IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 6, Ver. I (Nov. - Dec. 2016), PP 18-24
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Implementation of New Gate Level Architecture for Carry-Select Adder for better Performance

N.Nimshi[1],J.E.N.Abhilash[2], Dr.B.Subrahmaneswara Rao[3]
Department of ECE, Swarnandhra College of Engineering & Technology, Narsapur, A.P, India

Abstract: *This paper presents the reduction of logic operations involved in conventional carry select adder (CSLA) and binary to excess -1 converter BEC-based CSLA are analyzed to study the data dependence and to identify redundant logic operations. Here eliminated all the redundant logic operations present in the conventional CSLA, BEC- CSLA and proposed a new logic formulation for CSLA. In the proposed scheme, depending on the initial carry the total operation decides the operation of two individual blocks which itself generates the two final sum's and carry individually. Here the main advantage is the output does not require any multiplexer. Only one block it-self works initially on input carry so totally half of the logic implementation is automatically reduced. An efficient CSLA design is obtained using optimized logic units. The proposed CSLA design involves significantly less area and delay than the present BEC-based CSLA. The architecture has been verified on XILINX Spartan - 3E and respective synthesis result shows that the proposed SQRT-CSLA involves nearly 35% less area–delay–product (ADP) and consumes 50% less energy than the proposed SQRT-CSLA, on average, for different bit-widths like 8, 16 and 32 bit widths than the BEC-based SQRT-CSLA.*

Index Terms: *Adder, CSLA (Carry Select Adder),Arithmetic unit,low-power design,BEC (binary to excess -1).*

I. Introduction

Design of area- and power-efficient high-speed data path logic 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. **L**ow-Power, area-efficient, and high-performance VLSI systems are increasingly used in portable and mobile devices, multi-standard wireless receivers, and biomedical instrumentation.

The main component of an arithmetic unit is an adder. A complex digital signal processing (DSP) system involves several adders. An efficient adder design essentially improves the performance of a complex DSP system A ripple carry adder (RCA) uses a simple design, main concern is carry propagation delay (CPD).Carry look-ahead and carry select (CS) methods have been suggested to reduce the CPD of adders .A conventional carry select adder (CSLA) is an RCA–RCA configuration that generates a pair of *sum* words and *output-carry* bits corresponding the anticipated input-carry (c_{in} =0 and 1) and selects one out of each pair for *final-sum* and *final-output-carry*. A conventional CSLA has less CPD compared to RCA, but with usage of dual RCA the design is not attractive.

Few attempts have been made to avoid dual use of RCA in CSLA design. Kim and Kim used one RCA and one add-one circuit which is implemented using a multiplexer(MUX) instead of two RCAs. A square-root (SQRT)-CSLA was proposed to implement large bit-width adders with less delay. In a SQRT CSLA, CSLAs with increasing size are connected in a cascading structure, his design is to provide a parallel path for carry propagation that helpsto reduce the overall adder delay. The suggested BEC-based CSLA involves less logic resources than the conventional CSLA, but it has marginally higher delay.

A CSLA based on common Boolean logic (CBL)is proposed which 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 which requires more logic resource and delay than BEC-based SQRT-CSLA. Here the logic optimization largely depends on availability of BEC-based SQRT-CSLA redundant operations in the formulation, whereas adder delay mainly depends on data dependence. In theexisting designs, logic is optimized without giving any consideration to the data dependence. In this brief, an analysis onlogic operations involved in conventional and BEC-based CSLAs to study the data dependence and to identify redundant logic operations was done. Based on this analysis, a logic formulation for the CSLA is designed.

The main contribution are logic formulation based on data dependence and optimized carry generator (CG) and CS design. Based on the proposed logic formulation, we have derived an efficient logic design for CSLA. Due to optimized logic units, the proposed CSLA involves significantly less ADP than the existing CSLAs. In this brief it is shown that the SQRT-CSLA using the proposed CSLA design involves nearly 32%

less ADP and consumes 33% less energy than that of the corresponding SQRT-CSLA. The rest of this brief is organized as follows. Logic formulation of CSLA is presented in Section II. The proposed CSLA is presented in Section III and the performance comparison is presented in Section V. The conclusion is given in Section VII.

