Department of Electrical and Computer Engineering
University of California, Davis

Lab 2: An Adder-Based Design

I. Prelab

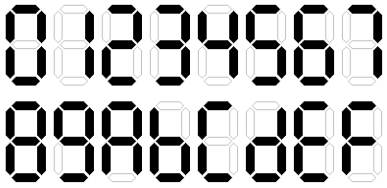
Complete the following and submit your work at the beginning of your lab session.

1. Read the document: *Notes on active high/low inputs/outputs for the DE-10 Lite* on the course web page
2. [15 pts] Complete a preliminary design of your `seg7.v`, `fa.v`, and `add4bit.v` (a draft of your verilog code)

Details not specified in this Lab should be chosen by you and plainly explained in your lab report.

II. A Hexadecimal display driver

Design and implement a purely-combinational (non-recursive) 7-segment display driver module seg7.v in verilog that inputs a 4-bit signal and drives 7 outputs that correspond to 0–9, A–F with a **low** output turning **on** an LED segment and a **high** output turning the corresponding segment **off (active low)**. Use a single case statement.



III. A Combinational 4-bit Adder

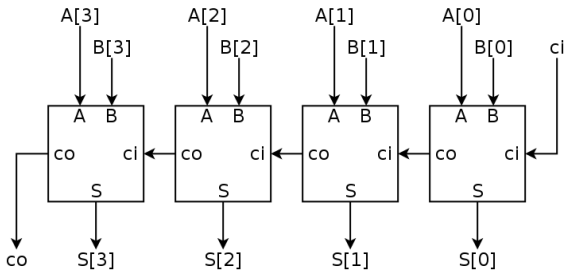
Design and implement the following modules in verilog:

1. Write a module `fa.v` for a full-adder in Verilog using either `wires` or `regs`. A full-adder has inputs *a, b,* and *ci* (carry-in) and produces outputs *s* (sum) and *co* (carry-out). Its Boolean expressions are as follows:

$$s = a \oplus b \oplus c_i$$

$$c_o = ab + ac_i + bc_i$$

2. Create a module `add4bit.v` in Verilog that implements a 4-bit ripple-carry adder by instantiating four full-adder circuits. The adder is built with a ripple-carry adder design (consult a reference if you are rusty with the details). An example block diagram for a 4-bit ripple-carry adder is shown:



The adder has the following ports:

Signal Name	Number of Bits	Input/Output	Description
a	4	Input	Input operand
b	4	Input	Input operand
ci	1	Input	LSB Carry-In
s	4	Output	Output Sum
co	1	Output	MSB Carry-out

IV. Top-level design

Use SystemBuilder to create a new Quartus project and top-level module called *top.v* with the following features:

1. It instantiates two copies of add4bit.v
2. One 4-bit input of each 4-bit adder is tied to four SW switches—both adders connected to the same 4 switches
3. The other input of one adder is tied to 0000. The other input of the other adder is tied to 0001.
4. Each adder's output is tied to a seg7.v circuit and then connected to a HEX display.

When working correctly, one hex display will show the value on the four switches and the other active hex display will show the same value plus one. For example, if the switches are set to 1001, the two active hex displays should display "9" and "A". If they are set to "1111", the displays should show "F" and "0".

V. Implementation and Verification on the DE10-Lite

Since you are not using Modelsim to debug your design, identify the most important internal signals and connect them to the LEDR and unused HEX LEDs for debugging purposes. Download your design onto the DE10-Lite board and verify it works correctly.

VI. Submitted Work [100 pts total]

[15 pts] **Prelab**

[60 pts] **Lab Checkoff: FPGA board**

Demonstrate your design compiling, downloading, and operating to your TA. Your TA will ask you to generate and will record a `certutil` hash of your top-level files and this must match the hash of the files you upload to canvas so no file modifications are possible after your demo. Calculate the hash using the following steps:

1. in Quartus, double click one or two key modules in the Project Navigator Hierarchy pane to open their Verilog
2. right click the file's tab and select Copy Full Path
3. open a Command Prompt window (by typing "cmd" into the Windows search bar)
4. type "`certutil -hashfile filepath`" where *filepath* is the pasted file path that was copied earlier

[25 pts] **Lab Report**

Draw a detailed circuit diagram of your add4bit.v and top.v in either pdf or jpg format.

Submit all Verilog hardware and testbench code that you wrote. Do not include any code that you did not write such as files generated by Quartus or IP components.

a) Upload a copy to Canvas by performing the following steps by the end of your lab session—this is essential to receive any credit for the entire lab.

1. Make a folder on your computer
2. Copy all files to be submitted into the folder
3. "zip" the folder into a single .zip file
4. Log onto Canvas, click Assignments, find the correct lab number
5. Upload the .zip file

2021/04/06 seg7.v must be purely-combinational, do not use recursive algorithms commonly found online!
2021/04/01 Posted