Design and Implementation of Efficient Carry Select Adder using Novel Logic Algorithm

V. Thamizharasi
Senior Grade Lecturer, Department of ECE, Government Polytechnic College, Trichy, India

***Abstract*: Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. The logic operations involved in conventional carry select adder (CSLA) and binary to excess-1 converter (BEC)-based CSLA are data dependence and redundant logic operations. In this paper, proposed a new logic formulation for CSLA to eliminate the redundant logic operations present in the conventional CSLA. In the proposed scheme, the carry select (CS) operation is scheduled before the calculation of final-sum, which is different from the conventional approach. The proposed CSLA design has reduced area and delay as compared with BEC-based CSLA. Due to the small carry-output delay, the proposed CSLA design is suitable for square-root (SQRT) CSLA. The performance of the CSLA is evaluated by implementing a MAC unit by using conventional, BEC and proposed CSLA in the adder part, for different bit -widths. This work focuses on the performance of CSLA in terms of area, delay and power, the analysis of the result shows that the proposed SQRT-CSLA has less area-delay product and has less power-delay product, when compared with existing SQRT-adders. The system has been designed efficiently using Verilog HDL codes and simulated using Quartus II 9.1 and hardware implementation is done by using Altera-FPGA.**

***Keywords*: CSLA, arithmetic unit, low power, area efficient.**

1. Introduction

Design of area and power-efficient high-speed data path logic systems are one of the most areas of research in VLSI system design are increasingly used in portable and mobile devices, multi standard wireless receivers, and biomedical instrumentation [10], [11]. An adder is the main component of an arithmetic unit. A complex digital signal processing (DSP) system involves several adders. An efficient adder design essentially improves the performance of a DSP system. Some other applications of adders are in Multiply – Accumulate (MAC) unit. Adders are also used in multipliers, in high speed integrated circuits and in digital signal processing to execute various algorithms like FFT, IIR and FIR. A ripple carry adder (RCA) uses a simple design, but carry propagation delay (CPD) is the main concern in this adder. Carry look-ahead and carry select (CS) adders 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 input-carry (cin = 0 and 1) and selects one out of each pair for final-sum and final-output-carry A conventional CSLA has less CPD than an RCA, but the design is not attractive since it uses a dual RCA. Few attempts have been made to avoid dual use of RCA in CSLA design. Kim and Kim [12] used one RCA and one add-one circuit instead of two RCAs, where the add-one circuit is implemented using a multiplexer (MUX). He et al. [13] 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. Ramkumar and Kittur [14] suggested a binary to BEC-based CSLA. The BEC-based CSLA involves less logic resources than the conventional CSLA, but it has marginally higher delay.

The logic operations involved in conventional carry select adder (CSLA) and binary to excess-1 converter (BEC)-based CSLA are data dependence and to identify redundant logic operations. In this paper, proposed a new logic formulation for CSLA to eliminate the redundant logic operations present in the conventional CSLA. In the proposed scheme, the carry select (CS) operation is scheduled before the calculation of final-sum, which is different from the conventional approach. The performance of the CSLA is evaluated by implementing a MAC unit by using conventional, BEC and proposed CSLA in the adder part. This work focuses on the performance of CSLA in terms of area, delay and power, the analysis of the result shows that the proposed SQRT-CSLA has less area-delay product and has less power-delay product, when compared with existing SQRT-adders.

This is briefed as follows. First present the detailed structure and the function of conventional CSLA, BEC based CSLA design. The conventional CSLA has been chosen for comparison with the BEC based CSLA and proposed CSLA design. The delay and power evaluation methodology of the conventional, BEC and proposed SQRT CSLA are presented. Therefore the main aim of the project is to design and implement a high speed carry select adder to enhance the speed of addition and perform fast arithmetic functions. The proposed design is applied to the MAC unit structure in the adder part to evaluate the performance of the proposed design. This work estimates that the proposed CSLA has less area-delay product and power-delay product than existing CSLAs.