1.1 Delay and Area Evaluation Methodology of the Basicadder Blocks

The AND, OR, and Inverter (AOI) implementation of an XOR gate is shown in Fig. 1.The gates between the dotted lines are performing the operations in parallel and the numeric representation of each gate indicates the delay contributed by that gate. The delay and area evaluation methodology considers all gates to be made up of AND, OR, and Inverter, each having delay equal to 1 unit and area equal to 1 unit. By adding up the number of gates in the longest path of a logic block maximum delay is contributed. The area evaluation is done by counting the total number of AOI gates required for each logic block. Based on this approach, the CSLA adder blocks of 2:1 mux, Half Adder(HA), and FA are evaluated.

1.2 BEC:

The main idea is to use BEC instead ofthe RCA with C_{in} = 1 in order to reduce the area and power consumption of the regular CSLA. To replace the n-bit RCA, an n+1-bit BECis required. The basic function of the CSLA is obtained byusing the 4-bit BEC together with the mux as shown in Fig 2. One input of the 8:4 muxgets as it input (B3, B2, B1, and B0) and another input of the mux is theBEC output. This produces the two possible partial results in paralleland the mux is used to select either the BEC output or the direct inputsaccording to the control signal C_{in}. The importance of the BEC logicstems from the large silicon area reduction when the CSLA with largenumber of bits are designed. The Boolean expressions of the 4-bit BECis listed as (note the functional symbols ~ NOT, & AND, ⊕XOR)

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

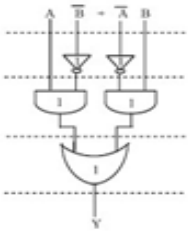


Fig1: Delay and area evaluation of XOR gate

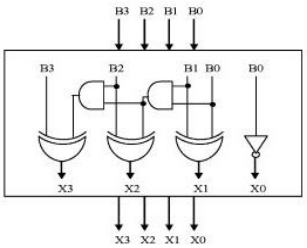


Fig2: 4-b BEC

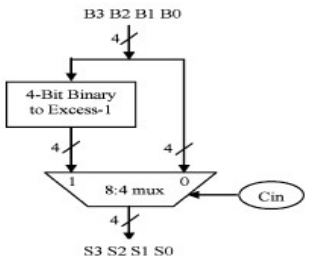


Fig 3:4-b BEC with 8:4 mux

II. Logic Formulation

The CSLA has two units: 1) the *sum* and *carry* generator unit (SCG) and 2) the *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. Here a study of the logic designs are suggested for the SCG unit of conventional and BEC-based CSLAs of by suitable logic expressions. The main objective of this study is to identify redundant logic operations and data dependence. Accordingly, all redundant logic operations and sequence logic operations are removed based on their data dependence.

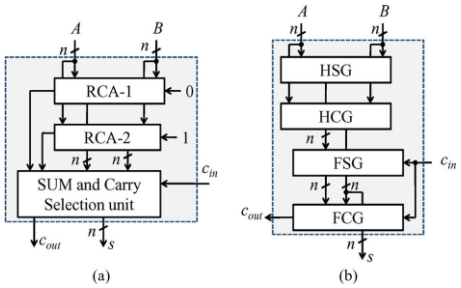


Fig. 4. (a) Conventional CSLA; *n* is the input operand bit-width. (b) The logicoperations of the RCA is shown in split form, where HSG, HCG, FSG, and FCGrepresent half-sum generation, half-carry generation, full-sum generation, andfull-carry generation, respectively

2.1 Logic Expressions of the SCG Unit of theConventional CSLA

As shown in Fig. 4(a), the SCG unit of the conventional CSLA is composed of two *n*-bit RCAs, where *n* is the adder bit-width. The logic operation of the *n*-bit RCA is performed in four stages: 1) *half-sum* generation (HSG); 2) *half-carry* generation (HCG); 3) *full-sum* generation (FSG); and 4) *full carry* generation (FCG)

Suppose two *n*-bit operands are added in the conventional CSLA, then RCA-1 and RCA-2 generate *n*-bit *sum* (s_0 and s_1) and output-carry (c^0_{out} and c^1_{out}) corresponding to input-carry (c_{in}=0 and c_{in}=1),respectively. Logic expressions of RCA-1 and RCA-2 of the SCG unit of the *n*-bit CSLA are given as

