World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:10, No:3, 2016

Performance Analysis of High Speed Adder for DSP Applications

N. Mahendran, S. Vishwaja

***Abstract*—**The Carry Select Adder (CSLA) is a fast adder which improves the speed of addition. From the structure of the CSLA, it is clear that there is opportunity for reducing the area. The logic operations involved in conventional CSLA and binary to excess-1 converter (BEC) based CSLA are analyzed to make a study on the data dependence and to identify redundant logic operations. In the existing adder design, the carry select (CS) operation is scheduled before the final-sum, which is different from the conventional CSLA design. In the presented scheme, Kogge stone parallel adder approach is used instead of existing adder design it will generate fast carry for intermediate stages and also improves the speed of addition. When compared to existing adder design the delay is less in the proposed adder design.

***Keywords*—**Binary to excess-1 converter, delay, carry select adder, Kogge stone adder, speed.

I. Introduction

REDUCED area and high speed data path logic systems are the main areas of research in VLSI system design. High-speed addition and multiplication has always been a fundamental requirement of high-performance processors and systems. 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 [1].

The CLSA is used in many computational systems to alleviate the problem of carry propagation delay (CPD) by independently generating multiple carries and then selecting 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 and then the final sum and carry are selected by the Multiplexers (mux) [2].

A conventional CSLA design is not attractive since it uses a dual RCA. Few attempts have been made to avoid dual use of RCA in CSLA design. In [3], one RCA and one add-one circuit is used instead of two RCAs, where the add-one circuit is implemented using a MUX. In [4] is proposed a square-root (SQRT)-CSLA to implement large bit-width adders with less delay. In a SQRT CSLA, CSLAs with increasing size are connected in a cascading structure. The main objective of SQRT-CSLA design is to provide a parallel path for carry propagation that helps to reduce the overall adder delay. A BEC based CSLA involves less logic resources than the conventional CSLA, but it has marginally higher delay [5]. A CSLA based on common Boolean logic (CBL) is also proposed [6].

The CBL-based CSLA [7] involves significantly less logic resource than the conventional CSLA but it has longer CPD, which is almost equal to that of the RCA. To overcome this problem, a SQRT-CSLA based on CBL was proposed [7]. However, the CBL-based SQRT-CSLA design of [7] requires more logic resource and delay than the BEC-based SQRT-CSLA [5]. In the existing adder design, logic optimization largely depends on availability of redundant operations in the formulation, whereas adder delay mainly depends on data dependence. An analysis was made on logic operations of conventional and BEC-based CSLAs to study the data dependence and to identify redundant logic operations.

In the presented adder design, SQRT CSLA architectures are designed using Kogge stone adder in order to reduce the area and delay of adder [8]-[10].

II. Existing Adder Design

A Logic Operation for Conventional CSLA

The CSLA has two units, as sum and carry generator unit (SCG), sum and carry selection unit. The SCG unit consumes most of the logic resources of CSLA and significantly contributes to the critical path. Different logic designs have been suggested for efficient implementation of the SCG unit. A study is made on the logic designs suggested for the SCG unit of conventional and BEC-based CSLAs by suitable logic expressions. The main objective of this is to identify redundant logic operations and data dependence. Accordingly, all redundant logic operations and sequence logic operations based on their data dependence is identified and removed [11]-[13].

Logic operations of RCA 1 and RCA 2 of SCG unit for n bit CSLA are given as:

$$S^0_0(i) = A(i) \oplus B(i),\ C^0_0(i) = A(i) * B(i) \quad (1)$$

$$S^0_1(i) = S^0_0(i) \oplus C^0_1(i-1) \quad (2)$$

$$C^0_1(i) = C^0_0(i) + S^0_0(i) * C^0_1(i-1),\ C^0\text{out} = C^0_1(n-1) \quad (3)$$

$$S^1_0(i) = A(i) \oplus B(i),\ C^1_0(i) = A(i) * B(i) \quad (4)$$

$$S^1_1(i) = S^1_0(i) \oplus C^1_1(i-1) \quad (5)$$

$$C^1_1(i) = C^1_0(i) + S^1_0(i) * C^1_1(i-1),\ C^1\text{out} = C^1_1(n-1) \quad (6)$$

