International Journal of Research in Engineering, Science and Management
Volume-2, Issue-4, April-2019
www.ijresm.com | ISSN (Online): 2581-5792

Implementation of Low Power and Area Efficient Carry Select Adder

Pankaj R. Hotkar[1], Vishal K. Kulakarni[2], Pranay Y. Kamble[3], K. S. Kazi[4]
[1,2,3]*Student, Department of E & TC, BMIT, Solapur, India*
[4]*Associate Professor, Department of E & TC, BMIT, Solapur, India*

***Abstract*: Design of low area and low power forms the largest systems in VLSI system design. Carry Select Adder (CSLA) is one of the fastest adders to perform arithmetic operations comparing with all conventional adders. From the structure of CSLA there is a scope for reducing the area and power. Based on the modification of 16 and 32-bit Carry Select Adder (CSLA) architectures have been developed and compared with the regular CSLA architecture. A Carry-select adder (CSLA) can be implemented by using Ripple carry adder. The proposed system i.e. 64-bit CSLA has reduced more power and area as compared with the regular 64-bit CSLA.**

***Keywords*: XILINX Software.**

1. Introduction

Design of area and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. 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. In the full adder circuit carry has to travel from state to state. Previous states carry need to require for the present state to do the operation. So, naturally when we increase the number of bits the propagation delay and the delay of each stage increases. Now if we don't have to depend on the transmission of the carry we can predict the carry of each stage. Now a day our computers speed is fast high in terms of GHz. So conceptually we need to improve the speed for the given design by decreasing several numbers of stage or gates. Carry Select Adder (CSLA) has a more balanced delay, and requires lower power and area [3]. The CSLA is used in many Processors to alleviate the issue of propagation delay by generating multiple carries independently and then select a carry to generate the sum [2]. As CSLA uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input c_{in} =0and c_{in} =1, then the final sum and carry are selected by the multiplexers it is considered to be area inefficient [3]. 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 consumptions [1]. However, the CSLA needs more area because of using multiples of Ripple Carry Adder for generating sum and carry on the dependency of carry input C_{in}=0 and C_{in}=1. Then the final results of sum and carry are selected by the multiplexers from bit to bit going to increase. Finally, reliable results at the output will depend upon the number of stages [3].

A. BEC

As stated above the main idea of this work is to use BEC instead of the RCA with C = 1 in order to reduce the area and power consumption of the regular CSLA. To replace the n-bit RCA, an n+1 bit BEC is required. A structure of a 4-bit BEC is shown in Fig. 1.

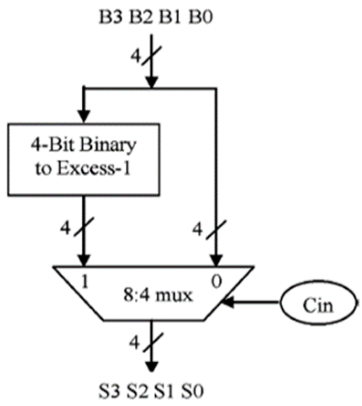

Fig.1. 4-bit BEC

The basic 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 [1]. 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}. The importance of the BEC logic stems from the large silicon area reduction when the CSLA with large number of bits are designed [7]. The Boolean expressions as follows.

X0 = B0
X1 = B0⊕B1
X2 = B2⊕ (B0 B1)
X3 = B3⊕ (B0 B1 B2)

2. Problem statement

Carry Select Adder is one of the fast adder used in many data path applications. The proposed work is to reduce power consumption and area of CSLA. To reduce the power consumption of data path we need to reduce Area of the adder. The proposed design will be implemented using ripple carry adders and BEC (binary to excess-1 converter).

