units such as carry generator (CG) and carry sum (CS) blocks to achieve efficient area and speed of the design.

Based on the proposed logic formulation, an efficient logic design for SQRT CSLA has been derived. Due to the optimized logic units, the proposed SQRT CSLA involves significantly less ADP than the regular and existing adders. The SQRT-CSLA using this proposed design involves nearly 42% less ADP and consumes less energy than that of the existing SQRT CSLAs, which is advantageous in terms of Area and Delay.

The rest of this report is presented as below. Conventional CSLA with the present Logic formulation is given in Section II. The proposed CSLA using optimized logic formulation is presented in Section III and the performance analysis is presented in Section IV. Finally, in Section V conclusion is given.

II. CONVENTIONAL CSLA

A regular Carry Select Adder (CSLA) provides a substantial compromise between the Ripple Carry Adder (RCA), which occupies a small area and has a longer delay, and the Carry Look Ahead Adder (CLA), which occupies a larger area and has a shorter delay. In the CSLA, both the *n*-bit operands, Ai and Bi are divided into k blocks of possibly different sizes. An example of an 8-bit implementation of CSL comprising uniform sized blocks is shown in Fig 1(a).

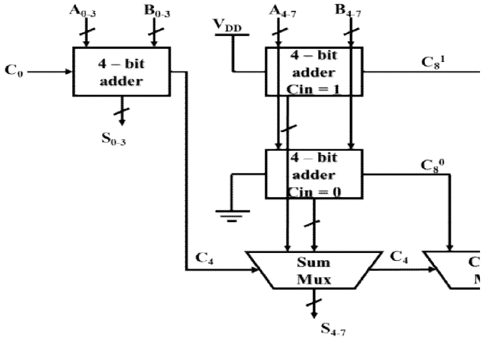


Fig. 1(a). An 8-bit conventional architecture of CSLA

The first block with $C_{in} = C_0$ is implemented by a 4-bit adder, typically the RCA, while the second block is evaluated conditionally with a pair of adders, thus permitting two additions to be executed in parallel. One addition assumes that the carry in (C_{in}) is zero while the other addition assumes that the C_{in} is one. Accordingly, two sums and two carry-outs get generated. In order to select the correct set of sum bits (S_{4-7}) and the carry in for the next block (C_8) from the respective multiplexers, a control signal, C_4, which is computed from the preceding block is required. The additional cost of the CSLA over the RCA is the duplicate carry chain and the select logic.

The Carry Select Adder (CSLA) offers a considerable negotiation between the RCA, occupying small area with longer delay, and the Carry Look Ahead (CLA) Adder, occupying larger area with shorter delay. In the CSLA, both the n-bit operands, Ai and Bi are divided into k blocks of possibly different sizes (Variable Size CSLA).

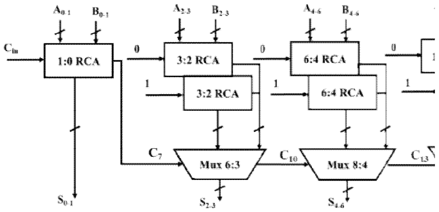


Fig. 1(b). An 16-bit conventional architecture of CSLA

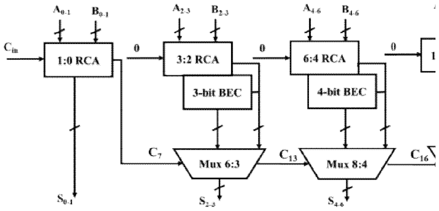
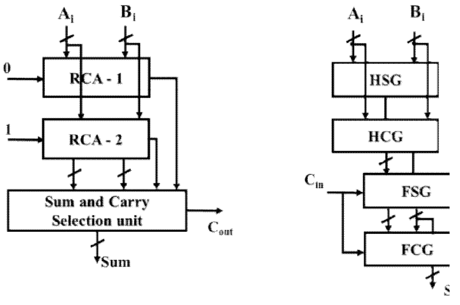


