Design and Implementation of Carry Select Adder Using Binary to Excess-One Converter

PALURI NAGARAJA ¹
Nagaraja.paluri@gmail.com ¹

KANUMURI KOTESWARA RAO²
koti_r@yahoo.com²

¹PG Scholar, Dept of ECE, Prakasam Engineering College, Kandukur, Prakasam(Dt), A.P, India

²Associate Professor, Dept of ECE Prakasam Engineering College, Kandukur, Prakasam(Dt), A.P, India.

Abstract:
In many data-processing processors Carry Select Adder (CSLA) is one of the fastest adders used to perform arithmetic functions. The upcoming technologies depict 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-μm 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 areas 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 \( C_{in}=0 \) and \( C_{in}=1 \), then the final sum and carry are selected by multiplexers.

**III. BEC (Binary to Excess-1 converter)**

The basic idea of this work is to use Binary to Excess-1 converter (BEC) instead of RCA with \( C_{in}=1 \) 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 \( C_{in}=1 \). 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. 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 \( C_n=1 \) with BEC. This replaced BEC performs the same operation as that of the replaced RCA with \( C_n=1 \). 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.
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 and half-carry word C0 of width n bits each. Both CG0 and CG1 receive 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 CG0 and CG1 are optimized to take advantage of the fixed input-carry bits. The optimized designs of CG0 and CG1 are 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 C0(1) 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 C01 and C11 follow a specific bit pattern. If C01(i)=‘1’, then C11(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 S0 is XORed with Cin to obtain the LSB of S.

V. 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.
VI. 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