Design and Verification of Area Efficient Carry Select Adder

V D L Sivasankar[1] Ch. Swathi[2] B. Shirisha[3]
[1]M. Tech Student [2]Assistant Professor [3]Assistant Professor (HOD)
[1,2,3]Department of Electronics and Communication Engineering
[1,2,3]Vathsalya Institute of science and technology, R.R Dist, T.S, India

Abstract— Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. From the structure of the CSLA, it is clear that there is scope for reducing the area and power consumption in the CSLA. This work uses a simple and efficient gate-level modification to significantly reduce the area and power of the CSLA. Based on this modification 16, 32 square - root CSLA (SQRT CSLA) architecture have been developed and compared with the regular SQRT CSLA architecture. The proposed design has reduced area and power as compared with the regular SQRT CSLA with only a slight increase in the delay. This work evaluates the performance of the proposed designs in terms of delay, area.
Key words: RISC, Lopower, BEC, Carry select adder.

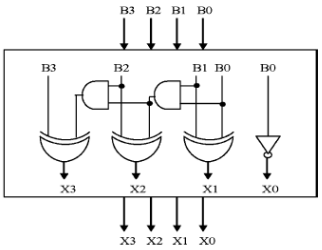
I. INTRODUCTION

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.

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 [1]. However, the CSLA is not area efficient because it uses multiple pairs 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 (mux).

The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA with in the regular CSLA to achieve lower area and power consumption [2]–[4]. The main advantage of this BEC logic comes from the lesser number of logic gates than the -bit Full Adder (FA) structure. The details of the BEC logic are discussed in Section III.

This brief is structured as follows. Section II deals with the delay and area evaluation methodology of the basic adder blocks. Section III presents the detailed structure and the function of the BEC logic.



4 Bit BEC

II. ADDER BLOCKS

A. Ripple carry adder:

It is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a C_{in}, which is the C_{out} of the previous adder. This kind of adder is a ripple carry adder, since each carry bit "ripples" to the next full adder. Note that the first (and only the first) full adder may be replaced by a half adder.

In a 32-bit [ripple carry] adder, there are 32 full adders, so the critical path (worst case) delay is 3 (from input to carry in first adder) + 31 * 2 (for carry propagation in later adders) = 65 gate delays.

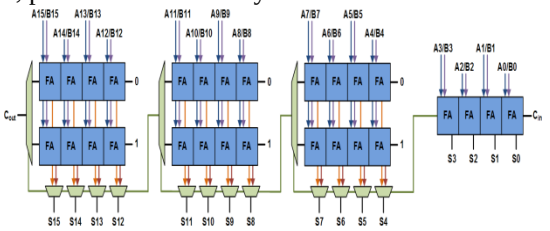
B. Carry Select Adder:

In electronics, a carry-select adder is a particular way to implement an adder, which is a logic element that computes the n+1-bit sum of two n-bit numbers. The carry-select adder is simple but rather fast, having a gate level depth of O(root(n)).

The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known.

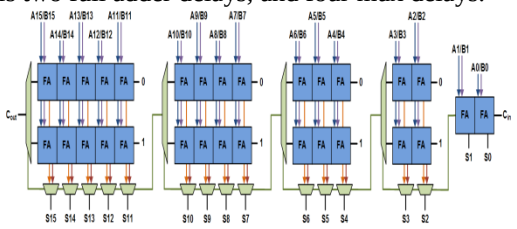
C. Uniform-sized adder:

A 16-bit carry-select adder with a uniform block size of 4 can be created with three of these blocks and a 4-bit ripple carry adder. Since carry-in is known at the beginning of computation, a carry select block is not needed for the first four bits. The delay of this adder will be four full adder delays, plus three MUX delays.


Uniform sized adder

D. Variable-sized adder:

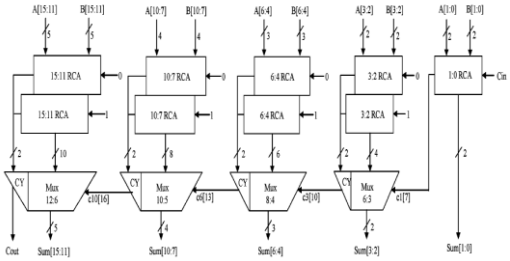
A 16-bit carry-select adder with variable size can be similarly created. Here we show an adder with block sizes of 2-2-3-4-5. This break-up is ideal when the full-adder delay is equal to the MUX delay, which is unlikely. The total delay is two full adder delays, and four mux delays.