2. CSLA

A. Conventional CSLA

The ripple carry adder is composed of cascaded full adders for 4-bit adder, as shown in Fig. 1. It is constructed by cascading full adder blocks in series. The carry out of one stage is fed directly to the carry-in of the next stage. For an n-bit parallel adder it requires n full adders, the critical path is n-bit carry propagation path in the full-adders. As the bit number n increases, the delay time of carry ripple adder will increase accordingly in a linear way. In order to improve the shortcoming of carry ripple adder to remove the linear dependency between computation delay time and input word length, carry select adder is presented. The CSLA has two units sum and carry generator unit (SCG) and the sum and carry selection unit is shown in Fig. 2. The SCG unit of the conventional CSLA [3] is composed of two n-bit RCAs, where n is the adder bit-width. 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. The main objective is to identify redundant logic operations and data dependence. Accordingly, remove all redundant logic operations and sequence logic operations based on their data dependence.

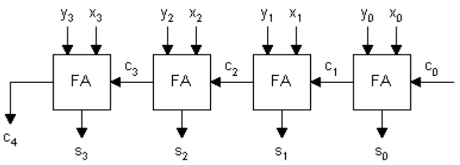


Fig. 1. 4-bit Ripple Carry Adder

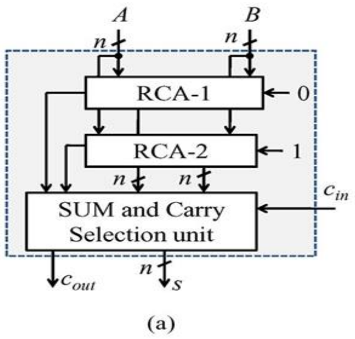


Fig. 2. n-bit conventional CSLA

B. Logic Expressions of the SCG unit of the conventional CSLA

As shown in Fig. 1. The logic operation of the n-bit RCA is performed by full adder generation. Suppose two n-bit operands are added in the conventional CSLA, then RCA-1 and RCA-2 generate n-bit sum (s0 and s1) and output-carry (c0out and *c*1out) 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

$$s_1^0(i) = A(i) \wedge B(i) \wedge c_1^0(i-1) \tag{1a}$$

$$c_1^0(i) = \left(A(i) \cdot B(i)\right) | (B(i) \cdot c_1^0(i-1)) | (A(i) \cdot c_1^0(i-1)) \quad c_{out}^0 = c_1^0(n-1) \tag{1b}$$

$$s_1^1(i) = A(i) \wedge B(i) \wedge c_1^1(i-1) \tag{2a}$$

$$c_1^1(i) = \left(A(i) \cdot B(i)\right) | (B(i) \cdot c_1^1(i-1)) | (A(i) \cdot c_1^1(i-1)) \quad c_{out}^1 = c_1^1(n-1) \tag{2b}$$

Where $c_1^0(-1) = 0, c_1^1(-1) = 1$ *and* $0 \leq i \leq n-1$.

These redundant logic operations can be removed and provide an optimized design for RCA-2, in which the HSG and HCG of RCA-1 is shared to design RCA-2.

C. BEC based CSLA

The conventional CSLA is not area efficient and delay for the operation is more because it uses multiple set 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. To avoid this problem, the regular CSLA structure is modified using n-bit Binary to Excess-1 code converter (BEC) to improve the speed of operation. To improve the speed of operation use the Binary to Excess-1 Converter (BEC) instead of RCA with Cin=1 in the regular CSLA to achieve less delay. A structure and the function table of a 4-b BEC function of the CSLA is obtained by using the 4-bit BEC together with the mux. One input of the 8:4 mux gets as it input (B3, B2, B1, and B0) and another input of the mux is the BEC output. This produces the two possible partial results in parallel and the mux is used to select either the BEC output or the direct inputs according to the control signal Cin. The importance of the BEC logic stems from the large silicon area reduction when the CSLA with large number of bits are designed.