Fig. 1(c). An 16-bit BEC based architecture of CSLA

Fig 1(b) shows a 16-bit implementation of a CSLA containing variable sized blocks. The first block with $C_{in} = C_0$ is generally implemented by and 2-bit RCA adder, while the other blocks are estimated conditionally with a pair of adders, allowing additions to be performed in parallel. One addition assumes the carry in as zero and the other assumes carry in to be one. Hence, two sums and two carry outs will be produced. A control signal is used to select the correct sum and carry out values. Redundant carry chain and the select logic constitutes for additional cost in CSLA compared to RCA. A 16-bit implementation of a CSLA using Binary to Excess-1 Converter (BEC) based logic containing variable sized blocks is shown in Fig 1(c). The BEC based CSLA requires less logic resources than the regular CSLA, but the delay is marginally high.

The CSLA has mainly two units 1) A Generator unit for the *sum* and *carry* (SCG) and 2) A Selection unit for the *sum* and *carry* (SCC). The SCG unit generally consumes most of the logic resources of CSLA and considerably contributes to the critical path. Different logic designs have been proposed for effective implementation of the SCG unit. A study of the logic designs for the SCG unit of regular and BEC based CSLAs of [7] by appropriate logic expressions has been given below. The main purpose of this paper is to recognize redundant logic operations and data dependence. Consequently, by removing all redundant logic operations and sequencing logic operations based on their data dependence will optimize the design.



(a) (b)

Fig. 2(a). Simplified form of Conventional CSLA.
2(b). An 8-bit conventional architecture of CSLA

A. Logic Expressions of the SCG Unit of the Regular SQRT CSLA

As shown in Fig. 2(a), the SCG unit of the regular CSLA [3] is composed of 2 n-bit RCAs, where n is bit-width of the adder. The logic operation of the n-bit RCA is implemented in 4 stages: 1) *half-sum* generation (HSG); 2) *half-carry* generation (HCG); 3) *full-sum* generation (FSG); and 4) *full-carry* generation (FCG) as shown in Fig. 2(b). For Addition of two n-bit operands, the RCA-1 and RCA-2 generate n-bit*sum* ($s^0$and s^1) and carry output ($c^0_{out}$and c^1_{out}) equivalent to input carry (c_{in}=0 and c_{in}=1), respectively.
The Logic expressions of RCA-1 and RCA-2 of the SCG unit of the n-bit regular CSLA are given as

$$S_0^0(i) = A(i) \oplus B(i),\ C_0^0\,(i) = A(i)\ .\ B(i) \qquad (2.1)$$

$$S_1^0\,(i) = S_0^0\,(i) \oplus C_0^0\,(i\text{-}1) \qquad (2.2)$$

$$C_1^0\,(i) = C_0^0\,(i) + S_0^0\,(i)\ .\ C_0^1\,(i) = A(i)\ .\ B(i) \qquad (2.3)$$

$$S_1^1\,(i) = S_0^1\,(i) \oplus C_1^1\,(i\text{-}1) \qquad (2.4)$$

$$C_1^1\,(i) = C_0^1\,(i) + S_0^1\,(i)\ .\ C_1^1\,(i\text{-}1),\quad C_{out1}= C_1^1\,(n\text{-}1) \qquad (2.5)$$

$$\text{Where } C_1^0\,(\text{-}1) = 0,\ C_1^1\,(\text{-}1) = 1, \text{ and } 0 \le i \le n\text{-}1. \qquad (2.6)$$

As shown in (2.1)–(2.3) and (2.4)–(2.6), the logic expression of *{*S_0^0(i)*,* C_0^0(i)*}* are identical to that of *{*S_0^1(i)*,* C_0^1(i)*}*. These logic operations which are redundant can be removed to achieve an optimized design for RCA-2, where the HSG and HCG of RCA-1 is shared to construct RCA-2. Based on this approach, [5] and [6] have used an additional circuit instead of RCA-2 in the CSLA, and BEC circuit is used in [7] for the same work.

Since the BEC based CSLA offers the best area, delay and energy efficiency among the existing, here the BEC based logic expressions of the SCG unit are used for the CSLA design.

