



International Journal of Engineering Research and Science & Technology

ISSN : 2319-5991
Vol. 4, No. 2
May 2015



www.ijerst.com

Email: editorijerst@gmail.com or editor@ijerst.com

Research Paper

DESIGN CARRY SELECT ADDER WITH D-LATCHSatish Kumar Patnayak^{1*}, M Raja² and D Sailaja³*Corresponding Author: **Satish Kumar Patnayak** ✉ satishkumarpatnayak@gmail.com

The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. However, the CSLA is not time efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input, then the final sum and carry are selected by the multiplexers (mux). The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA in the regular CSLA to achieve high speed and low power consumption. At the same time to further reduce the power consumption, a new approach of CSLA with D LATCH is proposed in this project. In the proposed scheme, the Carry Select (CS) operation is scheduled before the calculation of final sum, which is different from the conventional approach. Bit patterns of two anticipating carry words (corresponding to $c_{in} = 0$ and 1) and fixed c_{in} bits are used for logic optimization of Carry selection. An efficient CSLA design is obtained using optimized logic units. The proposed CSLA design involves significantly less area and power than the recently proposed BEC-based CSLA.

Keywords: CSLA, RCA, BEC, D-LATCH**INTRODUCTION**

Due to the rapid growth of portable electronic component the low power arithmetic circuit has become very important in VLSI industry. Multiplier-Accumulator (MAC) unit is the main building block in DSP processor. Full Adder is a part of the MAC unit which can significantly affect the efficiency of whole system. Hence the reduction of power consumption of Full Adder circuit is necessary for low power application. The basic operation Carry Select Adder (CSLA) is parallel computation. CSLA generates many carries and partial sum. The final sum and carry are selected

by multiplexers. In the CSLA architecture, Addition usually impacts widely the overall performance of digital systems and a crucial arithmetic function. In electronic applications adders are most widely used. In 2002, a new concept of hybrid adders is presented to speed up addition process by Wang *et al.* that gives hybrid carry look-ahead/carry select adders design. In 2008, low power multipliers based on new hybrid full adders is presented. In digital adders, the speed of addition is having the limitation by the time required to propagate a carry through the adder. Design of area and power optimized data path

¹ M.Tech Student, Department of ECE, Sree Kavitha Engineering College, Khammam, Telangana 507 112, India.² Assistant Professor, Department of ECE, Sree Kavitha Engineering College, Khammam, Telangana 507 112, India.³ Associate Professor, Department of ECE, Sree Kavitha Engineering College, Khammam, Telangana 507 112, India.

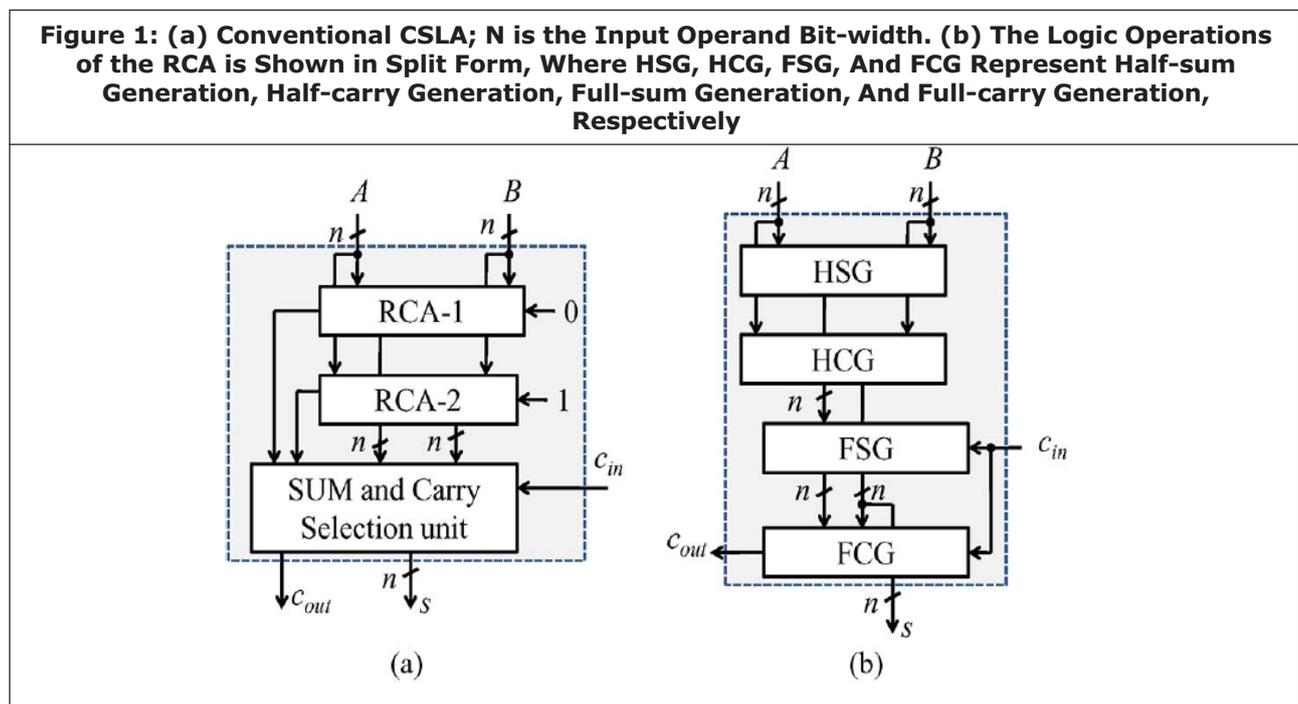
logic systems are one of the most important areas of research in VLSI system design. In electronic system and applications adders are mostly used. Wherever concept of multiplication comes adders come in to the picture. As we know millions of instructions per second are performed in microprocessors. So, the device area and power consumption is the most important constraint to be considered while designing multipliers and adders for those microprocessors. Due to device portability miniaturization the area and power consumption should be low. Devices like Mobile, Laptops, etc., require more battery backup. So, a VLSI designer has to optimize these parameters in a design. These constraints are not that much easy to achieve. So, depending on demand or application some compromise between constraints has to be made. Concatenating the N full adders forms N bit Ripple carry adder. In this carry out of previous full adder becomes the input carry for the next full adder. It calculates sum and carry according to the following equations. As carry ripples from one full