Variable sized Adder

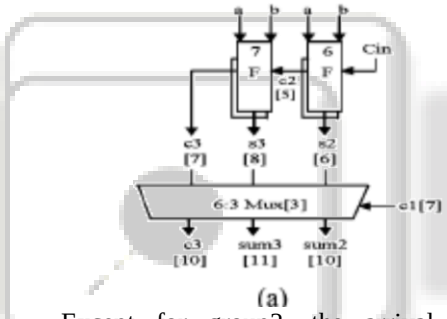
E. Regular CSA:

The structure of the 16-b regular SQRT CSLA is shown in Fig. 4. It has five groups of different size RCA. The delay and area evaluation of each group are shown in Fig. 5, in which the numerals within specify the delay values, e.g., sum2 requires 10 gate delays. The steps leading to the evaluation are as follows.



Regular CSA

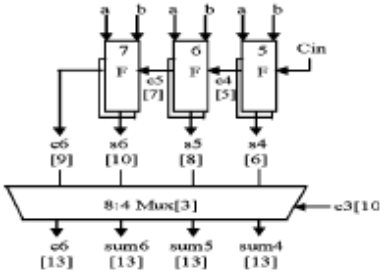
The group2 [see Fig.] has two sets of 2-b RCA. Based on the consideration of delay values of Table I, the arrival time of selection input c1(time(t)=7) of 6:3 mux is earlier than s3[t=8] and later thans2[t=6]. Thus, sum3 [t=11] is summation of s3 and mux [t=3] and sum2 [t=10] is summation of c1 and mux



(a)

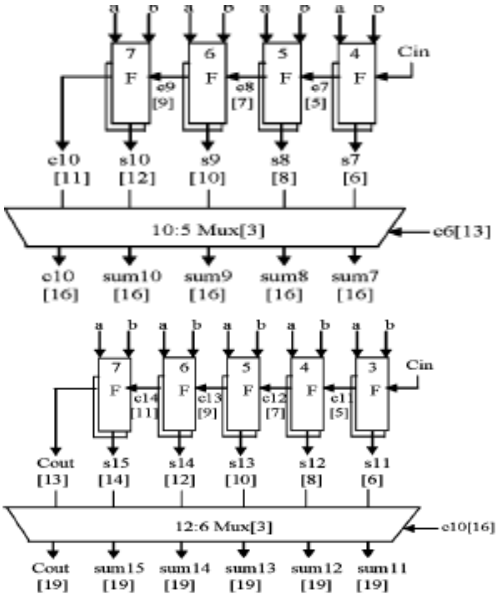
Except for group2, the arrival time of mux selection input is always greater than the arrival time of data outputs from the RCA's. Thus, the delay of group3 to group5 is determined, respectively as follows:

$$\{c6, \text{sum}[6:4]\} = c3[t=10] + \text{mux}$$
$$\{c10, \text{sum}[10:7]\} = c6[t=13] + \text{mux}$$
$$\{\text{cout}, \text{sum}[15:11]\} = c10[t=16] + \text{mux}.$$



The one set of 2-b RCA in group2 has 2 FA for cin=0 and the other set has 1 FA and 1 HA forcing=1. Based on the area count of Table I, the total number of gate counts in group2 is determined as follows:

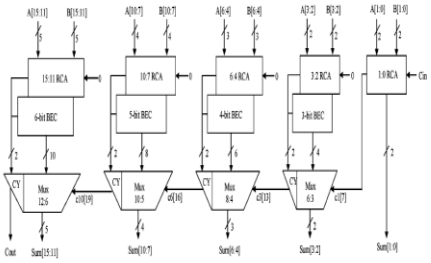
$$\text{Gate count} = 57\,(\text{FA} + \text{HA} + \text{Mux})$$
$$\text{FA} = 39(3 * 13)$$
$$\text{HA} = 6(1 * 6)$$
$$\text{Mux} = 12(3 * 4).$$



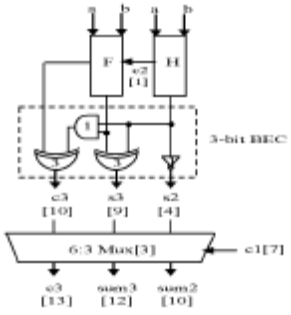
III. LOW POWER AREA EFFICIENT CARRY SELECT ADDER CIRCUIT

CSA adder, like ripple-carry adders, is the carry has to to travel through every full adder block. There is a way to improve the speed by duplicating the hardware due to the fact that the carry can only be either 0 or 1. The method is based on the conditional sum adder and extended to a carry-select adder. With one RCA.each computing the case of the one polarity of the carry-in, the sum can be obtained with a 2x1 multiplexer with the carry-in as the select signal.