B. Logic Expression of the SCG Unit of the BEC Based SQRT CSLA

In this, the RCA calculates n-bit *sum* S_1^0 and C_{out}^0 for C_{in} = 0. Then, the BEC unit produces (n + 1) bit excess-1 code from the values of S_1^0 and C_{out}^0 from the RCA. Here, the most significant bit (MSB) of BEC represented as C_{out}^1, in which n least significant bits (LSBs) represented with S_1^1. Accordingly, the logic expressions for the RCA is same as given in (2.1)–(2.3). Then, the logic expressions for an n-bit BEC based SQRT-CSLA are given as

$$S_1^1\,(0) = {\sim}S_1^0\,(0),\quad C_1^1\,(0) = S_1^0\,(0) \qquad (2.7)$$

$$S_1^1\,(i) = S_1^0\,(i) \oplus C_1^1\,(i\text{-}1) \qquad (2.8)$$

$$C_1^1\,(i) = S_1^0\,(i).\ C_1^0\,(i\text{-}1) \qquad (2.9)$$

$$C_{out}^1 = C_1^0\,(n\text{-}1) \oplus C_1^1\,(n\text{-}1) \qquad (2.10)$$

for $1 \le i \le n - 1$.

By observing, we can find from (2.1)–(2.3) and (2.7)–(2.10) that, in the case of the BEC based CSLA, C_1^1 depends on S_1^0, but in the regular CSLA it doesn't have any dependence. Therefore, data dependence increases in the BEC based CSLA. By considering logic expressions of the regular CSLA and a further study on the data dependence will help us to find an optimized logic expression for the CSLA.

It is interesting to observe that from (2.1)–(2.3) and (2.4)–(2.6) that logic expressions of S_1^0 and S_1^1 are same except for the terms C_1^0andC_1^1since(S_0^0=S_0^1=S_0). Also, we can find thatC_1^0andC_1^1 depends on *{*S_0, C_0, C_{in}*}*, where C_0=C_0^0=C_0^1. Since C_1^0andC_1^1 have no dependency on S_1^0 and S_1^1, the logic operation of C_1^0andC_1^1 can be done before computation of S_1^0 and S_1^1, and the select unit can select any one from the set (S_1^0*,* S_1^1) for the *final-sum* of the CSLA. Here, a significant amount of logic resource is spent for calculating *{*S_1^0*,* S_1^1*}*, which is an inefficient approach to reject one sum word after the calculation. Instead, we can select the required carry word from the anticipated carry words {C^0 and C^1} to compute the *final-sum*.

The selected carry word is added with the *half-sum*(S_0) to generate the *final-sum*(S).

Using this approach, we can have three design advantages: 1) Calculation of S_1^0 can be avoided in the SCG unit 2) The n-bit select unit is required instead of the (n+1) bit; area is reduced and 3) Carry output delay is minimized; speed is achieved.

These changes will result in an area, delay and energy efficient design for the CSLA. By removing all the redundant logic operations of (2.1)–(2.3) and (2.4)–(2.6) and rearranging them we get the optimized logic expressions as below

$$S_0\,(i) = A\,(i) \oplus B(i) \text{ and } C_0\,(i) = A(i)\cdot B(i) \qquad (2.11)$$

$$C_1^0\,(i) = C_1^0\,(i{-}1)\cdot S_0(i) + C_0\,(i) \text{ for } (C_1^0(0) = 0) \qquad (2.12)$$

$$C_1^1\,(i) = C_1^1\,(i{-}1)\cdot S_0(i) + C_0\,(i) \text{ for } (C_1^1(0) = 0) \qquad (2.13)$$

$$C(i) = C_1^0\,(i) \quad \text{if } (Cin = 0) \qquad (2.14)$$

$$C(i) = C_1^1\,(i) \quad \text{if } (Cin = 1) \qquad (2.15)$$

$$C_{out}= C(n\text{-}1) \qquad (2.16)$$

$$S(0) = S_0(0) \oplus C_{in} \text{ and } S(i) = S_0\,(i) = S_0(0) \oplus C(i\text{-}1). \qquad (2.17)$$

III. PROPOSED SQRT CSLA ADDER DESIGN

The proposed SQRT CSLA is based on the logic expressions given in (2.11)–(2.17), its arrangement is as shown in Fig. 3(a). It consists of HSG unit, CG unit, CS unit and FSG unit. The CG unit consists two CGs (CG_0 and CG_1) corresponding to input-carry '0' and '1', respectively. The HSG receives two n-bit width words (A_i and B_i) and generates *half-sum* word S_0 and a *half-carry* word $C_0$of n-bit width each.