S. Vishwaja, Student, and N. Mahendran are with the Electronics and Communication Engineering Department, M. Kumarasamy College of Engineering, Thalavapalyam, Karur (639113), India (e-mail: vishwajapugali@gmail.com, mahe.sec@gmail.com).

where C01(−1) = 0, C11(−1) = 1, and 0 ≤ i ≤ n − 1. As shown in (1)-(3) and (4)-(6), the logic operation of {S00(i), C00(i)} is identical to that of {S10(i), C10(i)}. These redundant logic operations can be removed to have an optimized design for RCA 2, in which the half sum generation (HSG) and half carry generation (HCG) of RCA-1 is shared to construct RCA 2. In HSG unit, the sum of two n bit is used as an input. HCG unit will give the carry of two n bit numbers. The full sum generation unit is based on the CS unit. In CS unit, it selects the final carry which give input to the final sum generation unit and finally final sum is found [13].

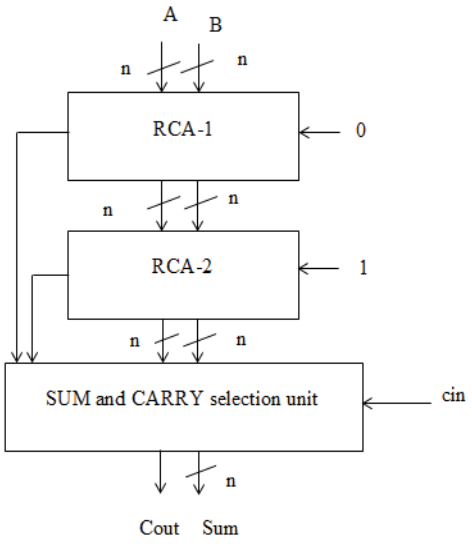

Fig. 1 Conventional CSLA

B Logic Operation for BEC Based CSLA

To reduce the area and power consumption of the regular CSLA, RCA with Cin =1 is replaced with BEC as shown in Fig. 2. An n-bit RCA can be replaced with an n+1-bit BEC. 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. The basic work is to use BEC in the regular CSLA to get lower area and improved speed of operation. This logic is replaced in RCA with C_{in}=1. The major advantage of this BEC logic comes from the fact that it uses lesser number of logic gates than the n-bit Full Adder (FA) structure [14].

The logic expressions of the BEC unit of the n bit BEC-based CSLA are given as

$$S^1{}_1(0) = S^0{}_1(0),\ C^1{}_1(0) = S^0{}_1(0) \quad (6)$$

$$S^1{}_1(i) = S^0{}_1(i) \oplus C^1{}_1(i-1) \quad (7)$$

$$C^1{}_1(i) = S^0{}_1(i) * C^1{}_1(i-1) \quad (8)$$

$$C^1\text{out} = C^0{}_1(n-1) \oplus C^1{}_1(n-1) \quad (9)$$

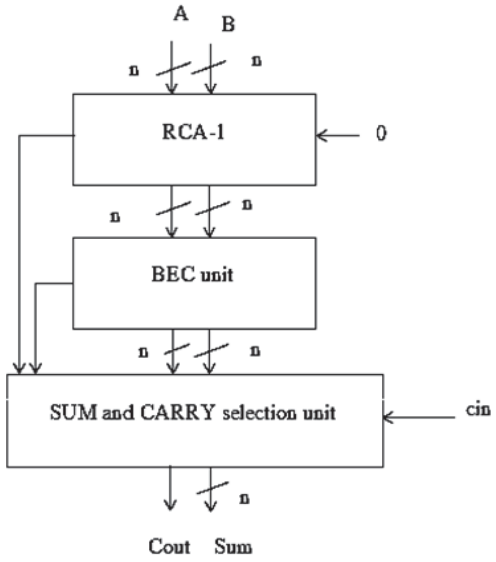

Fig. 2 BEC based SQRT CSLA.