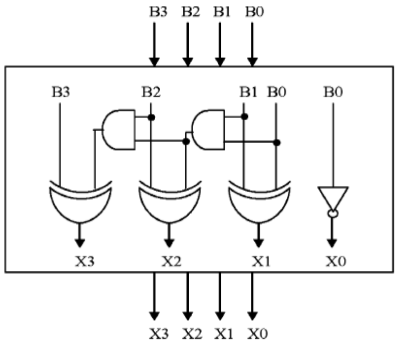


Fig. 3. 4-bit BEC

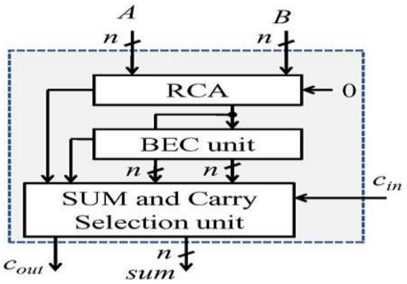


Fig. 4. N-bit CSLA using BEC.

D. Logic Expressions of the SCG Unit of the BEC based CSLA

The n-bit CSLA using BEC is shown in figure 2, the RCA calculates n-bit sum $s_0^1$and c^0out corresponding to cin = 0. The BEC unit receives $s_0^1$and c^0out from the RCA and generates (n

+ 1)-bit excess-1 code. The most significant bit (MSB) of BEC represent c^0out , in which n least significant bits (LSBs) represent s_1^1.The logic expressions of the RCA are the same as those given in (1a)–(1c). The logic expressions of the n-bit BEC-based CSLA are given as

$$s_1^1(0) = \sim s_1^0(0) \quad c_1^1(0) = s_1^0(0) \tag{3a}$$

$$s_1^1(i) = s_1^0(i) \wedge c_1^1(i-1) \tag{3b}$$

$$c_1^1(i) = s_1^0(i) \,.\, c_1^1(i-1) \tag{3c}$$

$$c_{out}^1 = c_1^0(n-1) \wedge c_1^1(n-1) \tag{3d}$$

for $1 \leq i \leq n-1$

From the above equation find that, the BEC-based CSLA, c_1^1depends on s_1^0 , 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.Considered the logic expressions of the conventional CSLA and a data dependence to find an optimized logic expression for the CSLA. The significant amount of logic resource is spent for calculating {s_0^1 and 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 $\{c^0, c^1\}$ 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, one can have three design advantages: 1) Calculation of s_1^0 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 power efficient design for the CSLA.

E. Proposed CSLA design

The proposed logic formulation for the CSLA is given as

$$s_0(i) = A(i) \wedge B(i) \qquad c_0(i) = A(i) \cdot B(i) \tag{4a}$$

$$c_1^0(i) = c_1^0(i-1) \,.\, s_0(i) + c_0(i)\, for\, (c_1^0(0) = 0) \tag{4b}$$

$$c_1^1(i) = c_1^1(i-1) \,.\, s_0(i) + c_0(i)\, for\, (c_1^1(0) = 1) \tag{4c}$$

$$c(i) = c_1^0(i) \text{ if } (c_{in} = 0) \tag{4d}$$

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

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

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

The proposed CSLA is based on the logic formulation given in (4a)–(4g), and its structure is shown in Fig. 5. 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_1^0 and c_1^1 corresponding to input-carry '0' and '1', respectively. The logic diagram of the HSG unit is shown in Fig. 6(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 cin. It selects c_1^0 when cin = 0; otherwise, it selects c_1^1 . The CS unit can be implemented using an n-bit 2-to-l MUX. However, find from the truth table of the CS unit that carry words c_1^0 and c_1^1 follow a specific bit pattern. If $c_1^0(i) = '1'$, then $c_1^1(i) = '1'$, irrespective of s_0 (i)and c_o(i), for $0 \leq i \leq n-1$. 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 cout, and (n − 1) LSBs are XORed with (n − 1) MSBs of half-sum (s0) in the FSG [shown in Fig. 6(f)] to obtain (n − 1) MSBs of final-sum (s). The LSB of s0 is XORed with cin to obtain the LSB of s.