$s_0^0(i).=A(i)\oplus B(i) c_0^0(i)=A(i)\cdot B(i)$ (1a)

$s_0^1(i)=s_0^0(i)\oplus c_0^1(i-1)$(1b)

$c_0^1(i) =c_0^0(i) + s_0^0(i) \cdot c_0^1(i-1)$

$c^0_{out}=c_0^1(n-1)$(1c)

$s_1^0(i)=A(i) \oplus B(i)\ c_1^0(i)=A(i)\cdot B(i)$(2a)

$s_1^1(i)=s_1^0(i) \oplus c_1^1(i-1)$(2b)

$c_1^1(i) =c_1^0(i) + s_1^0(i) \cdot c_1^1(i-1)$

$c^1_{out}=c_1^1(n-1)$ (2c)

where$c_0^1(-1) = 0$, $c_1^1(-1) = 1$, and

$0 \le i \le n-1$.

As shown in (1a)–(1c) and (2a)–(2c), the logic expression of $\{ s_0^0 (i), c_0^0 (i)\}$ is identical to that of $\{ s^1_0(i),c^1_0(i)\}$. These redundant logic operations can be removed to have an optimized design for RCA-2, in which the HSG and HCG of RCA-1 is shared to construct RCA-2. Based on this, an add-one circuit is used instead of RCA-2 in the CSLA, in which a BEC circuit is used in for the same purposed. Since the BEC-based CSLA offers the best area–delay–power efficiency among the existing CSLAs, we discuss here the logic expressions of the SCG unit of the BEC-based CSLA as well.

2.2 Logic Expression of the SCG Unit of the BEC-Based CSLA

As shown in Fig.5, the RCA calculates *n*-bit *sum* s_0^1 and c^0_{out} corresponding to c_{in} = 0. The BEC unit receives s_0^1 and c^0_{out} from the RCA and generates (*n* +1)-bit excess-1 code. The most significant bit (MSB) of BEC represents c^1_{out}, in which *n* least significant bits (LSBs) represent s_1^1.

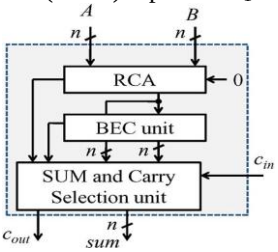


Fig. 5. Structure of the BEC-based CSLA; *n* is the input operand bit-widthof the RCA are the same as those given in (1a)–(1c).

The logicexpressions of the BEC unit of the *n*-bit BEC-based CSLA aregiven as

$s_1^1(0) = s_0^1(0)\ c_1^1(0) = s_0^1(0)$ (3a)

$s_1^1(i) = s_0^1(i) \oplus c_1^1(i-1)$ (3b)

$c_1^1(i) = s_0^1(i) \cdot c_1^1(i-1)$(3c)

$c^1_{out} = c_0^1(n-1) \oplus c_1^1(n-1)$ (3d)

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

We can find from (1a)–(1c) and (3a)–(3d) that, in the case of the BEC-based CSLA, c_1^1depends on s_0^1, which otherwise has no dependence on s_0^1 in the case of the conventional CSLA. The BEC method therefore increases data dependence in the CSLA. The logic expressions of the conventional CSLA are considered and made a further study on the data dependence to find an optimized logic expression for the CSLA .It is interesting to note from (1a)–(1c) and (2a)–(2c) that logic expressions of $s_0^1$and $s_1^1$are identical except the terms$c_0^1$and c_1^1since ($s_0^0 = s_1^0 = s_0$). In addition, we find that $c_0^1$and c_1^1depend on $\{s_0, c_0, c_{in}\}$, where $c_0 = c_0^0 = c_1^0$. Since $c_0^1$and c_1^1 have no dependence on s_0^1 and s_1^1, the logic operation of c_0^1 and c_1^1can be scheduled before $s_0^1$and s_1^1, and the select unit can select one from the set (s_0^1, s_1^1) for the *final-sum* of the CSLA.