In the case of BEC based CSLA ((1)-(3) and (4)-(6)), C11 depends on S01, which otherwise has no dependence on S01 in the case of the conventional CSLA. The BEC method therefore increases data dependence in the CSLA. It is interesting to note from (1)-(3) and (4)-(6) that logic expressions of S01 and S11 are identical except the terms C01 and C11 since (S00 = S10 = S0). In addition, C01 and C11 depend on {S0, C0, Cin}, where C0= C00 = C10. Since C01 and C11 have no dependence on S01 and S11, the logic operation of C01 and C11 can be scheduled before S01 and S11, and the select unit can select one from the set (S01, S11) for the final-sum of the CSLA [14], [18]-[20].

A significant amount of logic resource is spent for calculating {S01, S11}, and it is not an efficient approach to reject one sum-word after the calculation. Instead, one can select the required carry word from the anticipated carry words {C0 and C1} to calculate the final-sum. The selected carry word is added with the half-sum (S0) to generate the final-sum (S). Using this method, calculation of S01 is avoided in the SCG unit, the n - bit select unit is required instead of the (n + 1) bit and small output - carry delay. All these features result in an area delay and energy-efficient design for the CSLA. All the redundant logic operations from (1)-(3) and (4)-(5) are removed and rearranged based on their dependence [14].

C Logic Operation for Existing Adder Design

The CSLA structure has sum and carry selection unit. The main objective of this study is to identify redundant logic operations and data dependence [15].

The logic formulation for the CSLA is given as:

$$S_0(i) = A(i) \oplus B(i),\ C_0(i) = A(i) * B(i) \quad (10)$$

$$C^0{}_1(i) = C^0{}_1(i-1) * S_0(i) + C_0(i), \text{ for } \_\ C^0{}_1(0) = 0 \quad (11)$$

$$C^1{}_1(i) = C^1{}_1(i-1) * S_0(i) + C_0(i), \text{ for } \_\ C^1{}_1(0) = 1 \quad (12)$$