Both $CG_0$and CG_1 units receives$S_0$and$C_0$from the HSG unit and generate two n-bit full-carry words C_1^0 and C_1^1corresponding to input-carry '0' and '1'. The logic implementation of the HSG unit is as shown in Fig. 3(b). The logic circuits of CG_0 and CG_1 units were optimized to take the advantage of fixed input carry bits. The optimized designs of CG_0 and CG_1 are shown in Fig. 3(b) itself.

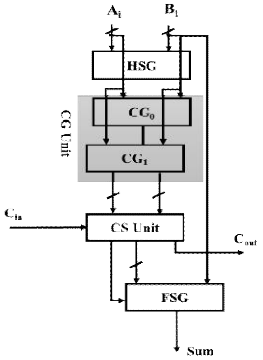


Fig. 3(a). Proposed design of CSLA

The CS unit selects one final carry word from the available carry words at its input line using the control signal C_{in}. It selects C_1^1 when C_{in}= 1; otherwise, it selects C_1^0 (C_{in}= 0). The CS unit can be implemented using an *n*-bit 2-to-l MUX. However, we find from the truth table of the CS unit that carry words C_1^0 and C_1^1follow a specific bit pattern. If $C_1^0(i) = 1$, then $C_1^1(i)$irrespective of S_0 (i) and C_0 (i) = for 0 $\leq i \leq n-1$.

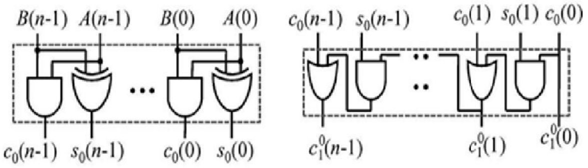


Fig. 3(b). Logic implementation of HSG Unit & CG Unit

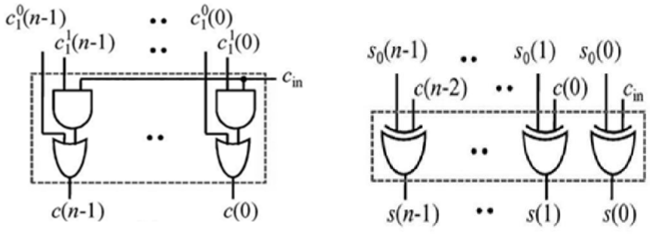


Fig. 3(c). Optimized design of Carry Selection (CS) Unit

This logic is used for optimization of the CS unit. The optimized design of the CS unit is as shown in Fig. 3(c), it consists of n number of AND–OR gates. The CS unit gives the final carry word c. The MSB of the c is sent to output as C_{out}, and the (n − 1) LSBs are XORed with (n − 1) MSBs of half-sum (S_0) in the FSG to obtain the (n − 1) MSBs of final-sum (s). Finally, the LSBs of S_0 is XORed with C_{in} to obtain the LSB of s.

Based on the above logic, to get the maximum concurrence CSLAs of increasing size are used in the carry propagation path. Using this design, adders with large bit widths can be designed with significantly less delay than the regular CSLA of same size. However, carry propagation delay between the CSLA stages of SQRT CSLA is critical and important for the overall adder delay. Due to early generation of output carry with multipath carry propagation feature, it is better to use the proposed CSLA design than the existing designs for area, delay efficient implementation of SQRT CSLA.