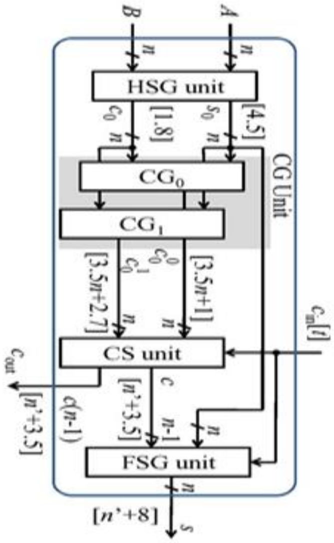


Fig. 5. n-bit proposed CSLA design

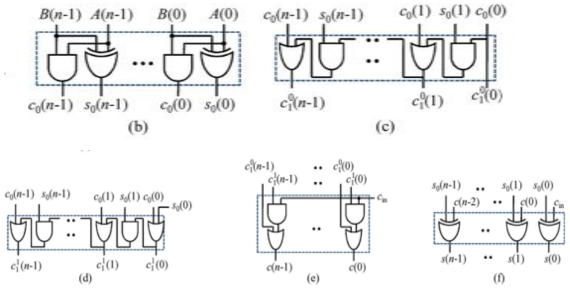


Fig. 6. b) Gate-level design of the HSG. (c) Gate-level optimized design of (CG0) for input-carry = 0. (d) Gate-level optimized design of (CG1) for input-carry = 1. (e) Gate-level design of the CS unit. (f) Gate-level design of the final-sum generation (FSG) unit.

F. Single-stage CSLA

The single- stage carry select adder is constructed by connecting a number of equal length adder stages. For an n-bit adder, it could be implemented with equal length of carry select adder and is called as single –stage carry select adder.The proposed CSLA is based on the logic formulation and its structure 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_1^0 and c_1^1 corresponding to input-carry ‘0’ and ‘1’, respectively. The logic circuits of CG_0 and CG_1 are optimized to take advantage of the fixed input-carry bits. The CS unit selects one final carry word from the two carry words available at its inputs using the control signal cin. It selects c_1^0 when cin = 0; otherwise, it selects c_1^1 . The CS unit can be implemented using an n-bit 2-to-l MUX. However find from the truth table of the CS unit that carry words c_1^0 and c_1^1 follow a specific bit pattern. If $c_1^0\,(i) =' 1'$, 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. 3(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 cout, and (n − 1) LSBs are XORed with (n − 1) MSBs of half-sum (s0) in the FSG to obtain (n − 1) MSBs of final-sum (s). The LSB of s0 is XORed with cin to obtain the LSB of s.

G. *Multistage CSLA (SQRT-CSLA)*

The multipath carry propagation feature of the CSLA is fully exploited in the SQRT-CSLA, which is composed of a chain of CSLAs. CSLAs of increasing size are used in the SQRT-CSLA to extract the maximum concurrence in the carry propagation path. Using the SQRT-CSLA design, large-size adders are implemented with significantly less delay than a single-stage CSLA of same size. However, carry propagation delay between the CSLA stages of SQRT-CSLA is critical for the overall adder delay. Due to early generation of output-carry with multipath carry propagation feature, the proposed CSLA design is more suitable than the existing CSLA designs for area–delay efficient implementation of SQRT-CSLA. A 16-bit SQRT-CSLA design using the proposed CSLA is shown in Fig. 4, where the 2-bit RCA, 2-bit CSLA, 3-bit CSLA, 4-bit CSLA, and 5-bit CSLA are used. Considered the cascaded configuration for different bit-widths. To demonstrate the advantage of the proposed CSLA design in SQRT-CSLA, estimated the area and delay of SQRT-CSLA using the proposed CSLA design and the BEC-based CSLA of [6] for bit-widths 8, 16, 32.