adder to the other, it traverses longest critical path and exhibits worst case delay.

REGULAR CSLA

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. We made a study of the logic designs 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, we remove all redundant logic operations and sequence logic operations based on their data dependence.

As shown in Figure 1(a), the SCG unit of the conventional CSLA (Ceiang, 1998) 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 (s0 and s1) and output-carry (c_{out}^0 and c_{out}^1) 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

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

$$s_1^0(i) = s_0^0(i) \oplus c_1^0(i-1) \quad \dots(1b)$$

$$c_1^0(i) = c_0^0(i) + s_0^0(i) \cdot c_1^0(i-1) \quad c_{out}^0 = c_1^0(n-1) \quad \dots(1c)$$

$$s_0^1(i) = A(i) \oplus B(i) \quad c_0^1(i) = A(i) \cdot B(i) \quad \dots(2a)$$

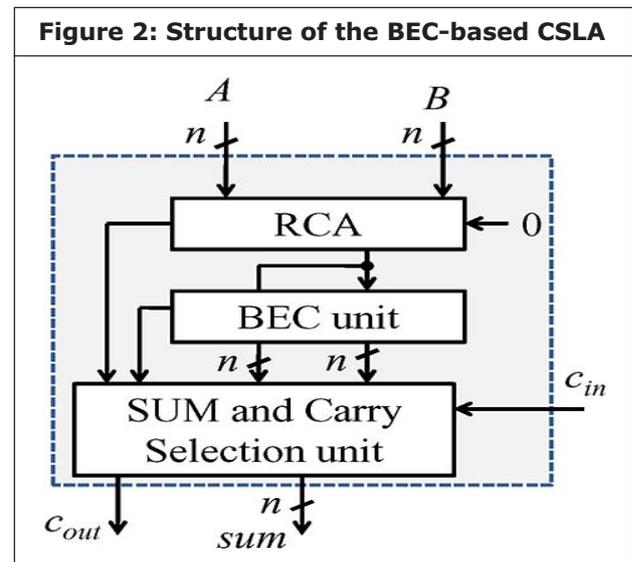
$$s_1^1(i) = s_0^1(i) \oplus c_1^1(i-1) \quad \dots(2b)$$

$$c_1^1(i) = c_0^1(i) + s_0^1(i) \cdot c_1^1(i-1) \quad c_{out}^1 = c_1^1(n-1) \quad \dots(2c)$$

where $c_1^0(-1) = 0$, $c_1^1(-1) = 1$, and $0 \leq i \leq n - 1$. 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 $\{s_0^1(i), c_0^1(i)\}$. 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, Ramkumat *et al.* (2010) and Rabaey (2001) have used an add-one circuit instead of RCA-2 in the CSLA, in which a BEC circuit is used in Abu-Sharma (1995) for the same purpose. 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.

