AN AREA EFFICIENT ADDER DESIGN FOR VLSI CIRCUITS

Lekshmi Devi V.S*

Abstract— In VLSI world, the design of area and power efficient high speed logic data paths has always been a hot topic of research. Carry Select Adder (CSLA) is one of the fastest adders used in many processors for performing fast arithmetic functions. The objective of the project is to develop a synthesizable CSLA model by making use of use of a simple and efficient gate-level modification. Modified design was compared for area, power and speed with the existing regular CSLA architecture. The performance evaluation and subsequent comparison have been done using VHDL programming and the synthesizing tool used was Xilinx 13.1v.

Keywords— **CSLA,Xilinx,BEC**

* Electronics Department, Noorul Islam University, Kumarakovil , Kanyakumari District , Tamilnadu , India

I. INTRODUCTION

Design of area and power efficient high-speed logic systems have always been an important area of research in VLSI system design.While performance and area remained to be the two major design tolls, power consumption had become a critical concern in today's VLSI system design. The need for low-power VLSI system arises from two main forces. First, with the increase in operating frequency and processing capacity per chip, large currents have to be delivered thus increasing the power consumption, which in turn increases the heat dissipation. Second, battery life in portable electronic devices was limited. Low power design directly leads to prolonged, safe operation time in these portable devices.

Addition is the most fundamental arithmetic operation and has ranked as the most extensively used operation whose application ranges from application-specific digital signalling processor to general purpose processors. A system's performance is generally determined by the performance of the adder because the adder is generally the slowest element in the system. Furthermore, it is generally the most area consuming. Therefore low-power area efficient adder design has been an important part in low- power VLSI system design.

Among the myriad of aggressive techniques, carry select adder (CSLA) had been an eminent technique in the space-time tug-of-war of integrated circuit design. Conventionally, CSLA is being implemented with dual ripple carry adder (RCA) with the carry-in o f 0 and 1, respectively.

This paper is based on the idea of using a Binary to Excess-1 Converter (BEC) instead of a ripple carry adder (RCA) with Cin=1 in the regular CSLA to achieve lower area and power consumption. that uses lesser number of logic gates than the n-bit Full Adder (FA) structure. The regular CSLA has been chosen for comparison with the proposed design as it has a more balanced delay, and requires lower power and area.

A Carry Select Adder has been simulated and the simulation results have been compared with the existing design. The rest of the paper is organized as follows: Section II deals with evaluation of BEC logic. Section III reports the evaluation of existing design; Section IV details the proposed system analysis in detail and section V includes the simulation results.

II. BINARY TO EXCESS -1 CONVETER

The basic idea is to use Binary to Excess-1 Converter (BEC) instead of RCA with Cin=1 in the regular CSLA to achieve lower area and power consumption. To replace the n-bit RCA, an n+1-bit BEC is required.

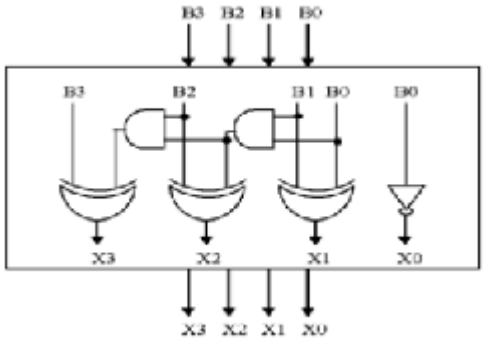


Fig 1 Schematic of a four bit BEC

The Boolean expressions of the 4-bit BEC is listed as

X0 = ~B0

X1 = B0 ^ B1

X2 = B2 ^ (B0 & B1)

X3 = B3 ^ (B0 & B1 & B2)

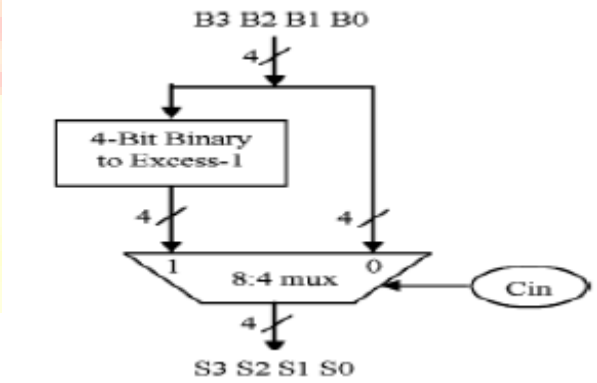


Fig 2 Schematic of a 4 bit BEC with 8:4 mux

Figure 2 illustrates how the basic function of the CSLA is achieved 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 C_{in}.