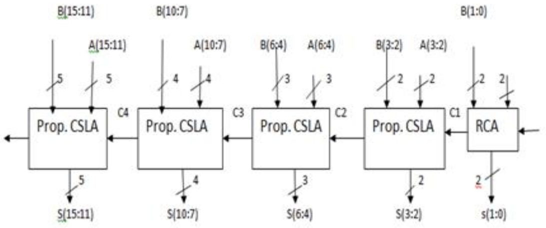


Fig. 7. Proposed SQRT-CSLA for n = 16

H. *Implementation of MAC unit using CSLA*

Adders play an important part in today’s digital signal processing (DSP) systems. So need to design high speed, area efficient adders. The performance of the CSLA is evaluated by implementing an MAC unit using the Conventional CSLA, BEC based CSLA and Proposed Carry Select Adder and then comparing the results in terms of area, delay and power.

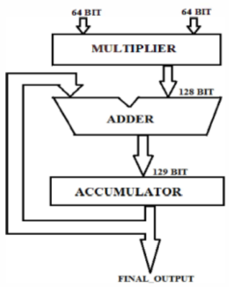


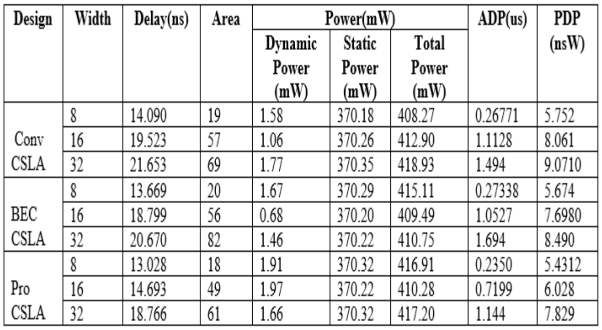
Fig. 8. MAC unit Architecture.

The adder part is replaced with the conventional, BEC and proposed CSLA and the MAC unit is implemented and the performance of the design is evaluated in terms of area, delay and power. In this paper use a MAC unit implementation using conventional, BEC and proposed CSLA. The MAC unit is implemented using different the carry select adders and then the results are compared. The performance of the CSLA is evaluated in terms of area, delay and power.

3. Table

Comparison of Area, Delay, Power of Existing and Proposed for Single- Stage CSLA are given in Table1.The given Table 1 shows that the proposed CSLA involves nearly 27% less area and 1.3% less delay than the conventional CSLA. Consequently, the conventional CSLA of involves 24% higher ADP and 13% higher PDP than the proposed CSLA, for 32 bit-width. Compared with the BEC-based CSLA, the proposed CSLA design has marginally less ADP. However, in the BEC-based CSLA, delay increases at a much higher rate than the proposed CSLA design for higher bit widths. Compared with the BEC based CSLA, the proposed CSLA involves 1.6% more power, but it involves nearly 26% less ADP and nearly 3% less PDP due to less area complexity. Power consumption is more in proposed single- stage CSLA when it compared with the existing CSLAs, to reduce the power consumption go for multi-stage CSLA.

Table 1
Comparison of Area, Delay, and Power of Existing and Proposed for Single- Stage CSLA



Design	Width	Delay(ns)	Area	Power(mW)			ADP(us)	PDP (nsW)
				Dynamic Power (mW)	Static Power (mW)	Total Power (mW)		
Conv CSLA	8	14.090	19	1.58	370.18	408.27	0.26771	5.752
	16	19.523	57	1.06	370.26	412.90	1.1128	8.061
	32	21.653	69	1.77	370.35	418.93	1.494	9.0710
BEC CSLA	8	13.669	20	1.67	370.29	415.11	0.27338	5.674
	16	18.799	56	0.68	370.20	409.49	1.0527	7.6980
	32	20.670	82	1.46	370.22	410.75	1.694	8.490
Pro CSLA	8	13.028	18	1.91	370.32	416.91	0.2350	5.4312
	16	14.693	49	1.97	370.22	410.28	0.7199	6.028
	32	18.766	61	1.66	370.32	417.20	1.144	7.829

A comparison of conventional, BEC and proposed CSLA is made in terms of delay, area and power and listed in the below table.

