128 BIT CARRY SELECT ADDER USING BINARY TO EXCESS-ONE CONVERTER FOR DELAY REDUCTION AND AREA EFFICIENCY

1 Mrs.K.K. Varalaxmi, M.Tech, Assoc. Professor, ECE Department, 1varuhello@Gmail.Com
2 Shaik Shamshad Begum
PG Scholor. VLSI System Design
2sshamshadb@gmail.com
1, 2 AVR&SVR College of Engineering & Technology, nadyal, A.P.

Abstract:
In many data-processing processors Carry Select Adder (CSLA) is one a fastest adders used to perform arithmetic functions. The upcoming technologies depicts that there is a scope for reducing the area and power consumption in the CSLA. This work uses a simple gate level modification to significantly reduce the area and power of the CSLA. Based on this modification CSLA architecture have been developed and can be compared with the regular CSLA architecture. The proposed design has reduced area and power as compared with the regular CSLA with only a slight increase in the delay. This work evaluates the performance of the proposed designs in terms of delay, area, power, and their products by hand with logical effort and through custom design and layout in 0.18-m CMOS process technology. The results analysis shows that the proposed CSLA structure is better than the regular CSLA.

Keywords- Delay; Area; Array Multiplier, low power, VHDL Modeling & Simulation.

I. Introduction

Area and power reduction in data path logic systems are the main area of research in VLSI system design. High-speed addition and multiplication has always been a fundamental requirement of high performance processors and systems. 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 major speed limitation in any adder is in the production of carries and many authors have considered the addition problem. The CSLA is used in many computational systems to moderate 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 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). To overcome above problem, the basic idea of the proposed work is by using n-bit binary to excess-1 code converters (BEC) to improve the speed of addition. This logic can be
implemented with any type of adder to further improve the speed. Using Binary to Excess -1 Converter (BEC) instead of RCA 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 Full Adder (FA) structure.

II. Basic Adder Block

On the basis of requirements such as area, delay and power consumption some of the complex adders are Ripple Carry Adder, Carry look-Ahead Adder and Carry Select Adder. Ripple Carry Adder (RCA) shows the compact design but their computation time is longer. Time critical applications make use of Carry Look-Ahead Adder (CLA) to derive fast results but it leads to increase in area. But the carry select adder provides a compromise between the small areas but longer delay of RCA and large area with small delay of Carry Look Ahead adder.

Ripple Carry Adder consists of cascaded “N” single bit full adders. Output carry of previous adder becomes the input carry of next full adder. Therefore, the carry of this adder traverses longest path called worst case delay path through N stages. Fig. 1 shows the block diagram of ripple carry adder. Now as the value of N increases, delay of adder will also increase in a linear way. Therefore, RCA has the lowest speed amongst all the adders because of large propagation delay but it occupies the least area. Now CSLA provides a way to get around this linear dependency is to anticipate all possible values of input carry i.e. 0 and 1 and evaluate the result in advance. Once the original value of carry is known, result can be selected using the multiplexer stage. Therefore the conventional CSLA makes use of Dual RCA’s to generate the partial sum and carry by considering input carry Cin=0 and Cin=1, then the final sum and carry are selected by multiplexers.

III. BEC

The basic idea of this work is to use Binary to Excess-1 converter (BEC) instead of RCA with Cin=1 in conventional CSLA in order to reduce the area and power. BEC uses less number of logic gates than N-bit full adder structure. To replace N-bit RCA, an N+1 bit BEC is required. Therefore, Modified CSLA has low power and less area than conventional CSLA. SQRT CSLA has been chosen for comparison with modified design using BEC as it has more balanced delay, less area and low power. Regular SQRT CSLA also uses dual RCAs. In order to reduce the delay, area and power, the design is modified by using BEC instead of RCA with Cin=1. Therefore, the modified SQRT CSLA occupies less area, delay and low power. Further also, the parameters like delay, area and power can be reduced.
Fig. 3. 4-b BEC with 8:4 mux.