III. DELAY AND AREA EVALUATION OF EXISTING CSLA ARCHITECTURE

The delay and area evaluation methodology considers all gates to be made up of AND, OR, and NOT (AOI), each having a delay equal to 1 unit and an area equal to 1 unit. The method then adds up the number of gates in the longest path of a logic block that contributes to the maximum delay. The area evaluation has been done by counting the total number of AOI gates required for each logic block.

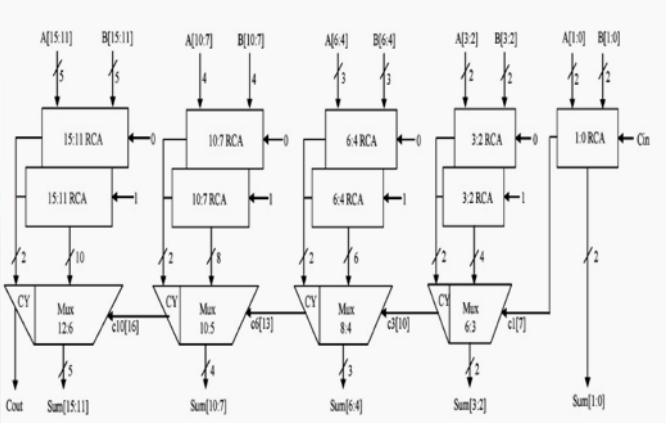


Fig 3 Block diagram of a regular CSLA

The 16-bit carry-select adder of figure 3 is divided into sectors of lengths 2, 3, 4, and 5, proceeding from least-significant to most-significant bit. The 4-bit sector of Figure 4 illustrates the general underlying principle. Within the sector, there are two 4-bit ripple-carry adders receiving the same data inputs but different carry-ins. The upper adder has a carry-in of zero; the lower adder a carry-in of one. The actual carry in from the preceding sector selects one of the two adders. If the carry-in is zero, the sum and carry-out of the upper adder are selected. If the carry-in is one, the sum and carry-out of the lower adder are selected. Logically, the result is no different than if a single ripple-carry adder were used. The difference, of course, is in performance. Instead of having to ripple through four full adders, the carry now only has to pass through a single multiplexer.

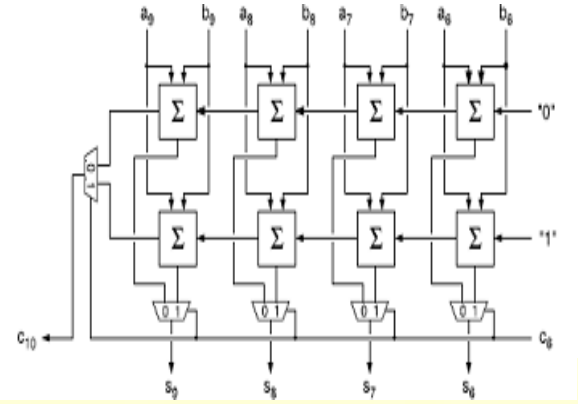


Fig 4 Four bit sector schematic

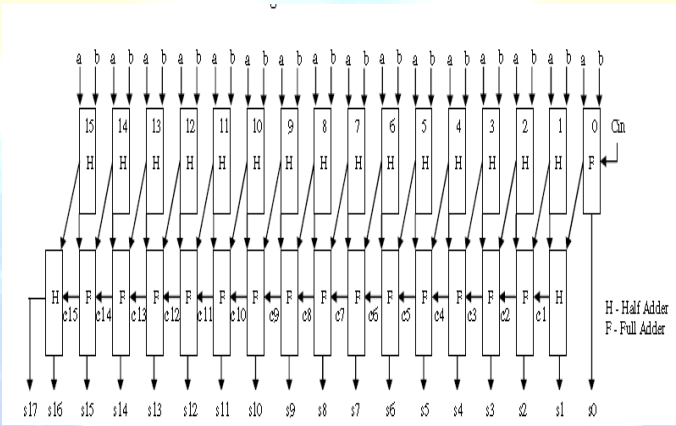


Fig 5 Detailed internal schematic of a regular 16-bit CSLA

The detailed internal architecture of a 16-bit conventional CSA is shown in Figure 5. It has 17-half adders and 15-full adders. Since the ripple carry adder (RCA) is used in the final stage, this structure yields large carry propagation delay. To reduce this delay, the final stage of CSA is divided into 5 groups .The first group includes $1+ \log_2 n$ bit value and other groups include $\log_2 n$ bit value, where n is the bit size of the adder. The divided groups are listed below.