Table 2
Comparison of Area, Delay, and Power of Existing and Proposed for Multi- Stage CSLA.

DESIGN	WIDTH	DELAY(ns)	AREA	POWER(mW)	ADP(us)	PDP(psW)
Conv CSLA	8	14.090	19	1.58	0.26771	22.26
	16	16.872	41	1.78	0.6917	30.03
	32	21.434	87	1.05	1.8647	22.50
BEC CSLA	8	13.669	20	1.67	0.27338	22.32
	16	14.171	37	1.21	0.5243	17.14
	32	15.458	88	0.52	1.3603	8.03
Pro CSLA	8	13.028	18	1.91	0.2350	24.88
	16	14.547	30	0.73	0.4364	10.61
	32	14.218	67	0.28	0.9526	3.98

Comparisons of Area, Delay, Power of Existing and Proposed for multi- Stage CSLA are given in Table 2.The given Table 2 shows that the proposed CSLA involves nearly 20% less area and 3% less delay and 5% less power than the conventional CSLA. Consequently, the conventional CSLA of involves 27% higher ADP and 19% higher PDP than the proposed CSLA, for 32 bit-width. Compared with the BEC-based CSLA, the proposed CSLA design has marginally less ADP. However, in the BEC-based CSLA, delay increases at a much higher rate than the proposed CSLA design for higher bit widths. Compared with the BEC based CSLA, the proposed CSLA involves 1.6% more power, but it involves nearly 26% less ADP and nearly 3% less PDP due to less area complexity. The area, delay and power of an proposed CSLA is reduced and therefore say that proposed CSLA is the High Speed Carry Select Adder. Now evaluate the performance of the Carry Select Adder by implementing an MAC unit using the Conventional, BEC and Proposed CSLA.

Table 3
Comparison of MAC Unit Implementation with Conventional, BEC and Proposed CSLA

DESIGN	WIDTH	DELAY(ns)	AREA	POWER(mW)	ADP(us)	PDP(psW)
Conv CSLA	16	6.491	37	7.88	0.240	51.1
	32	8.039	84	10.61	0.675	85.2
BEC CSLA	16	5.508	36	8.63	0.198	47.5
	32	5.731	76	10.80	0.435	61.8
Pro CSLA	16	5.468	32	8.09	0.174	44.2
	32	5.559	69	9.47	0.383	52.6

From the above comparison results listed in the table, says that the area, delay and power are reduced when the MAC unit is implemented with a proposed CSLA rather than an MAC unit that was implemented with a Conventional CSLA. Thus a high speed, area efficient and low power MAC unit can be designed using a proposed CSLA. Thus the proposed Carry Select Adder is a High Speed Carry Select Adder. The Conventional, BEC and Proposed Carry Select Adder designed are now implemented using an FPGA.The source code is dumped into the FPGA and the results are checked. The steps involved in the implementation of the CSLA using the FPGA are: first synthesize the code, generate programming file, create user constraints file (UCF) file by configuring the input and output pins of FPGA, create a cdc file, run the cdc file, make pin connections including clock. Now switch on the Altera FPGA kit and configure the target device and finally analyze the design using Chip scope Pro Analyzer. This way the high speed CSLA is implemented using the FPGA.

4. Figures

Comparison of area-delay-product and power-delay-product of existing and proposed for multi-stage CSLA to be shown in below graph.