As shown in Figure 2, the RCA calculates n-bit sum s_1^0 and c_{out}^0 corresponding to $c_{in} = 0$. The BEC unit receives s_1^0 and c_{out}^0 from the RCA and generates (n + 1)-bit excess-1 code. The Most Significant Bit (MSB) of BEC represents c_{out}^1 , in which n Least Significant Bits (LSBs) represent s_1^1 .



n is the input operand bit-width of the RCA are the same as those given in (1a)–(1c). The logic expressions of the BEC unit of the n-bit BEC-based CSLA are given as

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

$$s_1^1(i) = s_1^0(i) \oplus c_1^1(i-1) \quad (3b)$$

$$c_1^1(i) = s_1^0(i) \cdot c_1^1(i-1) \quad (3c)$$

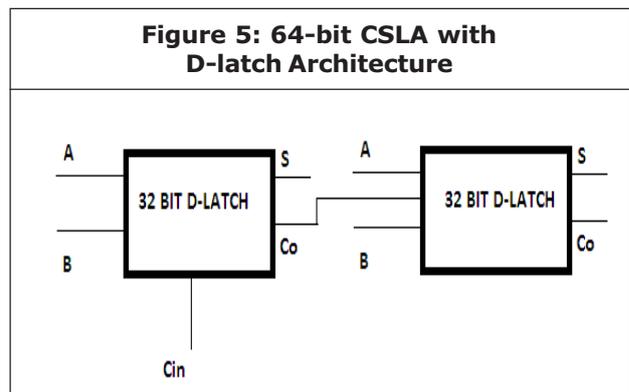
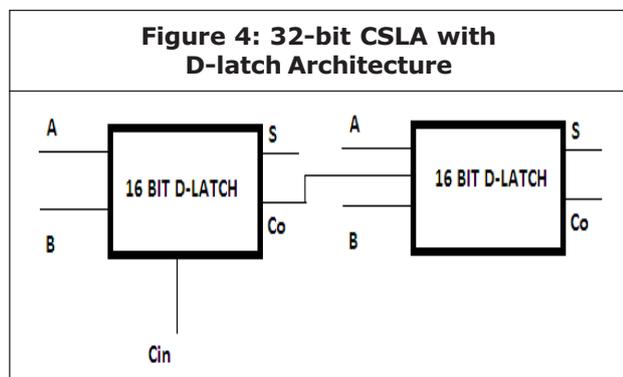
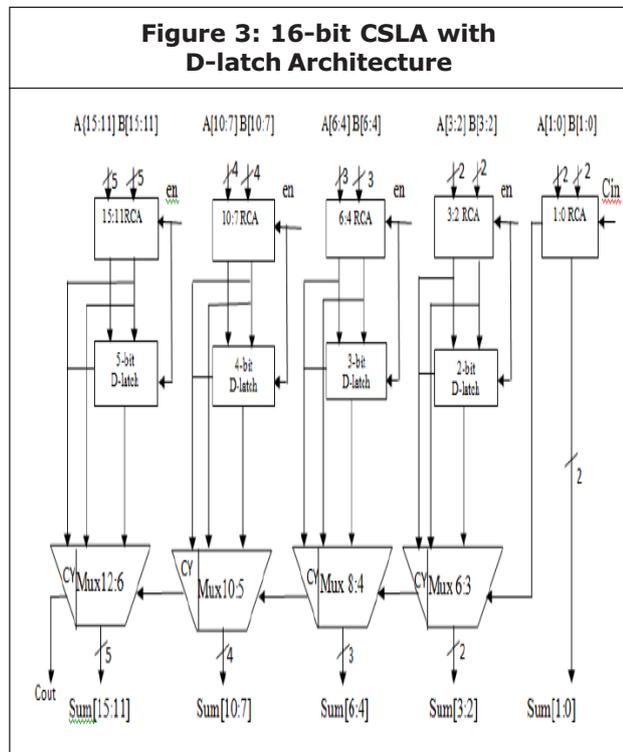
$$c_{out}^1 = c_1^0(n-1) \oplus c_1^1(n-1) \quad (3d)$$

for $1 \leq i \leq n - 1$. We can find from (1a)–(1c) and (3a)–(3d) that, in the case of the BEC-based CSA, c_1^1 depends on s_1^0 , which otherwise has no dependence on s_1^0 in the case of the conventional CSLA. The BEC method therefore increases data dependence in the CSLA. We have considered logic expressions of the

conventional CSLA and made a further study on the data dependence to find an optimized logic expression for the CSLA.