TABLE I
TRUTH TABLE OF 4-BIT BINARY TO EXCESS-1 CONVERTER

<table>
<thead>
<tr>
<th>Binary Logic $B_0 B_1 B_2 B_3$</th>
<th>Excess-1 Logic $X_0 X_1 X_2 X_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0001</td>
</tr>
<tr>
<td>0001</td>
<td>0010</td>
</tr>
<tr>
<td>0010</td>
<td>0011</td>
</tr>
<tr>
<td>0011</td>
<td>0100</td>
</tr>
<tr>
<td>0100</td>
<td>0101</td>
</tr>
<tr>
<td>0101</td>
<td>0110</td>
</tr>
<tr>
<td>0110</td>
<td>0111</td>
</tr>
<tr>
<td>0111</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>1001</td>
</tr>
<tr>
<td>1001</td>
<td>1010</td>
</tr>
<tr>
<td>1010</td>
<td>1011</td>
</tr>
<tr>
<td>1011</td>
<td>1100</td>
</tr>
<tr>
<td>1100</td>
<td>1101</td>
</tr>
<tr>
<td>1101</td>
<td>1110</td>
</tr>
<tr>
<td>1110</td>
<td>1111</td>
</tr>
<tr>
<td>1111</td>
<td>0000</td>
</tr>
</tbody>
</table>

Fig. 4 shows the 16-bit Conventional CSLA. The conventional CSLA is area consuming due to the use of dual RCA's.

Modified SQRT CSLA is similar to that of regular SQRT CSLA, the only difference is we replace RCA with Cin=1 with BEC. This replaced BEC performs the same operation as that of the replaced RCA with Cin=1. Fig. 5 shows the block diagram of modified SQRT CSLA. This structure consumes less area; delay and power than regular SQRT CSLA because of less number of transistors are used.

Fig. 4 16-bit conventional carry select adder

IV. Architecture of 128-bit CSLA

A 16-bit carry select adder can be developed in two different sizes namely uniform block size and variable block size.

The number of bits required for BEC logic is 1 bit
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 importance of the BEC logic stems from the large silicon area reduction when the CSLA with large number of bits are designed. The Boolean expressions of the 8-bit BEC is listed improved up to 40% to 90%, by performing the additions in parallel, and reducing the maximum carry delay.

The structure of the 128-bit regular SQRT CSLA is shown in Fig. 6. It has five groups of different size RCA. The delay and area evaluation of each group are shown in Fig. 3, in which the numerals with in specify the delay values, e.g., sum2 requires 10 gate delays.

V. Modified Architecture of 128-bit CSLA

This architecture is similar to regular 64-bit SQRT CSLA, the only change is that, we replace RCA with Cin=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 Cin=1. Fig 7 shows the Modified block diagram of 64-bit SQRT CSLA.

VI Comparison of Regular and Modified 128-bit CSLA

Percentage of delay overhead exhibits a similar decreasing trend with bit size. The delay overhead for the 8, 16, and 32-b is 14%, 9.8%, and 5.63% respectively, whereas for the 64-b it reduces to only 4.75%. The power–delay product of the proposed 8-b is higher than that of the regular CSLA by 5.2% and the area–delay product is lower by 2.9%. However, the power-delay product of the proposed 16-b CSLA reduces by 1.76% and for the 32-b and 64-b by as much as 8.18%, and 12.28% respectively. Similarly the area-delay product of the proposed design for 16-, 32-, 64-b and 128-b is also reduced by 6.7%, 11%, and 14.4% respectively.

Table 2: Comparison values
VI. Result

The implemented design in this work has been simulated using Verilog-HDL (Modelsim). The adders (of various size 16, 32, 64 and 128) are designed and simulated using Modelsim. After simulation the different size codes are synthesized using Xilinx ISE 10.1. The simulated files are imported into the synthesized tool and corresponding values of delay and area are noted. The synthesized reports contain area and delay values for different sized adders. The similar design flow is followed for both the regular and modified CSLA of different sizes.

VII. Conclusion

Power, delay and area are the constituent factors in VLSI design that limits the performance of any circuit. This work presents a simple approach to reduce the area, delay and power of CSLA architecture. The conventional carry select adder has the disadvantage of more power consumption and occupying more chip area. The proposed SQRT CSLA using common Boolean logic has low power, less delay and reduced area than all the other adder structures. It is also little bit faster than all the other adders. In this way, the transistor count of proposed SQRT CSLA is reduced having less area and low power which makes it simple and efficient for VLSI hardware implementations.

References