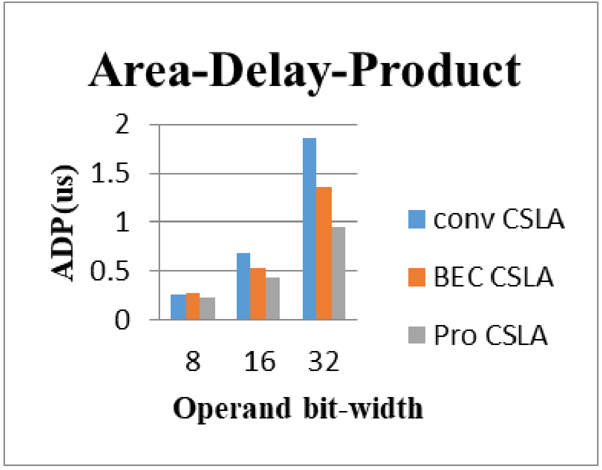


Fig. 9. The simulation results of the area-delay- product comparison in the conventional, BEC, Proposed SQRT-CSLA

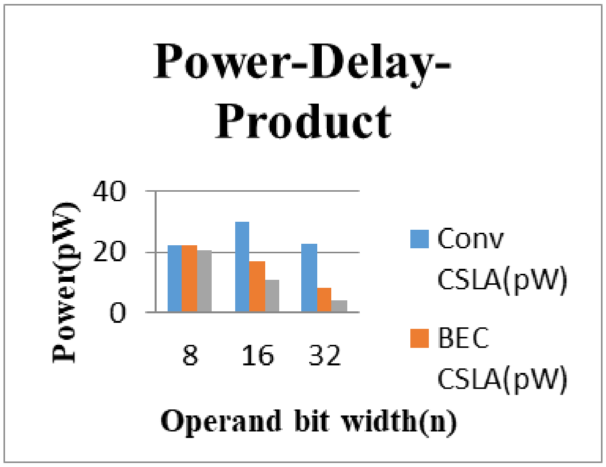


Fig. 10. The simulation results of the power-delay- product comparison in the conventional, BEC, Proposed SQRT-CSLA.

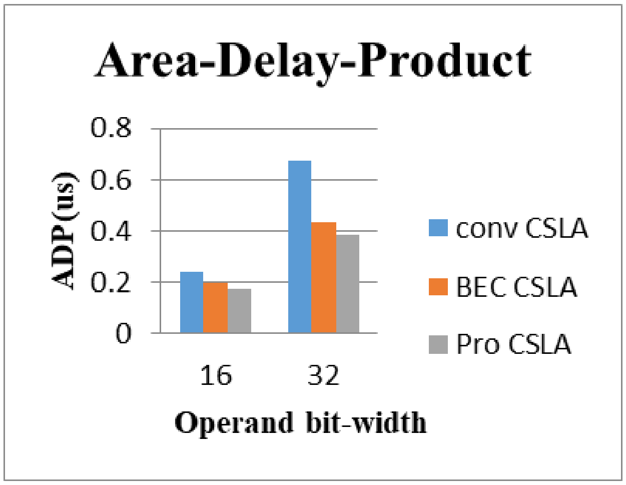


Fig. 11 The simulation results of MAC unit area- delay- product comparison using conventional, BEC, Proposed CSLA

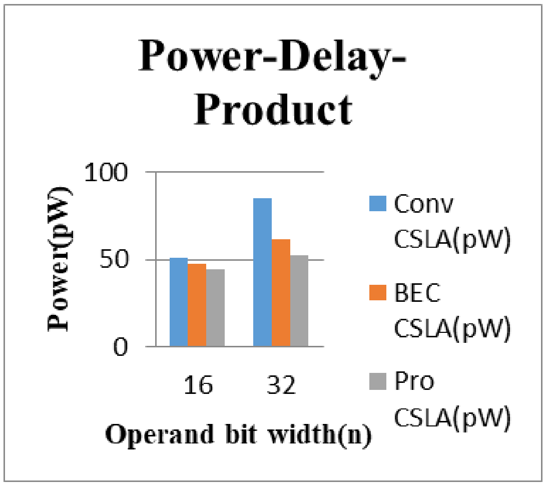


Fig. 12. The simulation results of MAC unit power- delay- product comparison using conventional, BEC, Proposed CSLA.