CSLA USING D-LATCH LOGIC

This method replaces the BEC circuit by D-latch. Latches are used to store one bit information. Their outputs are constantly affected by their inputs. In other words, when they are enabled, their content changes immediately according to their inputs. The architecture of proposed 16-bit CSLA is shown in Figure 3. It has different five



groups of different bit size RCA and D-Latch. Instead of using two separate adders in the regular CSLA, in this method only one adder is used to reduce the area, power consumption. Each of the two additions is performed in one clock cycle. This is 16-bit adder in which Least Significant Bit (LSB) adder is ripple carry adder, which is 2 bit wide. The upper half of the adder, i.e., most significant part is 14-bit wide which works according to the clock. Whenever clock goes high addition for carry input one is performed.

When clock goes low then carry input is assumed as zero and sum is stored in adder itself. From the figure it can understand that latch is used to store the sum and carry for $c_{in}=1$ and $c_{in}=0$. Carry out from the previous stage, i.e., least significant bit adder is used as control signal for multiplexer to select final output carry and sum of the 16-bit adder. If the actual carry input is one, then computed sum and carry latch is accessed and for carry input zero MSB adder is accessed is the output carry.

The architectures of the 16-bit, 32-bit and 64-bit CSLA with D-LATCH are shown in Figures 3, 4 and 5. The 64-bit CSLA with D-LATCH Architecture is designed based on the cascading of two 32-bit Architectures. In the D-LATCH architecture first stage is designed based on Ripple Carry adders and the second stage is designed based on the D-LATCH logic.

WORKING PRINCIPLE OF CSLA ADDER USING D LATCH

The bits from a and b (i.e., $a[1:0]$ and $b[1:0]$) as inputs to (1:0) RCA along with c_{in} as input. The bits from a and b (i.e., $a[3:2]$ and $b[3:2]$) as inputs to (3:2) RCA along with e_n . When $e_n=1$, the output of the (3:2) RCA is fed as input to the (2 bit) D-Latch and the output of the (2 bit) D-latch follows the input and given as an input to the (6:3) multiplexer.

When $e_n=0$, the last state of the (2 bit) D input is trapped and held in the latch and therefore the output from the (3:2) RCA is directly given as an input to the (6:3) multiplexer without any delay. Now the (6:3) multiplexer selects the sum bit according to the carry generated from (1:0) RCA which takes c_{in} as input carry and it is the selection bit and the inputs of the (6:3) multiplexer are the outputs obtained when $e_n=1$ and 0.

The bits from a and b (i.e., $a[6:4]$ and $b[6:4]$) are given as inputs to (6:4) RCA along with e_n as carry. When $e_n=1$, the output of the (6:4) RCA is fed as input to the (3 bit) D-Latch and the output of the (3 bit) D-latch follows the input and given as an input to the (8:4) multiplexer. When $e_n=0$, the last state of the (3 bit) D input is trapped and held in the latch and therefore the output from the (6:4) RCA is directly given as an input to the (8:4) multiplexer without any delay. Now the (8:4) multiplexer selects the sum bit according to the carry generated from (6:3) multiplexer which is the selection bit and the inputs of the (8:4) multiplexer are the outputs obtained when $e_n=1$ and 0.

The bits from a and b (i.e., $a[10:7]$ and $b[10:7]$) are given as inputs to (10:7) RCA along with e_n as carry. When $e_n=1$, the output of the (10:7) RCA

is fed as input to the (4 bit) D-Latch and the output of the (4 bit) D-latch follows the input and given as an input to the (10:5) multiplexer. When $e_n=0$, the last state of the (4 bit) D input is trapped and held in the latch and therefore the output from the (10:7) RCA is directly given as an input to the (10:5) multiplexer without any delay.

Now the (10:5) multiplexer selects the sum bit according to the carry generated from (8:4) multiplexer which is the selection bit and the inputs of the (10:5) multiplexer are the outputs obtained when $e_n=1$ and 0.

The bits from a and b (i.e., $a[15:11]$ and $b[15:11]$) are given as inputs to (15:11) RCA along with e_n as carry. When $e_n=1$, the output of the (15:11) RCA is fed as input to the (5 bit) D-Latch and the output of the (5 bit) D-latch follows the input and given as an input to the (12:6) multiplexer. When $e_n=0$, the last state of the (5 bit) D input is trapped and held in the latch and therefore the output from the (15:11) RCA is directly given as an input to the (12:6) multiplexer without any delay. Now the (12:6) multiplexer selects the sum bit according to the carry generated from (10:5) multiplexer which is the selection bit and the inputs of the (12:6) multiplexer are the outputs obtained when $e_n=1$ and 0.

RESULTS

In this paper we propose D-LATCH based CSLA architecture which makes use of Sequential component called D-Latch, which has the capability of holding the previous value. The inputs to this design are A, B of 64 bit width, C_i of single bit width. As it is a sequential component, it requires clk as input. For example, if $A = 15$, $B = 30$ and $C_i = 0$, then the output SUM will be 45 and C_o will be 0.