i). c4, s [4:0]

ii). c7, x [7:5]

iii). c10, x [10:8]

iv). c13, x [13:11]

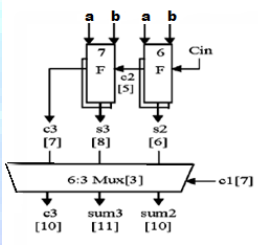
v). x [17:14]

The first group of output s[4:0] are directly assigned as the final output; the second group c7,x[7:5] manipulates the partial result by considering c4 is zero; the third group c10,x[10:8]

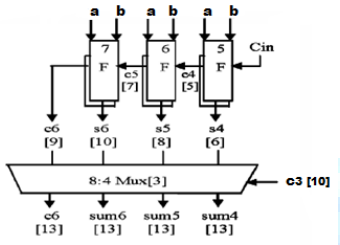
manipulates the partial result by considering c7 is zero; the fourth group c13,x[13:11] manipulates the partial result by considering c10 is zero and the fifth group x[17:14] manipulates the partial result by considering c13 is zero.Based on this approach, the delay and area evaluation of the CSLA adder block consisting of 2:1 mux, Half Adder (HA), and Full Adder (FA) are evaluated and listed in Table 1

Adder blocks	Delay	Area
XOR	3	5
2:1 Mux	3	4
Half adder	3	6
Full adder	6	13

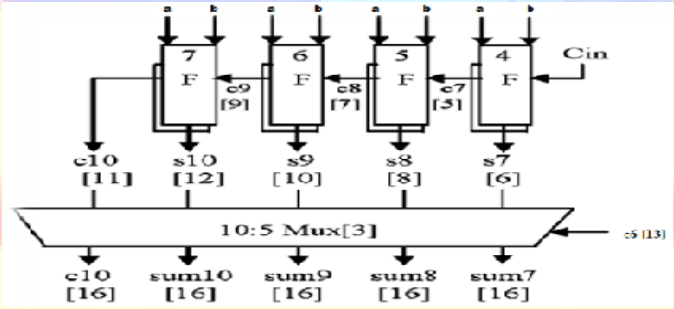
Table 1 Delay and area evaluation results of various gates in the adder block



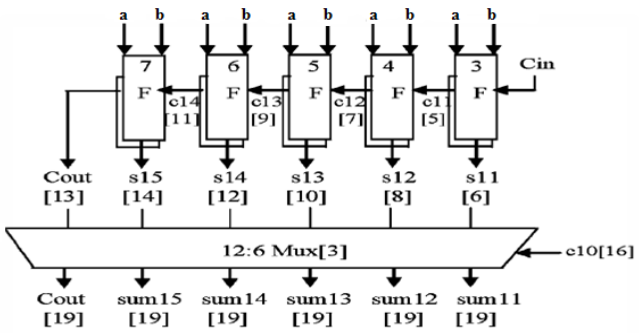
group 1



group 2



group 3



group 4

Fig 5 Delay and Area Evaluation of different stages of regular CSLA

The group2 has two sets of 2 bit RCA. The arrival time of selection input c1 [time (t) = 7] of 6:3 mux is earlier than s3 [t = 8] and later than s2 [t = 6]. Thus, sum3 [t = 11] is the summation of s3 and mux [t= 3].Similarly, sum2 [t = 10] is the summation of c1 and mux.

Except for group2, the arrival time of mux selection input is always greater than the arrival time of data outputs from the RCA's.Thus,the delay of group3 togroup5 can be determined, respectively as follows:

c6, sum [6 : 4] = c3 [t = 10] + mux

c10, sum [10 : 7] = c6 [t = 13] + mux

C_{out}, sum [15: 11] = c10 [t = 16] + mux.