The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA with in the regular CSLA to achieve lower area and power consumption. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure**.** This work is to use BEC instead of the RCA with cin=1 in order to reduce power consumption of the regular CSA. To replace the n-bit RCA, an n+1bit BEC is required [5].

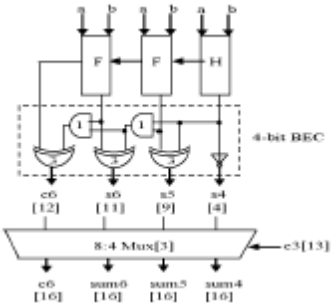


Modified CSA



Group 2 of Modified BEC

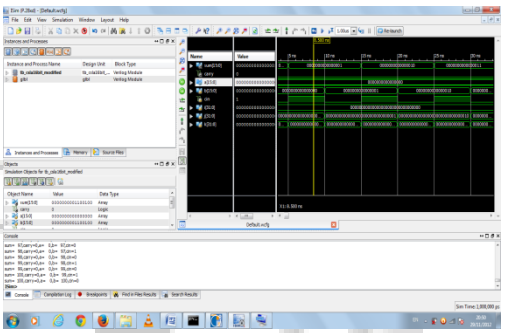
The group2 [see Fig] has one 2-b RCA which has 1 FA and 1 HA for Cin=0. Instead of another 2-b RCA with Cin=1 a 3-b BEC is used which adds one to the output from 2-b RCA



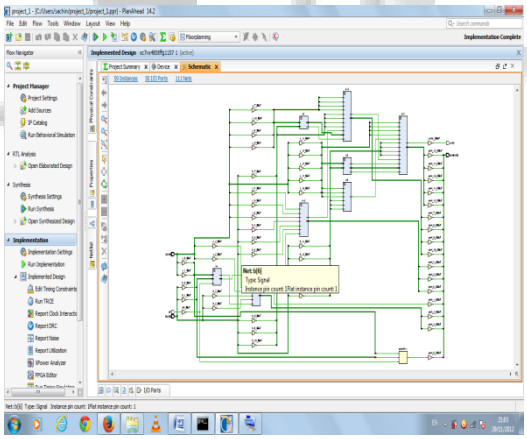
Group 4 of Modified BEC

Similarly, the estimated maximum delay and area of the other groups of the modified SQRT CSLA are evaluated.

IV. RESULT



Simulation Result



Synthesis RTL

V. CONCLUSION

Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. From the structure of the CSLA, it is clear that there is scope for reducing the area consumption in the CSLA. This work uses a simple and efficient gate-level modification to significantly reduce the area and power of the CSLA. Based on this modification 16- CSLA architecture have been developed and compared with the regular CSLA architecture. Carry Select Adder is realized by using verilog HDL. Simulation and synthesis by using Xilinx Palanahead. The delay of CSA is small high compare to Regular Carry select adder if u improve that one by no degration of area and power.

REFERENCES

[1] Verilog HDL- Digital Design and Synthesis, by Samir Palnitkar
[2] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Compute. pp. 340–344, 1962.
[3] B. Ramkumar, H.M. Kittur, and P. M. Kannan, "ASIC implementation of modified faster carry save adder," Eur. J. Sci. Res., vol. 42, no. 1, pp. 53–58, 2010
[4] T. Y. Ceiang and M. J. Hsiao, "Carry-select adder using single ripple carry adder," Electron. Lett. vol. 34, no. 22, pp. 2101–2103, Oct. 1998.
[5] J. M. Rabaey, Digital Integrated Circuits—A Design Perspective. Upper Saddle River, NJ: Prentice-Hall, 2001.
[6] Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for low power applications," in Proc. IEEE Int. Symp. Circuits Syst., 2005, vol. 4, pp. 4082–4085.