

# High Speed Non-Linear Carry Select Adder

D.Srimathi<sup>1</sup>, G.N.Jayabhavani<sup>2</sup>

<sup>1</sup> M.E, Applied Electronics, IFET College Of Engineering, Tamilnadu, India

<sup>2</sup> Assistant Professor,ECE , IFET College Of Engineering,Tamilnadu, India

\*\*\*

**Abstract** - The role of adder in addition process is significant. The addition operation can be performed by numerous adder. Nowadays the modern world requires fast adders. This requirement can be fulfilled by Carry select adder. The main operation of carry select adder depends upon the input carry value. The structure of CSLA can be altered to improve its performance. It can execute quick arithmetic operations. The Modified Non-linear (NON-LINEAR) carry select adder (CSLA) is formed by altering the structure of Conservative Non-linear carry select adder. The carry selection and final sum generation process is different when compared to the Conservative Non-linear carry select adder(CSLA) and Non-linear binary to excess-1 convertor(BEC) carry select adder. In the proposed work the utilization of logical elements and delay is comparatively less. Due to this the performance of adder is also improved.

**Key Words:** Adder, Non-linear binary to excess-1 convertor, Modified Non-linear carry select adder, Conservative Non-linear(NON-LINEAR) carry select adder, carry select adder(CSLA)

## I.INTRODUCTION

Adders are used in numerous fields. The requirement of digital world is simple, fast, low cost design of adders. The efficiency of adder can be improved by increasing its performance. The ripple carry adder (RCA) can be formed by placing the full adders in series manner. Its operation fully depends upon the previous stage carry value. It consumes more time to compute an operation. It is one of the disadvantages of Ripple carry adder. To overcome this problem a new adder has been designed which is known as Conservative carry select adder. The ripple carry adder act as a building blocks of conservative carry select adder. Its working principle also depend upon the input carry value. The working of Conservative carry select adder is clearly explained in paper[1].In which 3-bit adder requires 3 input. The first 2 input values are

acquired from A and B. While the input carry value will act as a third input. Based on the third input value the sum and carry out is produced. Even though it provide a better results It consists of several redundant operations .To eliminate this problem conservative Non-linear carry select adder is designed. The propagation delay indicate that the time taken to reach the data from one block to another block which can be achieved with the help of Non-linear CSLA so it is well suited for higher order bit widths. This process can be clearly explained in paper[2]. Even though conservative Non-linear carry select adder is better than conservative carry select adder its architecture is not desirable. Again to improve the performance of adder a new adder is proposed which is known as Non-linear BEC carry select adder. The throughput of Non linear BEC carry select adder has less delay. The efficiency can be improved by using the Non-linear architecture in BEC carry select adder[3].In the proposed scheme the number of XOR gates used has been reduced and the carry has been selected earlier to the sum production which reduces the delay. When compared to the existing system less logical operations is used in proposed scheme. The paper is structured as following below. The explanation of Conventional carry select adder is provided in section II. The information about non linear BEC carry select adder is demonstrated in Section III. The detail about the proposed work is exhibited in Section IV .The Test results were shown in section V. Section VI tells about the conclusion and future work.

## II. CONSERVATIVE NON-LINEAR CSLA

The ripple carry adder and multiplexer act as building blocks of Conservative Non-linear carry select adder. The structure of ripple carry adder is shown below in which series of 3-bit adder is used. The architecture composed of two ripple carry adder (RCA) and multiplexer in a parallel structure. It consists of two stages each stage produces the sum and carry out When the input carry value =0 the first stage sum will be produced as a final sum and its carry is considered as a output carry. Suppose the input carry value is 1 then its sum and carry is considered as a final output. The Conservative carry select adder acquires

more number of combinational elements so it is considered as a less significant.

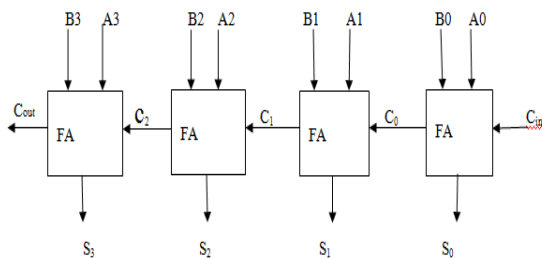


Fig.1.1 Structure of Ripple carry adder

Fig 1.1 symbolize the structure of Ripple carry adder (RCA).In the below equation S represents Sum while C represents Carry.

$$S = A \oplus B \oplus C_{in} \quad (1)$$

$$C = (A \cdot B) + C_{in}(A+B) \quad (2)$$

The equation (1) and (2) elucidate the formation of sum and carry.