The one set of 2-bit RCA in group2 has 2 FA for C_{in}=1 and the other set has one FA and one HA for C_{in}=0. Based on the area count of Table I, the total number of gate counts in group2 can be determined as follows and is listed in Table II

Gate count =57 (FA +HA + Mux)

FA=39

HA=6

Mux=12

Group	Delay	Area
Group2	11	57
Group3	13	87
Group4	16	117
Group5	19	147

Table II Area and Delay evaluation results of various sections of regular CSLA

IV MODIFIED CSLA BASED ON BEC

The structure is again divided into different groups of different bit size RCA and BEC.To replace the n-bit RCA, an n+1-bit BEC is required. One input to the mux goes from the RCA with Cin=0 and other input from the BEC.

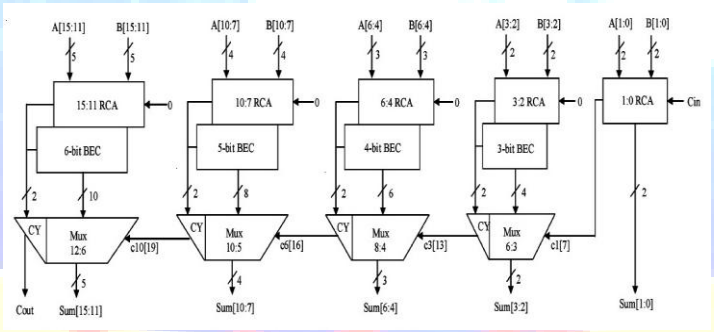


Fig 6 Block diagram of proposed CSLA

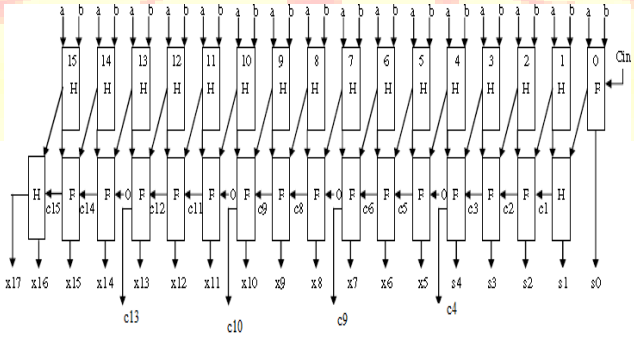


Fig 7 Detailed internal schematic of proposed CSLA

Here also, the design is spitted into 5 different stages and area and delay are evaluated as it had done previously. The delay and area estimation of each group are shown in Figures (a) through (d).The steps leading to the evaluation are.

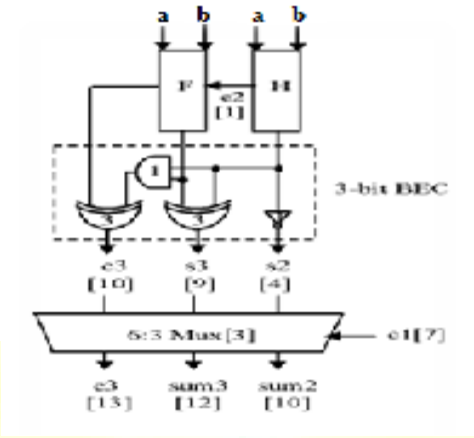
The group2 has one 2-bit RCA which consists of one FA and one HA for C_{in}=0.Instead of another 2 bit RCA with C_{in}=1, a 3 bit BEC is used which adds one to the output from 2 bit RCA. Based on the consideration of delay values of Table I, the arrival time of selection input c1 [t=7] of 6:3 mux is earlier than the s3 [t=9] and c3 [t=10] and later than the s2 [t=4]. Thus, the sum3 and final c3 (output from mux) are depending on s3 and mux and partial c3 (input to mux) and mux, respectively. The sum2 depends on c1 and mux2

For the remaining group's the arrival time of mux selection input is always greater than the arrival time of data inputs from the BEC's.Thus, the delay of the remaining groups depends on the arrival time of mux selection input and the mux delay.

