EC311 Introduction to Logic Design Spring 2017
Lab 2: Simple ALU Design

Multiplication and Multiplexers

Goals

- To gain experience with modularity of an HDL system.
- To utilize behavioral Verilog and conditional expressions.

1. Overview

In this lab you will design a very simple 4-bit Arithmetic-Logic Unit (ALU), that will be able to perform addition and multiplication, among other functions. The inputs to the ALU are two 4-bit numbers, A and B, and a 2-bit operation code, S. The output is an 8-bit result, Y.

Tasks

1. 4-bit Multiplier

Design a 4-by-4 multiplication module (called mod3 in figure 1), with two 4-bit inputs and one 8-bit output. You are allowed to use behavioral Verilog, so this module should be fairly simple.

2. Buffer

Our ALU includes a concatenation functionality, which is implemented in a Buffer module. Design a buffer module, which includes two 4-bit inputs and one 8-bit output. The output is the concatenation of the two inputs.

3. Binary Adder

The 4-bit binary adder module includes two 4-bit inputs and one 8-bit output. For this module, you may use your 4-bit adder-subtractor from the previous lab. If you choose to do so, set the mode-controller (M) to 0 for addition.

4. BCD Adder

The 4-bit BCD adder module includes two 4-bit inputs and one 8-bit output. For this module, you may again use your 4-bit BCD adder from the previous lab.

5. Multiplexer

Multiplexer is a device that has N control inputs and 2N data inputs. In figure 1, the multiplexer is called my_mux, which has four 8-bit inputs (A0…A3), and two 1-bit controls (S0,S1). Whenever {$\overline{S}$1,S0} = 2'b00, the output Y = A0; if {S1,S0} = 2'b01, then Y = A1 and so on.

You'll need to design such a multiplexer for the ALU. The easiest way to design a multiplexer in Verilog is to use a case statement. You may refer to your textbook for details how to use it. Don't forget that the case statement in Verilog has to be included in a procedural block (usually an always statement).

6. Final Design

Your final design should instantiate five modules: Buffer, Binary Adder, BCD Adder, Multiplier, and Multiplexer. It should have two 4-bit inputs, A and B, a 2-bit control input (shown as S1, S0), and an 8-bit output.

The 2-bit operation code control input is specified as follows:
00 – Concatenation
01 – Binary addition
10 – BCD addition
11 – Multiplication

Refer to the block diagram in figure 1.

Deliverables

- Demo your simulation waveforms and final design uploaded to the Xilinx Spartan-6 FPGA board. 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.

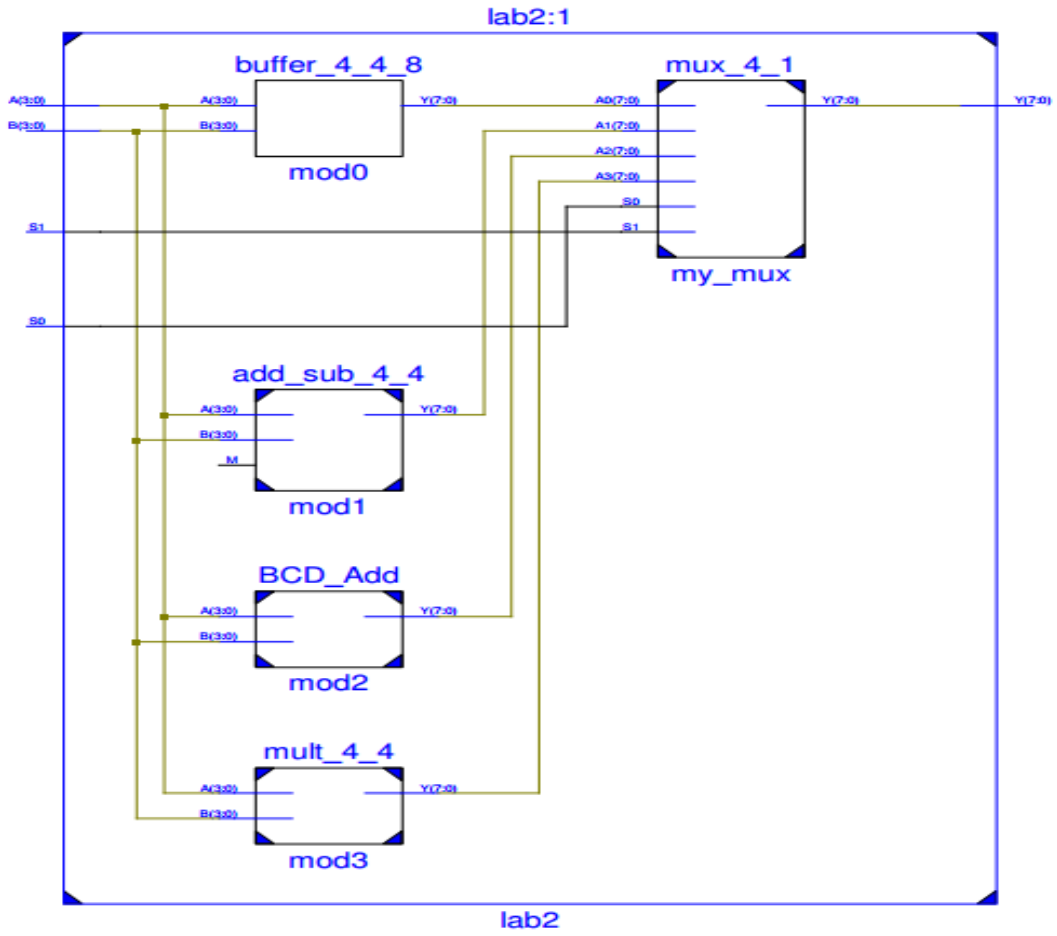

Figure 1: Block diagram for Lab 2