We find that a significant amount of logic resource is spent for calculating $\{s_0^1, s_1^1\}$, 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$\{c_0$ and $c_1\}$ to calculate the *final-sum*. The selected carry word is added with the *half-sum* (s_0) to generate the *final-sum* (s).Using this method, one can have three design advantages:1) Calculation of s_0^1 is avoided in the SCG unit; 2) the *n*-bit select unit is required instead of the (n + 1) bit; and 3) small output-carry delay. All these features result in an area–delay and energy-efficient design for the CSLA. We have removed all the redundant logic operations of (1a)–(1c) and (2a)–(2c) and rearranged logic expressions of (1a)–(1c) and (2a)–(2c) based on their dependence. The proposed logic formulation for the CSLA is given as

$s_0(i) = A(i) \oplus B(i) c_0(i) = A(i) \cdot B(i)$ (4a)

$c_0^1(i) = c_0^1(i-1) \cdot s_0(i) + c_0(i)$

for ($c_0^1(0) = 0$)(4b)

$c_1^1(i) = c_1^1(i-1) \cdot s_0(i) + c_0(i)$

for ($c_1^1(0) = 1$)(4c)

$c(i) = c_0^1(i)$ if ($c_{in} = 0$) (4d)

$c(i) = c_1^1(i)$ if ($c_{in} = 1$) (4e)

$c_{out} = c(n-1)$ (4f)

$s(0) = s_0(0) \oplus c_{in}$ $s(i) = s_0(i) \oplus c(i-1)$ (4g)

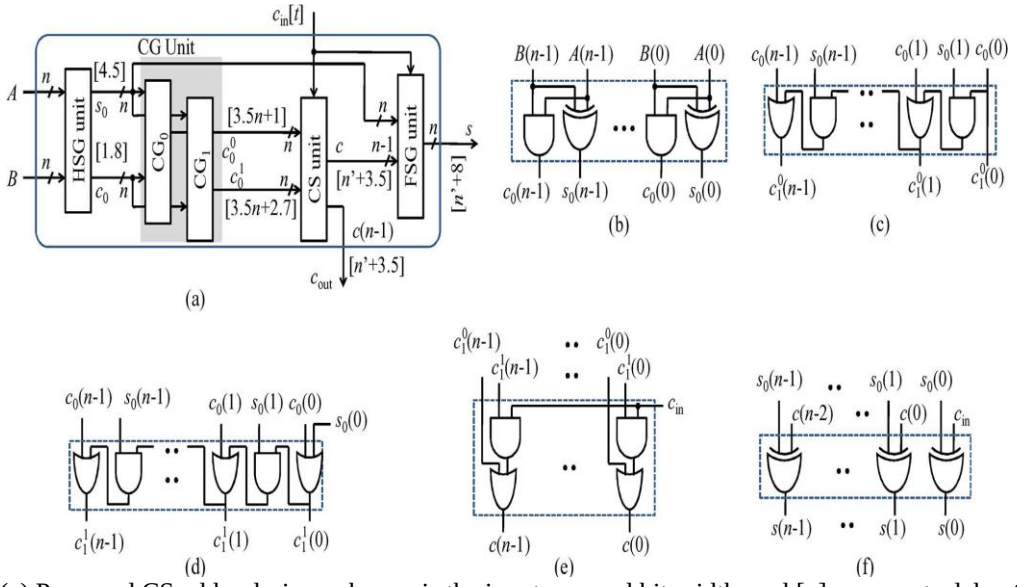


Fig. 6. (a) Proposed CS adder design, where *n* is the input operand bit-width, and [∗] represents delay (in the unit of inverter delay), $n = \max(t, 3.5n + 2.7)$.(b) Gate-level design of the HSG. (c) Gate-level optimized design of (CG_0) for input-carry = 0. (d) Gate-level optimized design of (CG_1) for input-carry = 1(e) Gate-level design of the CS unit. (f) Gate-level design of the final-sum generation (FSG) unit.