3. Methodology

A. *Architecture of regular 16-bit CSLA*

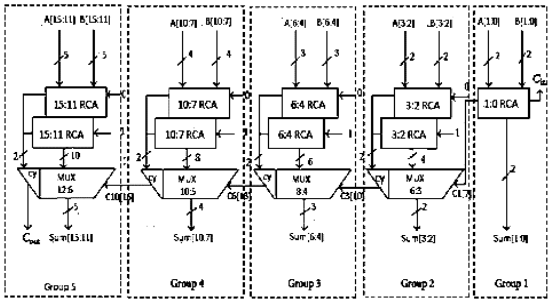

Fig. 2. Architecture of Regular 16-bit SQRT CSLA

A 16-bit carry select adder can be developed in two different sizes namely uniform block size and variable block size. Similarly, a 32, 64 and 128-bit can also be developed in two modes of different block sizes. Ripple-carry adders are the simplest and most compact full adders, but their performance is limited by a carry that must propagate from the least-significant bit to the most-significant bit. The various 16, 32, 64 and 128-bit CSLA can also be developed by using ripple carry adders. The speed of a carry-select adder can be improved up to 40% to 90%, by performing the additions in parallel, and reducing the maximum carry delay. Fig shows the Regular structure of 16-bit SQRT CSLA. It includes many ripple carry adders of variable sizes which are divided into groups. Group 1 contains 2-bit RCA which contains only one ripple carry adder which adds the input bits and the input carry and results to sum [1:0] and the carry out. The carry out of the Group 1 which acts as the selection input to mux which is in Group 1, selects the result from the corresponding RCA (Cin=0) or RCA (Cin=1). Similarly, the remaining groups will be selected depending on the Count from the previous groups.

B. *Architecture of modified 16-bit CSLA*

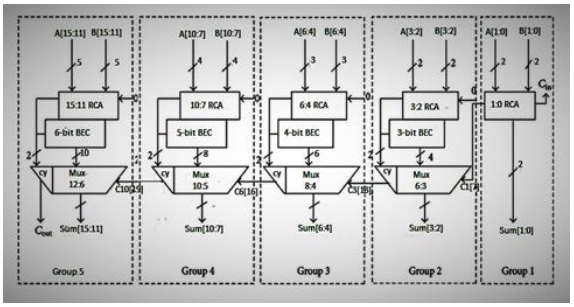

Fig. 3. Architecture of Modified 16-bit SQRT CSLA

This architecture is similar to regular 16-bit SQRT CSLA, the only change is that, we replace RCA with C_{in}=1 among the two available RCAs in a group with a BEC. This BEC has a feature that it can perform the similar operation as that of the replaced RCA with C_{in}=1. Fig. 4. shows the Modified block diagram of 16-bit SQRT CSLA. The number of bits required for BEC logic is 1 bit more than the RCA bits. The modified block diagram is also divided into various groups of variable sizes of bits with each group having the ripple carry adders, BEC and corresponding mux. As shown in the Fig.4, Group 0 contain one RCA only which is having input of lower significant bit and carry in bit and produces result of sum [1:0] and carry out which is acting as mux selection line for the next group, similarly the procedure continues for higher groups but they include BEC logic instead of RCA with C_{in}=1.Based on the consideration of delay values, the arrival time of selection input C1 of 8:3 mux is earlier than the sum of RCA and BEC. For remaining groups, the selection input arrival is later than the RCA and BEC. Thus, the sum1 and c1 (output from mux) are depending on mux and results computed by RCA and BEC respectively. The sum2 depends on c1 and mux. For the remaining parts 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 MUX depends on the arrival time of mux selection input and the mux delay. In this Modified CSLA architecture, the implementation code for Full Adder and Multiplexers of 6:3, 8:4, and 10:5 up to 12:6 were designed. The design code for the BEC was designed by using NOT, XOR and AND gates. Then 2, 3, 4, 5 up to 11-bit ripple carry adder was designed.

4. Conclusion

This seminar is aimed to reduce the area and power of CSLA architecture. The reduced number of gates of this technique offers the great advantage in the reduction of area and also the total power. We had studies that, due to this technique, the modified CSLA architecture is therefore, low area, low power and efficient for VLSI hardware implementation.

References

[1] B. Ramkumar and H. M. Kittur, "Low-Power and Area-Efficient Carry Select Adder," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 2, pp. 371-375, Feb. 2012.

[2] Sudhanshu Shekhar, Pandey, Amit Bakshi, "128 Bit Low Power and Area Efficient Carry Select Adder," in International Journal of Computer Applications, Volume 69, No. 6, May 2013.

[3] Pallavi Saxena, Urvashi Purohit, Priyanka Joshi, "Analysis of Low Power, Area- Efficient and High Speed Fast Adder," 2013.

[4] M. Chithra, G. Omkareswari, "128-bit carry select adder having less area and delay," in International journal of advanced research in electrical, electronics and instrumentation engineering vol. 2, issue 7, July 2013.

[5] Veena V. Nair, "Modified Low-Power and Area-Efficient Carry Select Adder using D-Latch," in International Journal of Engineering Science and Innovative Technology, Volume 2, Issue 4, July 2013.

[6] G. Prasad, V. S. P. Nayak, S. Sachin, K. L. Kumar and S. Saikumar, "Area and power efficient carry-select adder," *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, 2016, pp. 1897-1901.

[7] Yamini Devi Ykuntam. V. Nageswara Rao G. R. Locharla, "Design of 32-bit Carry Select Adder with Reduced Area," in International Journal of Computer Applications, vol. 75, no. 2, pp. 47-51, August 2013.