$$C(i) = C^0{}_1(i) \text{ if } (C_{in} = 0) \quad (13)$$

$$C(i) = C^1{}_1(i) \text{ if } (C_{in} = 1),\ C_{out} = C(n-1) \quad (14)$$

$$S(0) = S_0(0) \oplus C_{in},\ S(i) = S_0(i) \oplus C(i-1) \quad (15)$$

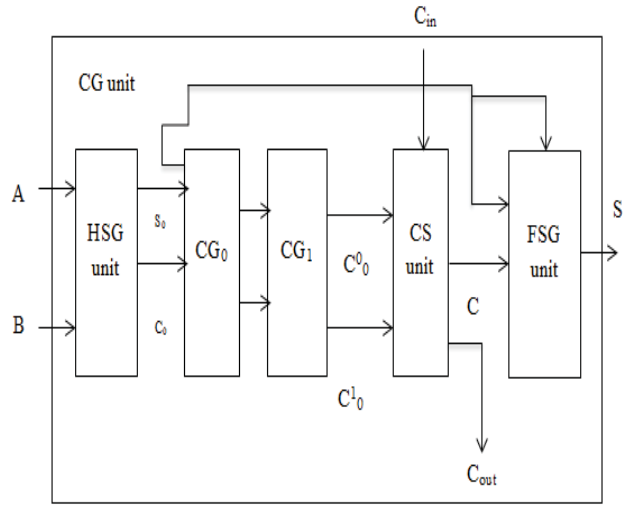

Fig. 3 Existing SQRT CSLA Adder design

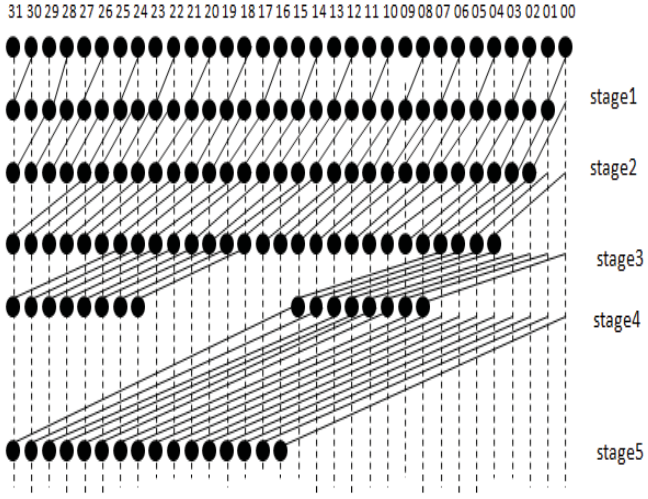

Fig. 4 32 Bit KS adder

III. Proposed SQRT CSLA Using Kogge Stone Adder

The parallel prefix adders are more flexible and are used to speed up the binary additions. Parallel prefix adders are obtained from Carry Look Ahead (CLA) structure. This uses tree structure form to increase the speed of arithmetic operation. Parallel prefix adders are fastest adders and these are used for high performance arithmetic circuits in industries. The construction of parallel prefix adder involves three stages,

- Pre-processing stage,
- Carry generation network,
- Post processing.

There are some parallel prefix adder types such as Kogge Stone adder, Brent Kung adder, Han Carlson adder, Ladner Fischer adder etc. The Kogge Stone (KS) adder is a parallel prefix which is a form of CLA adder. KS prefix adder is a fast adder design. KS adder has best performance in VLSI implementations. KS adder has large area with minimum fan-out. The KS is widely known as a parallel prefix adder that performs fast logical addition. KS adder is used for wide adders; because it shows the less delay among other architectures. Each vertical stage will produce Propagate and Generate bits. Generated bits are produced in the last stage and these bits are XORed with the initial propagate after the input sum of bits [16], [17]. The 32- bit KS adder is shown in Fig. 4.

The carry equation for proposed SQRT CSLA using KS adder is,

$$G_0 = a_0 \& b_0 \quad (16)$$

$$P_0 = a_0 \text{ xor } b_0 \quad (17)$$

$$C_1 = g_0 + p_0 cin \quad (18)$$

$$C_2 = (g_1 + p_1 g_0) + p_1 p_0 cin \quad (19)$$

$$C_3 = (g_2 + p_2 g_0) + p_2 p_1 c_1 \quad (20)$$

$$C_4 = (g_3 + p_3 g_2) + p_3 p_2 (g_1 + p_1 g_0) + p_3 p_2 p_1 p_0 cin \quad (21)$$

IV. Result and Discussion

We have coded the SQRT CSLA using KS adder design in existing adder design in VERILOG for bit-widths 16. All designs are synthesized in the Xilinx and Modelsim software. As shown in Table I, the SQRT CSLA using KS adder involves significantly less area and less delay than the existing SQRT CSLA designs. Figs. 5-8 show the simulation result of existing and proposed adder design. Figs. 9-11 show the comparison result of area, delay and ADP for existing and presented adder design.

TABLE I
Parameter Comparison on SQRT CSLA and KS Adder

Design	Area (um^2)	Delay (ns)	ADP
SQRT CSLA **B**ased on dual RCA	22	28.058	617.276
SQRT CSLA based on BEC	20	26.981	539.62
Existing SQRT CSLA design	18	25.742	463.356
Proposed SQRT CSLA using KS adder	18	23.496	422.928