III. Proposed Adder Design

The proposed CSLA is based on the logic formulation given in (4a)–(4g), and its structure is shown in Fig. 6(a). It consists of one HSG unit, one FSG unit, one CG unit, and one CS unit. The CG unit is composed of two CGs (CG_0 and CG_1) corresponding to input-carry '0' and '1'. The HSG receives two *n*-bit operands (*A* and *B*) and generate *half-sum* word s_0 and *half-carry* word c_0 of width *n* bits each. Both CG_0 and CG_1 receive s_0 and c_0 from the HSG unit and generate two *n*-bit full-carry words $c_0^1$and c_1^1corresponding to input-carry '0' and '1', respectively. The logic diagram of the HSG unit is shown in Fig. 4(b). The logic circuits of CG_0 and CG_1 are optimized to take advantage of the fixed input-carry bits. The optimized designs of CG_0 and CG_1 are shown in Fig. 6(c) and (d), respectively. The CS unit selects one final carry word from the two carry words available at its input line using the control signal c_{in}. It selects c_0^1 when $c_{in} = 0$; otherwise, it selects c_1^1. 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_0^1And c_1^1follow a specific bit pattern. If $c_0^1(i)$ ='1',then $c_1^1(i)$ =1,irrespective of $s_0(i)$ and $c_0(i)$, for $0 \le i \le n - 1$. This feature is used for logic optimization of the CS unit. The optimized design of the CS unit is shown in Fig. 6(e), which is composed of *n* AND–OR gates. The final carry word *c* is obtained from the CS unit. The MSB of *c* is sent to output as c_{out}, and $(n - 1)$ LSBs are XORed with $(n - 1)$ MSBs of *half-sum* (s_0) in the FSG [shown in Fig. 6(f)] to obtain $(n - 1)$ MSBs of *final-sum*(*s*). The LSB of s_0 is XORed with c_{in} to obtain the LSB of *s*.

IV. Simulation Results

Simulation results of Proposed carry select adder are as follows

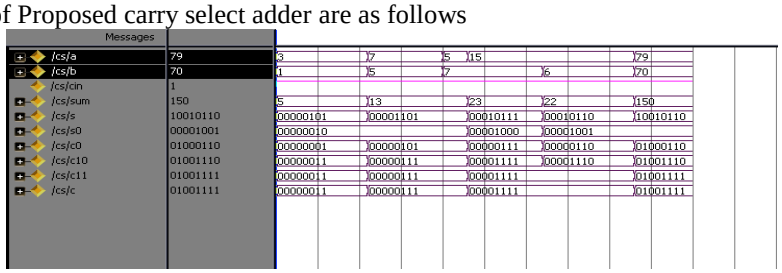


Fig 7: Simulation result for Proposed CS8

From the above simulation result of proposed carry select adder 8 bit the operands a,b are inputs with c_{in}whereas c10&c11 are carry outputs when c_{in} =0 or c_{in}=1.The final sum and carry are given by s &c. As we are adding two 8 bits we obtain 16bits as output with one carry bit.

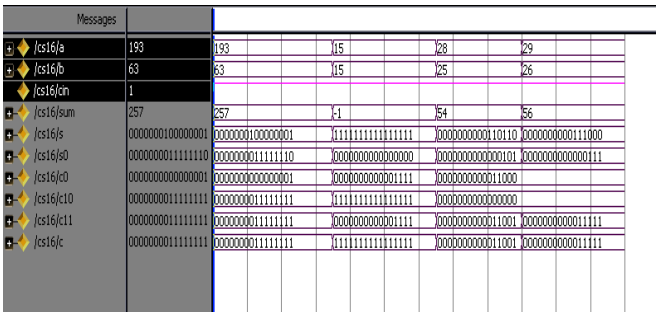


Fig 8: Simulation result for Proposed CS16

From the above simulation result of proposed carry select adder 16 bit the operands a,b are inputs with c_{in}whereas c10&c11 are carry outputs when c_{in} =0 or c_{in}=1.The final sum and carry are given by s &c. As we are adding two 16 bits we obtain 32 bits as output with one carry bit.

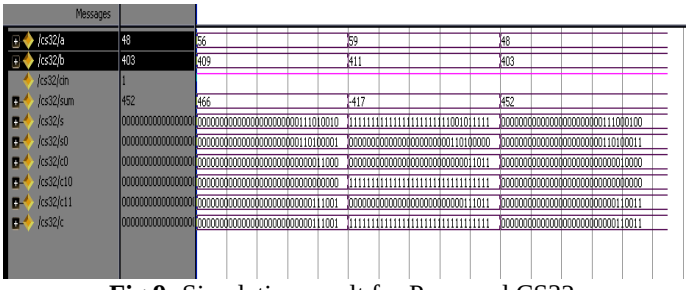


Fig 9: Simulation result for Proposed CS32

From the above simulation result of proposed carry select adder 32 bit the operands a,b are inputs with c_{in}whereas c10&c11 are carry outputs when c_{in} =0 or c_{in}=1.The final sum and carry are given by s &c. As we are adding two 32 bits we obtain 64 bits as output with one carry bit.