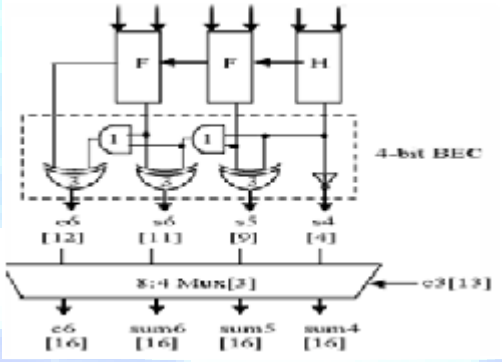
Similarly, the estimated maximum delay and area of the other groups of the modified CSLA are evaluated and listed in Table III.Comparing the tables, it is clear that the proposed modified CSLA saves 113 gate areas than the regular CSLA, with only 11 increases in gate delays.

Group	Delay	Area
Group2	13	43
Group3	16	61
Group4	19	84
Group5	22	107

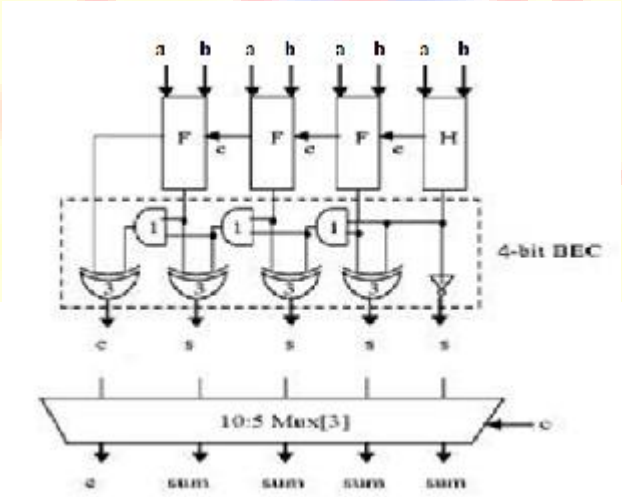
Table III Area and Delay evaluation results of various sections of Modified CSLA