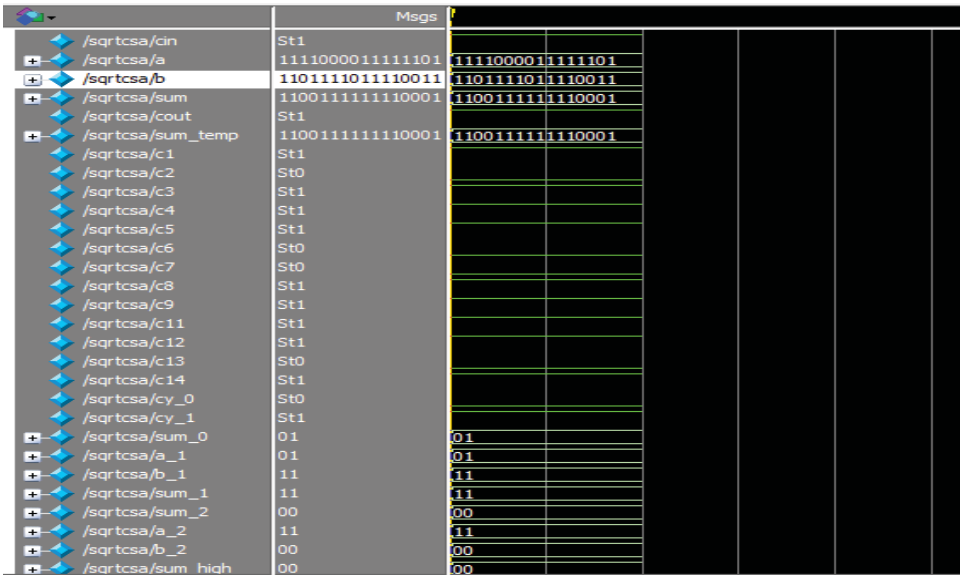

Fig. 5 Simulation output of conventional CSLA

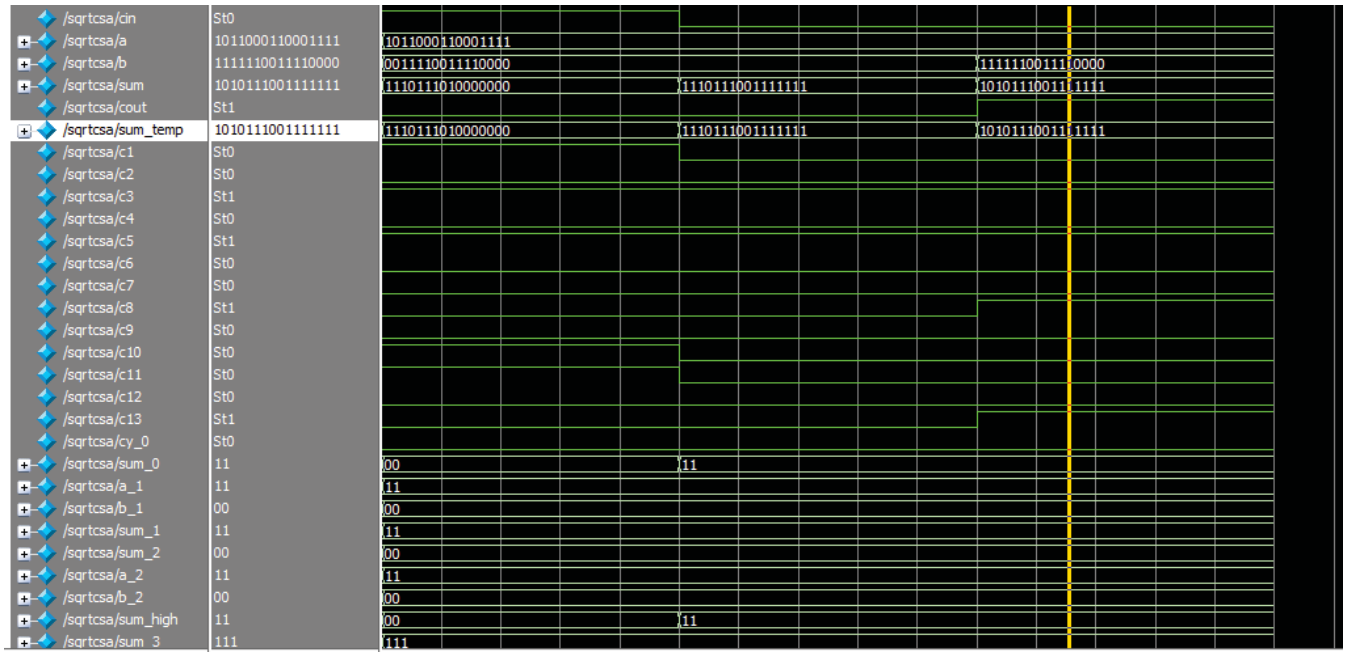

Fig. 6 Simulation output of BEC based CSLA CSLA

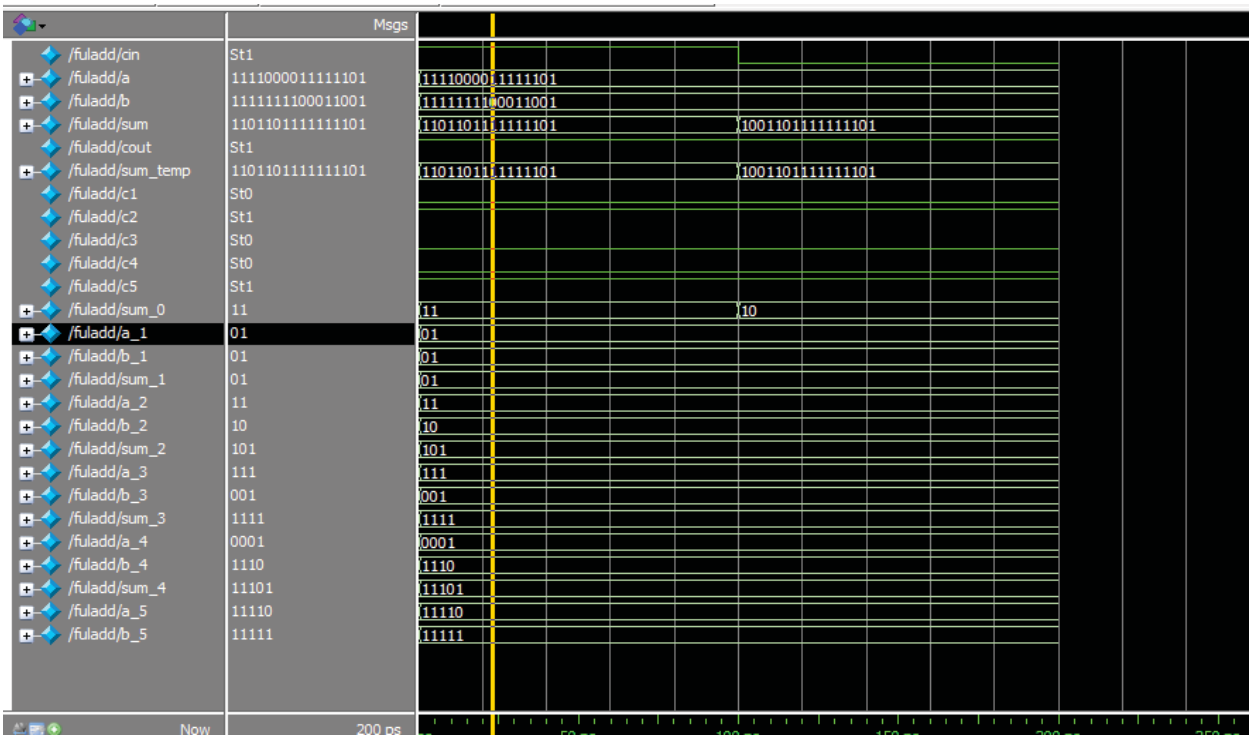

Fig. 7 Simulation output of existing SQRT CSLA Adder design

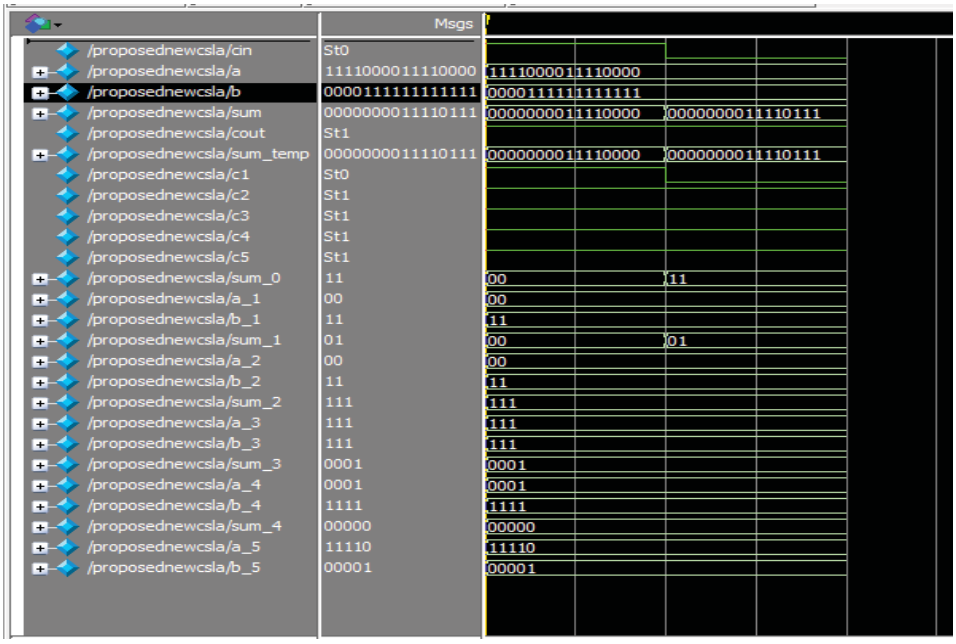

Fig. 8 Simulation output of proposed SQRT CSLA using KS Adder design

V. Conclusion

Digital adder suffers with the problem of CPD. BEC is one of the digital adders in which the delay gets reduced by simple gate level modification but the design is not attractive. To make a design better, an approach is presented in this paper with reduced the delay. The replacement of KS adder in place of RCA offers great advantage in the reduction of delay. Due to the small carry output delay, the presented KS adder design is a good candidate for the SQRT adder. The compared result shows that the existing SQRT-CSLA design involves 11% more ADP and consumes 2% delay than the proposed KS adder, for 16 bit-width. The future work is focused towards the multipliers which can be used in the structure of CSLA design.

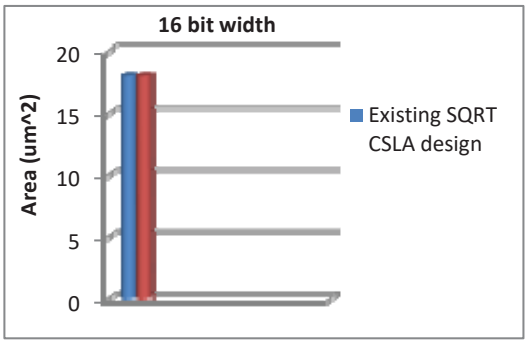

Fig. 9 Comparison of Area Consumption.