V. Performance Comparison

The above architecture has been simulated and synthesized on FPGAXC3S400K using XILINX ISE-12.4 tool.The performance of three proposed adders are evaluated and they are implemented using VHDL

Device Utilization Summary (estimated values) for 8bit:

Table-1

S.No	Logic Utilization	Available	RCA	BEC	PROPOSED CS
			Used	Used	Used
1.	No of slices	3584	16	15	7
2.	No of LUTS	7168	19	17	13
3.	No of IOBS	97	26	26	26
4.	Maximum delay(ns)		13.02	12.71	8.24

Device Utilization Summary (estimated values) for 16 bit:

Table-2

S.No	Logic Utilization	Available	RCA	BEC	PROPOSED CS
			Used	Used	Used
1.	No of slices	3584	28	26	17
2.	No of LUTS	7168	46	46	32
3.	No of IOBS	97	50	50	50
4.	Maximum delay(ns)		19.59	16.42	10.43

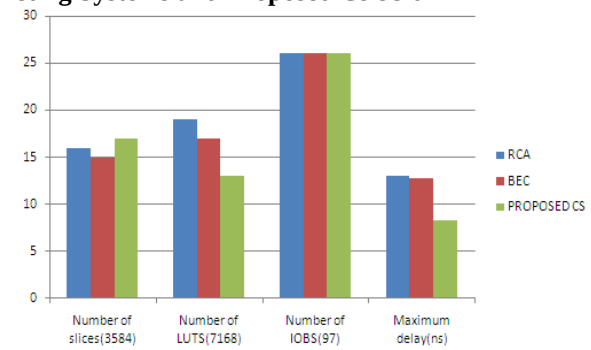
Device Utilization Summary (estimated values) for 32bit:

Table-3

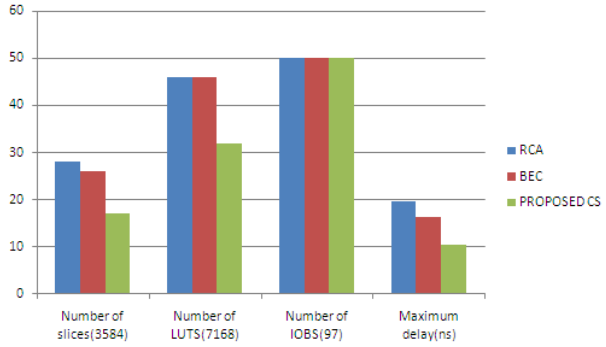
S.No	Logic Utilization	Available	RCA	BEC	PROPOSED CS
			Used	Used	Used
1.	No of slices	3584	58	51	39
2.	No of LUTS	7168	94	92	69
3.	No of IOBS	97	98	98	98
4.	Maximum delay(ns)		49.78	29.5	25.27

VI. Performance Comparision

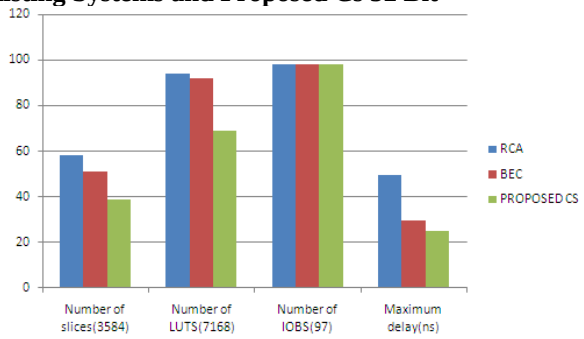
1. Comparision between Existing Systems and Proposed Cs 8bit



2. Comparision between Existing Systems and Proposed Cs 16 Bit



3. Comparision between Existing Systems and Proposed Cs 32 Bit



VII. Conclusion

Thus in order to reduce the area and power of modified SQRT CSLA architecture that we have implemented in this paper, a simple approach has been used. In this work, the numbers of gates have been reduced and this feature offers a greater advantage in the area and power reduction.The simulation results indicate that the modified SQRT CSLA is suffering from larger delay whereas in the 32-bit Proposed carry select adder, area and power are significantly reduced. The delay calculations used here can be computed using the XILINX tool. Due to the small carry output delay,the proposed CSLA design is a good candidate for the SQRT adder. The architecture has been verified on XILINX Spartan - 3E and respective synthesis result shows that the proposed SQRT-CSLA involves nearly 35% less area–delay–product (ADP) and consumes 50% less energy than the proposed SQRT-CSLA, on average, for different bit-widths like 8, 16 and 32 bit widths than the BEC-based SQRT-CSLA