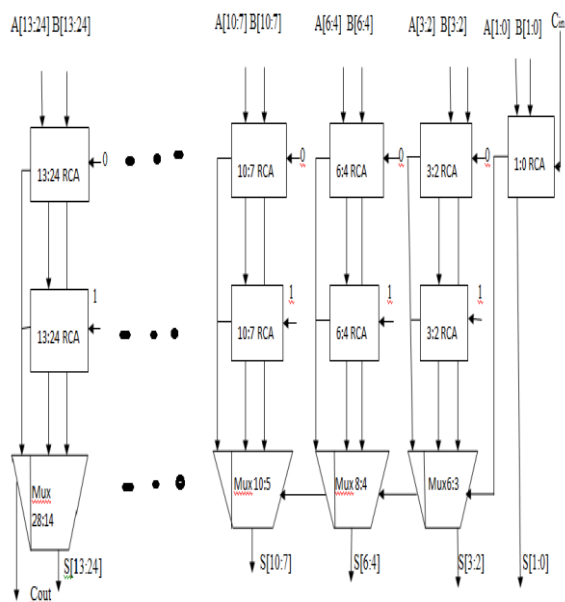


Fig. 1.2 Structure of 128 Bit Conservative Non-linear CSLA

Fig1.2 signifies the Conservative Non-linear carry select adder (CSLA) structure. It explains how the operation is performed in each section.

### III.NON-LINEAR BEC CSLA

The structure of Non-linear BEC carry select adder is obtained by modifying the second stage of Conservative carry select adder. The structure of second stage Non-linear BEC carry select adder is shown below. The operation of second stage is to increase the sum value by 1. It can perform this operation with less delay. The second stage block and multiplexer is placed parallel to the first stage which is present in the Conservative carry select adder. The second stage received the sum and carry out of the first stage it compute the operation and produce the sum and carry for its stage. It will ascend the input value by one. If the select line value of mux=0 the first stage sum and carry has been produced. Suppose the select line value=1 the second stage sum and carry will be taken as a final stage output.

$$Z_0 = \sim C_0 \quad (3)$$

$$Z_1 = C_0 \oplus C_1 \quad (4)$$

$$Z_2 = C_2 \oplus (C_0 \& C_1) \quad (5)$$

$$Z_3 = C_3 \oplus (C_0 \& C_1 \& C_2) \quad (6)$$

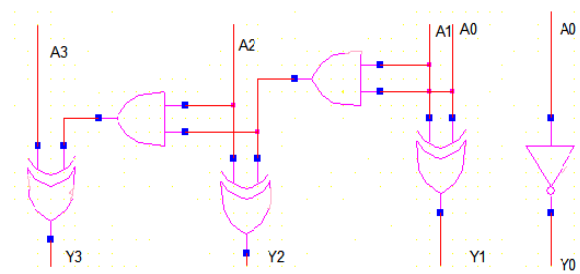


Fig.2.1 Structure of Binary to Excess-1 Convertor

Fig.2.1 epitomizes the common BEC structure in which the second stage operation has been carry out. The mathematical expressions symbolize the operation of a BEC structure. Depending upon the logical operations of a mathematical expressions the final sum and carry out were engender. The parameters C0,C1,C2 indicate the sum of first stage while C3 characterize the carry out of the first stage.

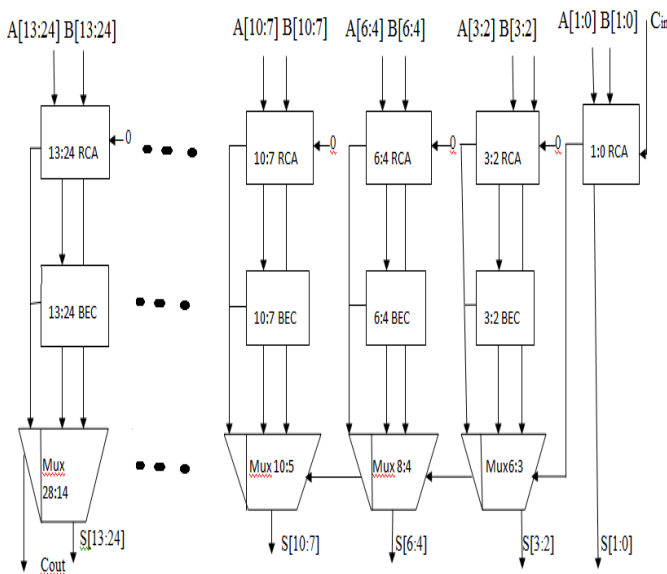


Fig 2.2 Structure of 128 Bit BEC based Non-linear CSLA

The parameters Z0, Z1, Z2 represent the sum of a BEC unit while Z3 represent the carry out of a BEC unit. Fig 2.2 show the construction of non linear BEC carry select adder.