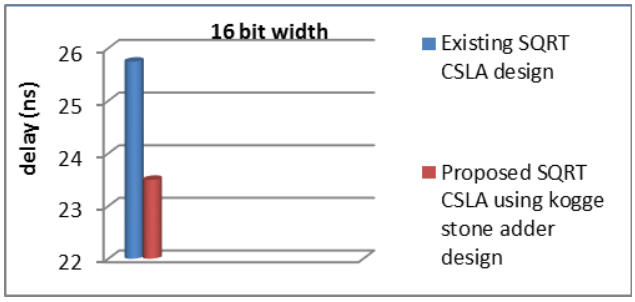

Fig. 10 Comparison of Delay

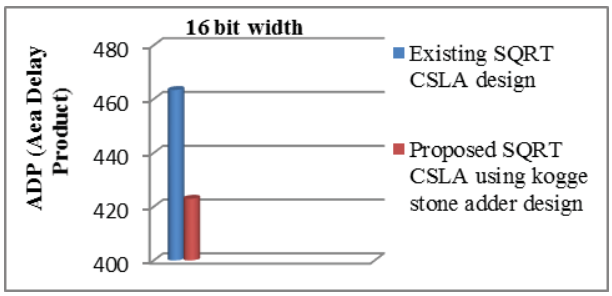

Fig. 11 Comparison of ADP

REFERENCES

[1] Allipeera, K. and Ahmed Basha, S. "An Efficient 64-Bit Carry Select Adder with Less Delay and Reduced Area Application", International Journal of Engineering Research and Applications (IJERA), 2012, vol.2, pp.78-84.
[2] Ramkumar and Kittur, H.M., "Low-power and area-efficient carry-select adder", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.20, 2012, pp.371–375.
[3] Kim, Y. and Kim, L., "64-bit carry-select adder with reduced area", Electron. Lett, 2001, vol. 10, pp.614–615.
[4] He, Y. and Chang, C.H., "An area efficient 64-bit square root carry select adder for low power application", in Proc. IEEE Int. Symp. Circuits Syst, 2005, vol.4, pp.4082–4085.
[5] Wey, I. and Ho, C., "An area-efficient carry select adder design by sharing the common Boolean logic term", in Proc. IMECS, vol.10,2012, pp.1-4.
[6] Manju, S. and Sornagopal, V., "An efficient SQRT architecture of carry select adder design by common Boolean logic", in Proc. VLSI ICEVENT, vol.16, 2013, pp.1-5.
[7] Chaitanya kumara, P. and Nagendra, R., "Design of 32 bit Parallel Prefix Adders", IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), vol.15, 2013, pp.150-160.
[8] Partha Mitra and Debarshi Datta, "Low Power High Speed SQRT Carry Select Adder", IOSR Journal of VLSI and Signal Processing,2013, vol.1, pp.46-51.
[9] Sarabdeep Singh and Dilip Kumar, "Design of area and power efficient modified carry select adder", International journal of computer application, 2011, vol.33.
[10] Ch. Pavan kumar and V. Narayana Reddy, "Design and Implementation of Modified Sqrt Carry Select Adder on FPGA", International Journal of Computer Trends and Technology (IJCTT), 2013, vol.5, pp.63-68.
[11] Pandu Ranga Rao and Priyanka Halle, "An Efficient Carry Select Adder with Less Delay and Reduced Area Application", IJETT, vol.4,2013, pp.3766-3770.
[12] Dr. P. Bhaskara Reddy, S.V.S. Prasad, K. Ananda Kumar, "An Area and Speed Efficient Square Root Carry Select Adder Using Optimized Logic Units", International Journal of Innovative Research in Computer Science & Technology (IJIRCST), Volume-3, Issue-5, 2015.
[13] Pandu Ranga Rao & Priyanka Halle, An Efficient Carry Select Adder with Less Delay and Reduced Area Application, IJETT, volume 4, issue 9,2013.
[14] Gauravkumar D. Jade, Asst. Prof. Prafful Dubey, Prof. Vijay Sharma, "Delay & Area Efficient Carry-Select Adder", IORD Journal of Science & Technology, Volume 2, Issue 2, 2015, pp. 47-52.
[15] Basant Kumar Mohanty & Sujit Kumar Patel, Area–Delay–Power Efficient Carry-Select Adder, IEEE transactions on circuits and systems II: express briefs, vol.61, no.6, 2014.
[16] Pakkiraiah Chakali, Madhu Kumar Patnala, "Design of High Speed Kogge-Stone Based Carry Select Adder", International Journal of Emerging Science and Engineering (IJESE), Volume-1, Issue-4, 2013.
[17] Mohammed Haseena Begum and Vamsi Mohana Krishna, V., "Design and Verification of Low Power and Area Effcient Kogge-Stone Carry Select Adder", International Journal of Engineering Research & Technology (IJERT), vol.8, 2013, pp.463-467.
[18] Gagandeep Singh and Chakshu Goel, "Area Efficient Carry Select Adder (AE-CSLA) using Cadence Tools", International Journal of Engineering Trends and Technology (IJETT), vol.10, 2013, pp.23-26.
[19] Deepthi Obul Reddy and Ramesh Yadav, P., "Carry Select Adder with Low Power and Area Efficiency", IJERD,2012, vol.3, pp.29-35.
[20] Shuchi Verma and Sampath Kumar, "Design & Analysis of Low Power, Area-Efficient Carry Select Adder", Int. Journal of Engineering Research and Applications, 2012, vol.4, pp.53-55.