5. Conclusion

In this paper, analyzed the logic operations involved in the conventional and BEC-based CSLAs are data dependence and redundant logic operations. In this paper, proposed a new logic formulation for CSLA to eliminate the redundant logic operations present in the conventional CSLA. In the proposed scheme, the CS operation is scheduled before the calculation of final-sum, which is different from the conventional approach. The proposed CSLA design involves significantly less area and delay than the recently proposed BEC-based CSLA. Due to the small carry output delay, the proposed CSLA design is suitable for the SQRT adder. All the three models of SQRT- CSLA are designed and are implemented in Verilog HDL using Quartus II 9.1 tool and the results are compared in terms of area, delay and power. The Proposed CSLA proves that has 12% less area-delay-product and 9% less power- delay-product than existing CSLAs .It is also implemented with Altera FPGA. The performance of this CSLA in terms of delay and power is evaluated by implementing an MAC unit by using the CSLA in the adder part and again it proves to be the High Speed and Low Power CSLA. Thus a high speed and low power MAC unit can be designed using Proposed CSLA. The Proposed CSLA architecture is therefore, high speed, low power and area efficient for VLSI hardware implementation.

References

[1] B.Ramkumar, (2010) Harish M Kittur, P.Mahesh Kannan, “ASIC Implementation of Modified Faster Carry Save Adder”, European Journal of Scientific, Vol. 42 No.1, pp.53-58.

[2] Hasan Krad and Aws Yousif Al-Taie, “Performance Analysis of a 32-Bit Multiplier with a Carry-Look-Ahead Adder and a 32-bit Multiplier with a Ripple Adder using VHDL”, Journal of Computer Science 4 (4): 305-308, 2008.

[3] Romana Yousuf and Najeeb-Ud-Din "Synthesis Of Carry Select Adder In 65 Nm Fpga" Tencon 2008-2008 IEEE Region 10 Conference.

[4] Samiappa Sakthikumaran I, S. Salivahanan, V. S. Kanchana Bhaaskaran J., V. Kavinilavu, B. Brindha And C. Vinoth "A Very Fast And Low Power Carry Select Adder Circuit" IEEE Electronics Computer Technology (ICECT), 2011 3rd International Conference (Volume: 1).

[5] V.G. Oklobdzija, “High-Speed VLSI Arithmetic Units: Adders and Multipliers”, in “Design of High-Performance Microprocessor Circuits”, Book edited by A. Chandrakasan, IEEE press, 2000.

[6] V.G. Oklobdzija, “High-Speed VLSI Arithmetic Units: Adders and Multipliers”, in “Design of High-Performance Microprocessor Circuits”, Book edited by A.Chandrakasan, IEEE press, 2000.

[7] Padma Devi, Ashima Girdher, Balwinder Singh, (2010) “Improved Carry Select Adder with Reduced Area and Low Power Consumption”, International Journal of Computer Applications Volume 3 – No.4.

[8] B. Parhami, (2010) Computer Arithmatic: Algorithms and hardware designs, 2nd Edition, Oxford University Press, New York.

[9] N. Vijayabala and T. S. Saravana Kumar, (July 2013)” Area minimization of carry select adder using boolean algebra” International Journal of Advances in Engineering & Technology.

[10] Hiroaki Suzuki, Woopyo Jeong, and Kaushik Roy (2004) “Low-Power Carry-Select Adder Using Adaptive Supply Voltage Based on Input Vector Patterns” pg no 313 to 318

[11] K. K. Parhi, VLSI Digital Signal Processing. New York, NY, USA: Wiley, 1998.

[12] O. J. Bedrij, “Carry-select adder,” IRE Trans. Electron. Comput, vol. EC-11, no. 3, pp. 340–344, Jun. 1962.

[13] 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.

[14] 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. Circuits Syst., 2005, vol. 4, pp. 4082–4085.

[15] 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.

[16] 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.

[17] 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.