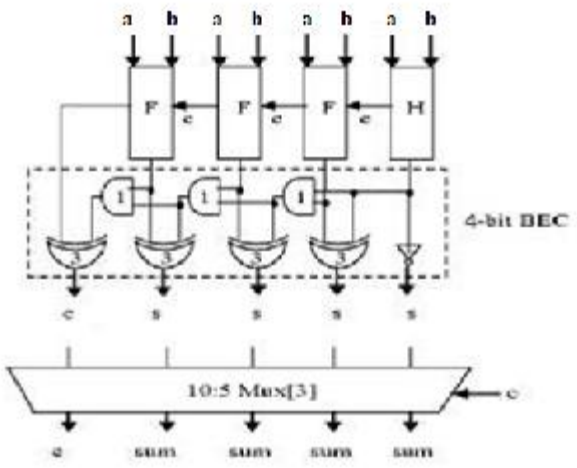
group 1



group 2



grgroup 3



group 5

Fig 8 Delay and Area Evaluation of different stages of modified CSLA

V. SIMULATION RESULTS

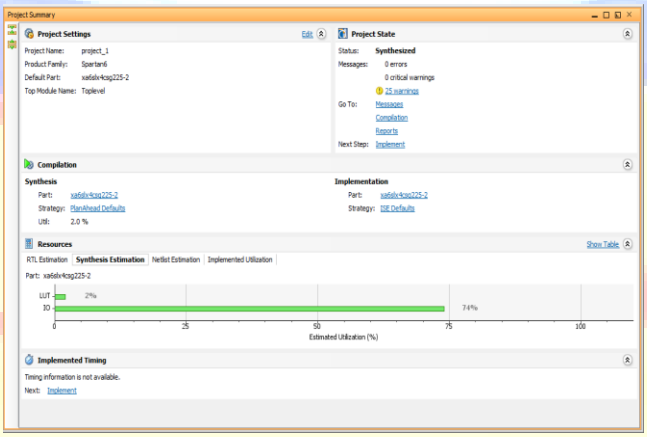


Fig 9 Resource allocation window of Existing design

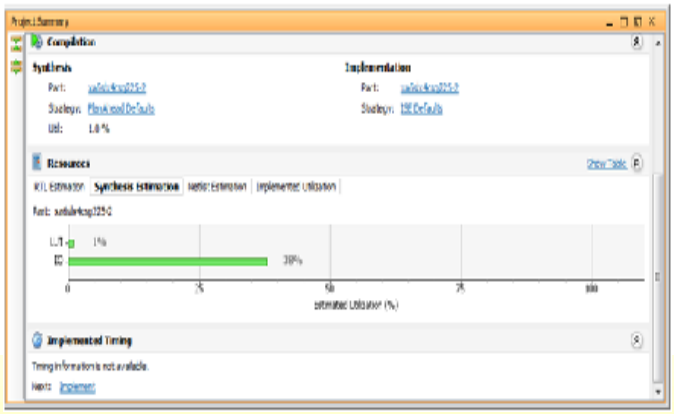


Fig 10 Resource allocation window of proposed design

A . COMPARISON OF SYNTHESIS RESULTS

ADVANCED HDL SYNTHESIS REPORT OF EXISTING SYSTEM

Number of Slice LUTs : 48 out of 2400 2%

Number used as Logic : 48 out of 2400 2%

Number with an unused LUT : 0 out of 48 0%

Number of fully used LUT-FF pairs: 0 out of 48 0%

Number of IOs : 146

Number of bonded IOBs : 98 out of 132 74%

Total number of paths / destination ports : 673 / 25

Delay : 21.914ns (Levels of Logic = 22)

Total : 21.914ns (8.253ns logic, 13.661ns route)

(37.7% logic, 62.3% route)

ADVANCED HDL SYNTHESIS REPORT OF PROPOSED SYSTEM

Number of Slice LUTs : 31 out of 2400 1%

Number used as Logic : 31 out of 2400 1%

Number of LUT Flip Flop (FF) pairs used : 31

Number with an unused Flip Flop : 31 out of 31 100%

Number of IOs : 50

Number of bonded IOBs : 50 out of 132 37%

Number of IOs : 50
Number of bonded IOB : 0 out of 132 37%
Total number of paths / destination ports : 209 / 17
Delay : 8.735ns (Levels of Logic = 8)
Total : 8.735ns (4.979ns logic, 3.756ns route)
(57.0% logic, 43.0% route)

V1 CONCLUSION

A simple approach was proposed here to improve the speed of addition. A carry select adder is used to speed up the final addition in many parallel multipliers. But due to the structure of the CSLA it occupies more chip area. As it uses multiple pairs of RCA's to generate partial sum and carry by considering C_{in}=0 and C_{in}=1, the complexity of the final adder structure was high. By replacing, as demonstrated in this paper, the RCA with C_{in}=1 with the BEC logic, obviously the maximum area and delay can be reduced in the final adder structure.

The reduced number of gates of this work offers the great advantage in the reduction of area and also the total power. Therefore, the BEC logic can be used with any type of adder to enhance the speed of addition. Area and delay evaluation of the proposed as well as the existing designs have been performed using Xilinx 13.1 and a comparison have also been recorded.

REFERENCES

1]. Padma Devi, Ashima Girdher, Balwinder Singh, "Improved Carry Select Adder with Reduced Area and Low Power Consumption", International journal of computer applications (0975 – 8887) volume 3 – no.4, June 2010.

`

[2].Youngjoon Kim and Lee-Sup Kim, " Low Power Carry Select AdderWith Reduced Area", IEEE J.Solid-State Circuits, vol. 36, no. 10, pp. 1538–1545, Oct. 2001

[3].B.Ramkumar, Harish M Kittur, P.Mahesh Kannan,"Asic Implementation Of Modified Faster Carry Save Adder", European Journal of Scientific Research Wassn 1450-216x Vol.42 No.1 (2010).

[4].I-C*hin Wey*,C*heng-chen ho*,YI-*sheng lin*,C*hien-chang pen*,"An Area-Efficient Carry Select Adder Design By Sharing The Common Boolean Logic Term", IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. vol. 49, no. 1, pp. 16–24, Jan. 2002.

[5].N. Ravikumar1, M. Vishwanath, B. Durga Malleswara Reddy, "An Area Efficient 32-Bit Carry-Select Adder For Low Power Applications", International Journal Of Computer&Communication Technology (Ijcct), Vol.-3, Wassue - 4, 2012.

[6].B*ehnamAmelifard*,,*FarzanFallah*,*Massoud Pedram*,"Closing The Gap Between Carry Select Adder And RippleCarryAdder: