AN OPTIMIZED IMPLEMENTATION OF 64-BIT CARRY SELECT ADDER IN FPGA TECHNOLOGY

V.BHARATHI  
honey.bharathi76@gmail.com

1PG Scholar,Dept of ECE, Sreenivasa Institute of Technology and Management Studies (Autonomous) Murkambattu, Chittoor, JNTU Anapatap University

D.Y. PUSHPA MITRA

yoursmitra.2008@gmail.com

2Associate Professor,Dept of ECE, Sreenivasa Institute of Technology and Management Studies (Autonomous) Murkambattu, Chittoor JNTU Anapatap University

Abstract: In several data-processing processors Carry Select Adder (CSLA) is one quickest adders used to perform arithmetic functions. The approaching technologies depicts that there is a scope for reducing the space and power consumption in the CSLA. This work uses a straightforward gate level modification to considerably cut back the space and power of the CSLA. Based mostly on this modification CSLA designs have been developed and will be compared with the regular CSLA design. The projected style has reduced space and power as compared with the regular CSLA with solely a slight increase in the delay. This work evaluates the performance of the projected styles in terms of delay, area, power, and their product by hand with logical effort and through custom style and layout in zero. 18-m CMOS method technology. The results analysis shows that the projected CSLA structure is healthier 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. Literature Survey

On the basis of necessities such as space, delay and power consumption a number of the advanced adders square measure Ripple Carry Adder, Carry look-Ahead Adder and Carry choose Adder. Ripple Carry Adder (RCA) shows the compact style however their computation time is longer. Time essential applications build use of Carry Look-Ahead Adder (CLA) to derive quick results however it leads to increase in space. However the carry choose adder provides a compromise between tiny the tiny the little] spaces however longer delay of RCA and giant 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 known as worst case delay path through N stages. Fig. one shows the diagram of ripple carry adder. Currently as the price of N will increase, delay of adder can additionally increase in a linear manner. Therefore, RCA has the lowest speed amongst all the

IJOEET
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.
TRUTH TABLE OF 4-BIT BINARY TO EXCESS-1 CONVERTER

<table>
<thead>
<tr>
<th>Binary Logic</th>
<th>Excess-1 Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_0 \ B_1 \ B_2 \ B_3$</td>
<td>$X_0 \ X_1 \ X_2 \ X_3$</td>
</tr>
<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.

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](image)

Fig. 4 16-bit conventional carry select adder

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...
Fig. 6. (a) Conventional CSLA is the input operand bit-width. (b) The logic operations of the RCA is shown in split form.

Where HSG, HCG, FSG, and FCG represent half-sum generation, half-carry generation, full-sum generation, and full-carry generation, respectively.

IV. Proposed Adder Design

The proposed CSLA is based on the logic formulation given in (4a)–(4g), and its structure is shown in Fig. 3(a). It consists of one HSG unit, one FSG unit, one CG unit, and one CS unit. The CG unit is composed of two CGs (CG0and CG1) corresponding to input-carry ‘0’ and ‘1’. The HSG receives two n-bit operands (A and B) and generate half-sum words s0and half-carry word c0of width n bits each. Both CG0and CG1receive s0 and c0 from the HSG unit and generate two n-bit full-carry words c01 and c11 corresponding to input-carry ‘0’ and ‘1’, respectively.

The logic diagram of the HSG unit is shown in Fig. 3(b). The logic circuits of CG0and CG1are optimized to take advantage of the fixed input-carry bits. The optimized designs of CG0and CG1are shown in Fig. 3(c) and (d), respectively.

The CS unit selects one final carry word from the two carry words available at its input line using the control signal cin. It selects c01 when cin = 0; otherwise, it selects c1. The CS unit can be implemented using an n-bit 2-to-1 MUX. However, we find from the truth table of the CS unit that carry words c01and c1follow a specific bit pattern. If c01(i) = 0, then c1(i) = 1, irrespective of s0(i) and c0(i), for 0 ≤ i ≤ n−1. This feature is used for logic optimization of the CS unit.

The optimized design of the CS unit is shown in Fig. 3(e), which is composed of n AND–OR gates. The final carry word c is obtained from the CS unit. The MSB of c is sent to output as cout, and (n−1) LSBs are XORed with (n−1) MSBs of half-sum (s0) in the FSG [shown in Fig. 3(f)] to obtain (n−1)MSBs of final-sum (s). The LSB of s0is XORed with cin to obtain the LSB of s.

Fig. 7. (a) Proposed CS adder design, where n is the input operand bit-width, and [*] represents delay (in the unit of inverter delay), n = max (t, 3.5n+2.7). (b) Gate-level design of the HSG. (c) Gate-level optimized design of (CG0) for input-carry=0. (d) Gate-level optimized design of (CG1) for input-carry=1. (e) Gate-level design of the CS unit. (f) Gate-level design of the final-sum generation (FSG) unit.

V. RESULTS
The implemented design in this work has been simulated using Verilog-HDL (Modelsim). The adders (of various size 16, 32, 64) 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.

Simulation Results:
16 bit:

64 bit:

Fig 8. 16 bit CSLA

64 bit:

Fig 9. 64 bit CSLA

Synthesis Results:
RTL schematic:
16 bit:

Fig 10. 16 bit CSLA

64 bit:

Fig 11. 64 bit CSLA

Design Summary:
16 bit:

Fig 12. 16 bit CSLA
VI. Conclusion

Power, delay and space square measure the constituent factors in VLSI style that limits the performance of any circuit. This work presents a straightforward approach to scale back the space, delay and power of CSLA design. The traditional carry choose adder has the disadvantage of additional power consumption and occupying additional chip space. The projected SQRT CSLA mistreatment common Boolean logic has low power, less delay and reduced space than all the opposite adder structures. It's additionally bit quicker than all the opposite adders. During this method, the semiconductor count of projected SQRT CSLA is reduced having less space and low power that makes it straightforward and economical for VLSI hardware implementations.

References