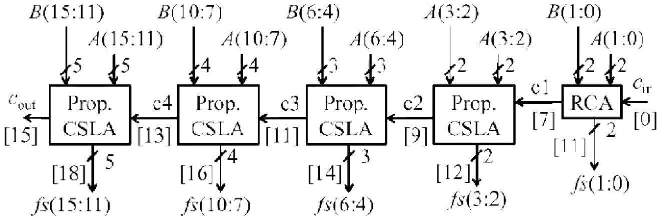


Fig. 3(d). Proposed design of SQRT-CSLA

A 16-bit SQRT CSLA design using the proposed CSLA is as shown in Fig. 3(d), it consists of 2-bit RCA followed by 2-bit, 3-bit, 4-bit and 5-bit CSLAs are used. For 32-bit we have considered the cascaded configuration of (2-bit RCA and 2, 3, 4, 6, 7, and 8 bit CSLAs).For 64-bit (2-bit RCA and 2, 3, 4, 6, 7, 8, 9, 11, and 12 bit CSLAs) and for 128-bit (2-bit RCA and 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 18 and 19 bit CSLAs) to optimize adder delay,

IV. PERFORMANCE ANALYSIS

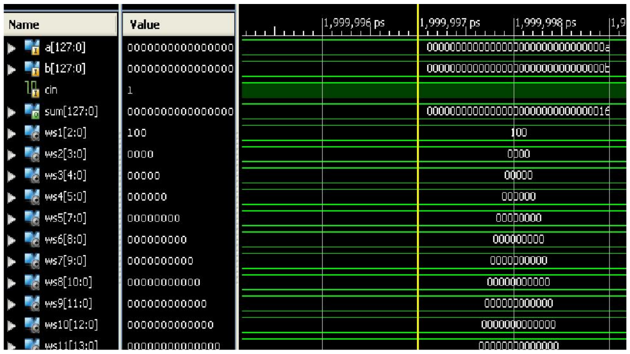
All the gates are considered to be 2-input AND, 2-input OR, and invertor (AOI). Generally, A 2-input XOR gate is composed of 2 AND gates, 1 OR gate and 2 NOT gates. The area and delay of the 128, 64, 32, 16 bit widths of proposed SQRT CSLA are given in Table 4.1.Simulation Results and design summary report of proposed 128 bit SQRT CSLA is as shown in Fig 4.1 and Fig 4.2

Respectively. Compared with the Regular CSLA designs the proposed SQRT-CSLA design involves 42% less Area-DelayProduct (ADP), on average, for different bit-widths.

Table 4.1: Area-Delay Performance of proposed CSLA

CSLA ADDER	Type	Logical Elements	Delay (ns)
16-bit	Regular	48	16.21
	Modified	43	16.12
32-bit	Regular	128	32.24
	Modified	103	23.35
64-bit	Regular	229	33.85
	Modified	208	30.68
128-bit	Regular	458	54.00
	Modified	437	43.45

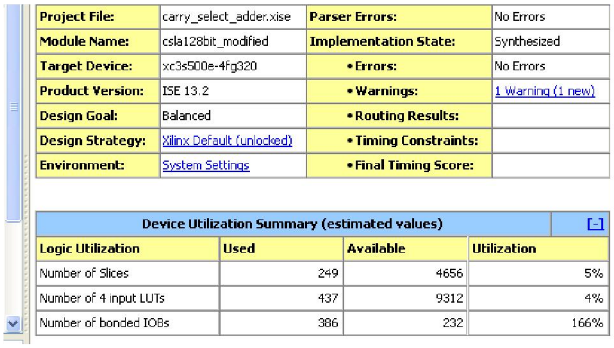
Simulation results:



Name	Value
a[127:0]	0000000000000000
b[127:0]	0000000000000000
cin	1
sum[127:0]	0000000000000000
ws1[2:0]	100
ws2[3:0]	0000
ws3[4:0]	00000
ws4[5:0]	000000
ws5[7:0]	00000000
ws6[8:0]	000000000
ws7[9:0]	0000000000
ws8[10:0]	00000000000
ws9[11:0]	000000000000
ws10[12:0]	0000000000000
ws11[13:0]	00000000000000

Fig 4.1 Simulation Results of 128-bit proposed CSLA

Design Summary Report:



Project File:	carry_select_adder.xise	Parser Errors:	No Errors
Module Name:	csla128bit_modified	Implementation State:	Synthesized
Target Device:	xc3s500e-4fg320	• Errors:	No Errors
Product Version:	ISE 13.2	• Warnings:	1 Warning (1 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values) [-]