#### IV. PROPOSED NON-LINEAR CSLA

The goal of proposed technique is to improve the performance of adder. By reducing the utilization of large number of gates the area get reduce. In existing techniques more number of XOR gates were used. An XOR gate is a combination of 2 AND, 2 NOT and 1 OR gates. So it can occupy large number of space. Due to this the time taken to compute the operation will also high. To eliminate this problem instead of XOR gate combination of AND, OR gates are used which can perform the same operation as that of XOR gate. By performing the XOR operation between A and B the partial sum and carry is generated. The carry C1 and C2 are produced by using the partial sum and carry. According to the input carry value one of the carry is chosen as a final carry. The carry out has been obtained from the most significant bit of C. The XOR operation play a major role to produce MSB bits of final-sum (S<sub>f</sub>). It is perform in between the most significant bit of partial sum and least significant bit of carry. The Xor operation is made between input carry and remaining bit of partial sum to obtain the remaining of final sum (S<sub>f</sub>).

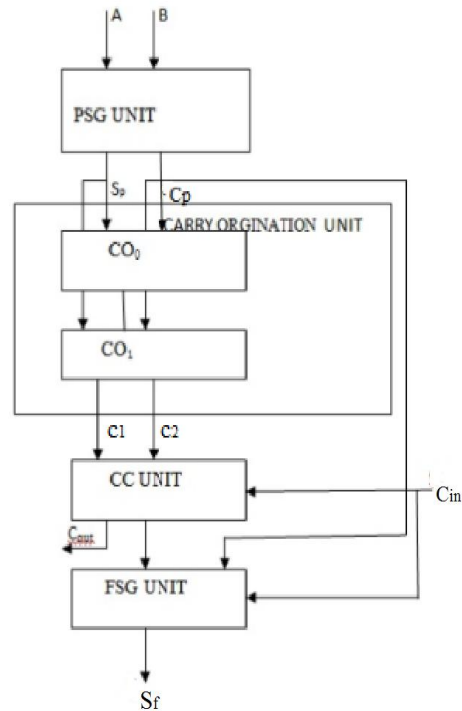


Fig 3.1 Block diagram of Non-linear Modified CSLA

$$S_p(i) = A(i) \oplus B(i) \tag{7}$$

$$C_p(i) = A(i) \cdot B(i) \tag{8}$$

$$CO_0(i) = C_1(i-1) \cdot S_p(i-1) + C_p(i) \text{ where } C_1(0) = 0 \tag{9}$$

$$CO_1(i) = C_2(i-1) \cdot S_p(i-1) + C_p(i) \text{ where } C_2(0) = 0 \tag{10}$$

$$C_1(i) = CO_0(i) \text{ when } (C_{in} = 0) \tag{11}$$

$$C_2(i) = CO_1(i) \text{ when } (C_{in} = 1) \tag{12}$$

$$C_{out} = C(n-1) \tag{13}$$

$$S_f(0) = S_p(0) \oplus C_{in} \tag{14}$$

$$S_f(i) = S_p(n-1) \oplus C(i-1) \tag{15}$$

The mathematical expressions provide information about the operation performed in each unit. It clearly explains how the final sum and carry out is generated.

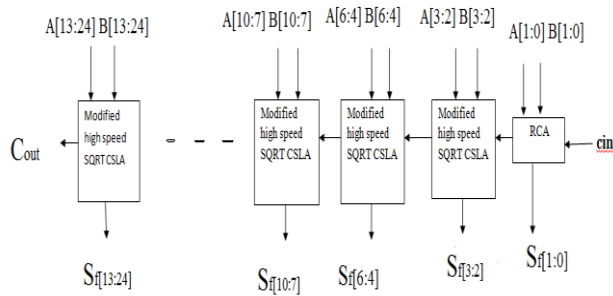


Fig 3.2 Structure of Modified Non-linear CSCLA

Fig. 3.2 portrays how the operation is performed in Modified Non-linear CSCLA. The significant of this technique is prior carry selection.

**V. TEST RESULTS**

The Verilog HDL language is used to implement the proposed work in QUARTUS II 9.1 software. The subsequent result show the performance variation between the existing and proposed work through its area and delay.