Figure 6: RTL Schematic View of 64 bit CSLA with D-Latch

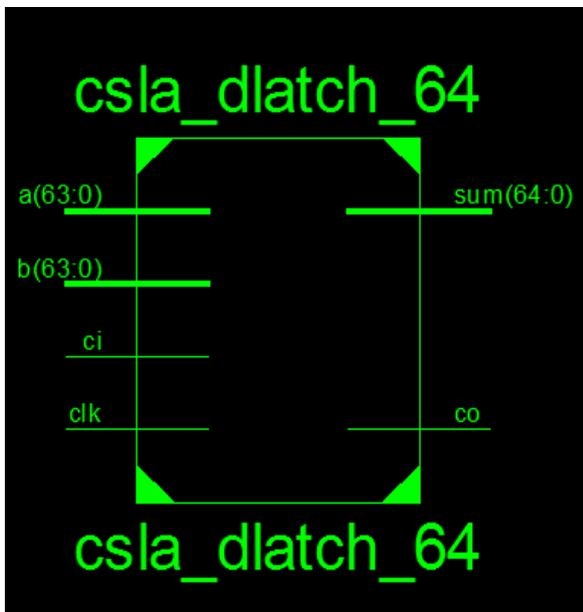


Figure 7: Waveform of 64 bit CSLA with D-Latch



Table 1: Comparison Table of Different CSLA's

	16 bit RCA	16 bit BEC	16 bit D LATCH	32 bit RCA	32 bit BEC	32 bit D LATCH	64bit RCA	64 bit BEC	64 bit D LATCH
Number of 4 input LUTs used	43	48	32	102	101	134	204	201	285
Available LUTs	9312	9312	9312	9312	9312	9312	9312	9312	9312
DELAY ns	14.362	14.73	12.211	16.117	21.756	16.506	27.539	37.772	18.873
Total supply power mw	76	76	76	76	76	76	76	76	76
Power Utilized mw	0.351	0.392	0.039	0.832	0.823	1.092	1.663	1.638	2.3235

CONCLUSION

A D-LATCH based CSLA architecture is proposed in this paper to reduce the area and power consumption of CSLA architecture than the recently proposed BEC based CSLA architecture. The reduction in area and power consumption parameters by this work offers the great advantage in the field of designing low power based portable devices. The functionality verification of the design is carried out by using ISE Simulator and the synthesis is also carried out by the XILINX ISE 12.3i. The HDL used for obtaining an RTL schematic and for designing the modules is VHDL. From the graphs and the tables it is concluded that, the proposed D-LATCH based design consumes less power and less area when compare to the BEC based and RCA based architectures.

REFERENCES

1. Ramkumar B and Harish M Kittur (2011), "Low Power and Area Efficient Carry Select Adder", *IEEE transactions on very large scale integration (VLSI) systems*.
2. Bedrij O J (1962), "Carry-select adder," *IRE Trans. Electron. Comput.*, pp. 340–344 .
3. Ceiang T Y and Hsiao M J (Oct. 1998), "Carry-select adder using single ripple carry adder," *Electron. Lett.*, Vol. 34, No. 22, pp. 2101-2103.
4. Ramkumar B, Kittur H M and Kannan P M (2010), "ASIC implementation of modified faster carry save adder," *Eur. J. Sci. Res.*, Vol. 42, No. 1, pp. 53-58.
5. Rabaey J M (1995), *Digital Integrated Circuits—A Design Perspective*, Upper Saddle River, NJ: Prentice-Hall, 2001.

6. Abu-Shama E and Bayoumi M (1995), "A New cell for low power adders," in *Proc.Int. Midwest Symp. Circuits and Systems*, pp. 1014-1017.
7. Kim Y and Kim L S (2001), "64-bit carry-select adder with reduced area," *Electron. Lett.*, Vol. 37, No. 10, pp. 614–615.
8. He Y, Chang C H, and Gu J (2005), "An area efficient 64-bit square root carry-select adder for low power applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, Vol . 4, pp. 4082-4085.
9. Oklobdzija V G (2000), "High Speed VLSI Arithmetic Units: Adders and Multipliers", in A Chandrakasan (Ed.), *Design of High Performance Microprocessor Circuits*, IEEE Press.



International Journal of Engineering Research and Science & Technology

Hyderabad, INDIA. Ph: +91-09441351700, 09059645577

E-mail: editorijerst@gmail.com or editor@ijerst.com

Website: www.ijerst.com