Logic Utilization	Used	Available	Utilization
Number of Slices	249	4656	5%
Number of 4 input LUTs	437	9312	4%
Number of bonded IOBs	386	232	166%

Fig 4.2 Design summary report of 128-bit proposed CSLA

V. CONCLUSION

An efficient modified SQRT-CSLA approach is proposed in this paper to reduce the area and delay of existing CSLA architectures. Design is made for different bit widths such as 128, 64, 32, and 16 by using a single RCA and BEC Unit. Identifying Redundant Logic operations and Data Dependency made the design efficient. As a result, the proposed SQRT CSLA shows better results in terms of area power and delay. Hence, proposed SQRT CSLA can be used for area, delay and energy efficient implementations in VLSI Architectures.

REFERENCES

[1] C. Nagendra, M.J.Irwin,"Area-time-power tradeoffs in parallel adders", in IEEE Trans. Circuits and Systems-II:ADSP, Vol.43, pp.689, Oct 1996.

[2] T.Y. Chang and M.J. Hsiao, "Carry select adder using single ripple carry adder," in Electronic Lett., Vol. 34, No. 22, pp. 2101-2103, Oct. 1993.

[3] Kiat – Seng, Kaushik Roy in Low Voltage Low Power VLSI Sub Systems book.

[4] O. J. Bedrij, "Carry-select adder," in *IRE Trans. Electron. Comput.*, vol. EC-11, no. 3, pp. 340–344, Jun. 1962.

[5] Y. Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," in *Electron. Lett.*, vol. 37, no. 10, pp. 614–615, May 2001.

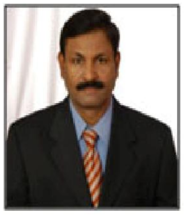
[6] Y. He, C. H. Chang, and J. Gu, "An area-efficient 64-bit square root carry-select adder for low power application," in *Proc. IEEE Int. Symp. CircuitsSyst.*, 2005, vol. 4, pp. 4082–4085.

[7] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry-select adder," in*IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 2, pp.371-375, Feb.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," in *Proc.IMECS*, 2012, pp. 1–4.

[9] S. Manju and V. Sornagopal, "An efficient SQRT architecture of carry select adder design by common Boolean logic," in *Proc. VLSI ICEVENT*, 2013, pp. 1-5.

[10] B.K.Mohanty, Sujit Kumar Patel, "Area-Delay-Power Efficient Carry-Select Adder"in IEEE Trans. on circuits and systems-II: Express Briefs, VOL.61, No.6, June 2014.



Dr. P. Bhaskara Reddy, Professor and Principal,B.E.(ECE), M.Tech (EI),Ph.D, is a young and dynamic Professor of ECE, has 26 years of Industry, Teaching, Research and Administrative experience in Reputed Engineering Colleges. In his experience he served various positions from Asst Professor to Principal. In his experience he published 9 Laboratory Manuals, 84 Research papers in National and International Level on Education, Electronics Communication, I.T., Computer Networks, etc., Guided 5 Research Scholars for their Doctorates, about 50 M.Tech,, MCA and B.Tech projects. Achieved several awards which include Rastraprathiba Award in 2004 from ICSEP, New Delhi.

Mr. S.V.S. Prasad, Associate Professor and HOD,B.Tech.(ECE), M.Tech (C&C), (Ph.D), is a dynamic and result oriented Assoc. Professor of ECE, has many years of Industry, Teaching, Research and Administrative experience in Reputed Engineering Colleges. In his experience he served various positions from Asst Professor to HOD. Published Several Research papers in National and International Level on Education, Electronics Communication, etc., His Research areas are in Image Processing and guidedmany students in theirM.Techand B.Tech projects.

Mr. K.Ananda Kumar, B.Tech.(ECE)from DMS SVH College of Engineering, Machilipatnam. Affliated to Nagarjuna University. And a PG Student in M.Tech (DSCE)from Marri Laxman Reddy Institute of Technology, Dundigal, Hyderabad.