References

[1]. S.Manju and V. Sornagopal, "An efficient SQRT architecture of carry selectadder design by common Boolean logic," in *Proc. VLSI ICEVENT*, 2013,pp. 15.
[2]. M. Z. Rahman and L. Kleeman, "A delay matched approach for the design of asynchronous sequential circuits,"Dept. Comput. Syst.Technol., Univ. Malaya, Kuala Lumpur, Malaysia, Tech. Rep. 05042013,2013.
[3]. B. Ramkumar and H.M. Kittur, "Low-power and area-efficient carry-select adder," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 2,pp. 371–375, Feb. 2012.
[4]. I.-C. Wey, C.-C. Ho, Y.-S. Lin, and C. C. Peng, "An area-efficient carryselect adder design by sharing the common Boolean logic term," in *Proc.IMECS*, 2012, pp. 1–4.
[5]. B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*,2nd ed. New York, NY, USA: Oxford Univ. Press, 2010
[6]. R. F. Tinder, *Asynchronous Sequential Machine Design and Analysis:A Comprehensive Development of the Design and Analysis of Clock-Independent State Machines and Systems*. San Mateo, CA, USA:Morgan, 2009.
[7]. A. P. Chandrakasan, N. Verma, and D. C. Daly, "Ultralow-power electronics for biomedical applications," *Annu. Rev. Biomed. Eng.*, vol. 10, pp. 247–274, Aug. 2008.
[8]. P. Choudhury, S. Sahoo, and M. Chakraborty, "Implementation of basic arithmetic operations using cellular automaton," in *Proc. ICIT*, 2008,pp. 79–80.
[9]. C. Cornelius, S. Koppe, and D. Timmermann, "Dynamic circuit techniques in deep submicron technologies: Domino logic reconsidered," in *Proc. IEEE ICICDT*, Feb. 2006, pp. 1–4.
[10]. Y. He, C. H. Chang, and J. Gu, "An area-efficient 64-bit square root carryselect adder for low power application," in *Proc. IEEE Int. Symp. CircuitsSyst.*, 2005, vol. 4, pp. 4082–4085
[11]. D. Geer, "Is it time for clockless chips? [Asynchronous processor chips]," *IEEE Comput.*, vol. 38, no. 3, pp. 18–19,Mar. 2005.
[12]. N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and SystemsPerspective*. Reading, MA, USA: Addison-Wesley, 2005.
[13]. M. D. Riedel, "Cyclic combinational circuits," Ph.D. dissertation,Dept. Comput. Sci., California Inst. Technol., Pasadena, CA, USA,May 2004.
[14]. M. Anis, S. Member, M. Allam, and M. Elmasry, "Impact of technology scaling on CMOS logic styles," *IEEE Trans. CircuitsSyst., Analog Digital Signal Process.*, vol. 49, no. 8, pp. 577–588,Aug. 2002.
[15]. 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.
[16]. F.-C. Cheng, S. H. Unger, and M. Theobald, "Self-timed carry look ahead adders," *IEEE Trans. Comput.*, vol. 49, no. 7, pp. 659–672,Jul. 2000.
[17]. K. K. Parhi, *VLSI Digital Signal Processing*. New York, NY, USA:Wiley,1998.
[18]. W. Liu, C. T. Gray, D. Fan, and W. J. Farlow, "A 250-MHz wavepipelined adder in 2-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 29,no. 9, pp. 1117–1128, Sep. 1994.
[19]. S. Nowick, "Design of a low-latency asynchronous adder using speculativecompletion," *IEE Proc. Comput. Digital Tech.*, vol. 143, no. 5,pp. 301–307, Sep. 1996.
[20]. O. J. Bedrij, "Carry-select adder," *IRE Trans. Electron. Comput.*,vol. EC-11, no. 3, pp. 340–344, Jun. 1962.