EC311 Introduction to Logic Design Spring 2017
Lab 1: Hardware Addition-Subtraction

Binary and BCD Adders-Subtractors

Goals

- Introduction to ISE and programming the Xilinx FPGA board.
- Introduction to Verilog modular code design.
- Designing simple combinational circuits.

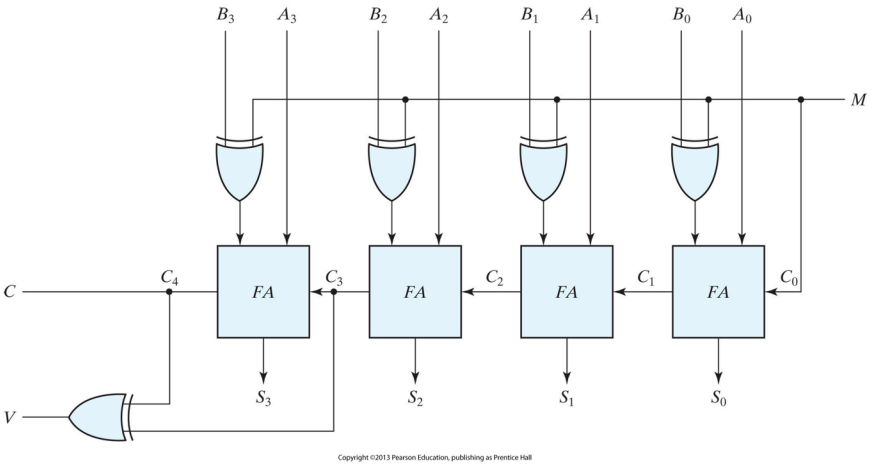
Overview

In this lab you will design two different adders - binary and BCD. Please refer to the textbook for more information about these adders.

Tasks

1. 4-bit Binary Adder-Subtractor

Design a 4-bit binary adder-subtractor. You can use the design shown below (taken from page 142 of the Mano, Ciletti textbook):



Your design has to abide by the following requirements:

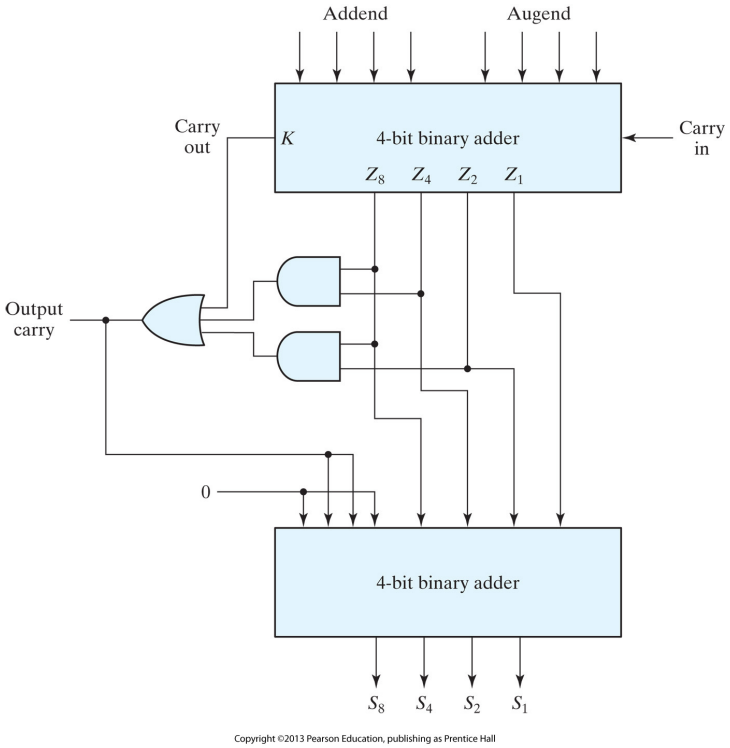
(a) Be modular (consisting of other submodules).
(b) Include a control for switching between addition and subtraction (M in the above figure).
(c) Use onboard switches to get inputs and onboard LEDs to output the result.
(d) Be able to detect an overflow.

Tip. Design a 1-bit adder first. You can use structural or behavioral programming to design a Full-Adder. Test your design before you continue.

Another Tip. Make a schematic with Full-Adders, primitives, wires, etc. on paper; name all the wires and blocks. That will help you avoid confusion when connecting the blocks.

2. BCD Adder

Design a Binary Coded Decimal (BCD) adder. Note that a BCD digit is always 4-bit long. This means that your adder will be able to add only 2 BCD digits ranging from 0 to 9 (because there are only that many switches on the board). To complete this assignment you will need to develop an algorithm for BCD addition. You can try designing a Truth-Table, K-Map, or just use your guts. You may also refer to the figure below (taken from page 146 of the Mano, Cilleti textbook), in which case your design should incorporate instances of your 4-bit binary adder/subtractor from part 1.



You may use behavioral Verilog (including if...else; case...endcase; statements.).

The BCD adder can be implemented in several ways. Select an implementation that you understand. Remember: the purpose of the lab is to learn the Verilog HDL and the CAD environment, not to develop a novel algorithm.

Deliverables

- Demo your simulation waveforms and final design uploaded to the Xilinx Spartan-6 FPGA board, for each of the tasks. You can show your design to any of the TAs or Lab

assistants. If for some reasons you cannot demo your work, email the course staff at least 3 (!) days before the deadline.

- Submit your Verilog code, one folder for each of the tasks on blackboard. It is recommended to tar or zip all your code together before uploading it.