Flow Summary	
Flow Status	Successful - Thu Oct 30 20:06:05 2014
Quartus II Version	9.1 Build 222 10/21/2009 SJ Web Edition
Revision Name	sqrt_cscla
Top-level Entity Name	s_conv_cscla128
Family	Stratix III
Met timing requirements	Yes
Logic utilization	1 %
Combinational ALUTs	387 / 38,000 ( 1 % )
Memory ALUTs	0 / 19,000 ( 0 % )
Dedicated logic registers	0 / 38,000 ( 0 % )
Total registers	0
Total pins	386 / 488 ( 79 % )
Total virtual pins	0
Total block memory bits	0 / 1,880,064 ( 0 % )
DSP block 18-bit elements	0 / 216 ( 0 % )
Total PLLs	0 / 4 ( 0 % )
Total DLLs	0 / 4 ( 0 % )
Device	EP3SL50F780C2

Fig.4.1 Logical elements utilization of Non-linear Conservative CSCLA

Fig 4.1 demonstrates the utilization of number of gates. Out of 38000 logical elements it acquired only 387 elements.

Timing Analyzer Summary									
Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths	
1 Worst-case tpd	N/A	None	63.309 ns	cin	s[125]	--	--	0	
2 Total number of failed paths								0	

Fig 4.2 Delay of Non-linear Conservative CSCLA

Fig 4.2 reveal about the excess time taken to compute a operation.

Flow Summary	
Flow Status	Successful - Wed Sep 17 15:56:27 2014
Quartus II Version	9.1 Build 222 10/21/2009 SJ Web Edition
Revision Name	Conventional_Carry_Select_Adder
Top-level Entity Name	bec_cscla128
Family	Stratix III
Met timing requirements	Yes
Logic utilization	1 %
Combinational ALUTs	364 / 38,000 ( < 1 % )
Memory ALUTs	0 / 19,000 ( 0 % )
Dedicated logic registers	0 / 38,000 ( 0 % )
Total registers	0
Total pins	386 / 488 ( 79 % )
Total virtual pins	0
Total block memory bits	0 / 1,880,064 ( 0 % )
DSP block 18-bit elements	0 / 216 ( 0 % )
Total PLLs	0 / 4 ( 0 % )
Total DLLs	0 / 4 ( 0 % )
Device	EP3SL50F780C2
Timing Models	Final

Fig 4.3 Logical elements utilization of Non-linear BEC CSCLA

Fig 4.3 convey about the number of gates consumed. Out of 38000 logical elements it acquired only 364 logical gates.

Timing Analyzer Summary									
Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths	
1 Worst-case tpd	N/A	None	56.764 ns	a[0]	s[126]	--	--	0	
2 Total number of failed paths								0	

Fig. 4.4 Delay of Non-linear BEC CSCLA

4.4 articulate about the extra time taken to complete a operation.

Flow Summary	
Flow Status	Successful - Thu Oct 30 20:12:06 2014
Quartus II Version	9.1 Build 222 10/21/2009 SJ Web Edition
Revision Name	sqft_csla
Top-level Entity Name	s_pro_csla128
Family	Stratix III
Met timing requirements	Yes
Logic utilization	< 1 %
Combinational ALUTs	308 / 38,000 (< 1 %)
Memory ALUTs	0 / 19,000 (0 %)
Dedicated logic registers	0 / 38,000 (0 %)
Total registers	0
Total pins	386 / 488 (79 %)
Total virtual pins	0
Total block memory bits	0 / 1,880,064 (0 %)
DSP block 18-bit elements	0 / 216 (0 %)
Total PLLs	0 / 4 (0 %)
Total DLLs	0 / 4 (0 %)
Device	EP3SL50F780C2

Fig 4.5 Logical elements utilization of Modified Non- linear CSLA

Fig 4.5 convey the number of gates consumed. Out of 38000 logical elements it acquired only 308 gates.

## VI.CONCLUSION

The implementation of proposed work was done with the help of QUARTUS 9.1. The proper usage of minimum number of gates lead to the reduction of area. The delay is reduced due to the former carry selection. In the proposed work the Modified Non-linear CSLA method indicate the parallel execution so the delay has been equalized in every stage. When compared to the existing methods the carry choosing and generation unit is dissimilar from the proposed work. The Non-linear BEC CSLA consume less delay but high area when match up to the Non-linear conservative CSLA. The experimental result and comparison table show that the proposed method is efficient when compared to the existing work.

Timing Analyzer Summary									
Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths	
1 Worst-case tpd	N/A	None	24.024 ns	a[1]	s[60]	--	--	0	
2 Total number of failed paths								0	

Fig 4.6 Delay of Modified Non-linear CSLA

Fig 4.6 describes about the extreme time taken to fulfill a operation.

**TABLE.1 COMPARISON BETWEEN EXISTING AND PROPOSED METHODS.**

DESIGN	WIDTH	AREA	DELAY(ns)
CONSERVATIVE NON-LINEAR CSLA	128	387	63.309
BEC BASED NON-LINEAR CSLA	128	364	56.764
MODIFIED NON-LINEAR CSLA	128	308	24.024

The Table.1 shown that the difference between Area and Delay of Proposed method and Existing Methods.