LAB Assignment #3 for ECE 443

Assigned: Wed., Sept 16, 2009
Due: Mon., Sept. 23, 2009

Description: Implement the Adder component of an ALU for the microcontroller that you will build in the next couple labs.

All components of the ALU are strictly combinational (the topic of the lectures for the next couple weeks). This lab will focus on building an adder with some additional functionality that will become evident later as we add other components to the microcontroller. The input/output signals to adder are given as follows:

Input:

- A, B: input buses (12 bits wide)
- add_ins: Add A and B
- sub_ins: Subtract B from A (A-B)
- skwlf_ins
- skwgf_ins

Output:

- ADD_OUT: output bus (12 bits wide)
- skw_success: Boolean value that depends on instuction and MSB of adder

The input operands, *A* and *B*, to the **AddUnit** are 12 bits wide, as is the output of the AddUnit, labeled ADD_OUT. You need to implement a *ripple carry* adder, the simplest (and smallest) type of adder. The adder can be configured to subtract by XORing the B operand bits and setting the **carry-in** bit of the Adder to 1. You will perform subtraction under one of three conditions, if any of *sub_ins* or *skwlf_ins* or *skwgf_ins* are set to '1'. The *skwxx* signals will indicate skip instructions that will be dependent on the result of a subtraction, i.e., *skwlf*: skip if w less than f (to be discussed in a future lab).
The output signal *skw_success* needs to be set as follows:
If *A* < *B*, then MSB of adder is '1' after subtraction, otherwise it is '0'.
If *skwlf_ins* is '1', then *skw_success* should be set to '1' if the MSB of adder is '1', else '0'
If *skwgf_ins* is '1', then *skw_success* should be set to '1' if the MSB of adder is '0' AND the low order 11 bits are NOT all 0s, else '0'

(NOTE: The *add_ins* input does NOT connect to anything (at this point) but please leave it in the port list. Also, leave the *carry_out* signal of the high order 1-bit adder unconnected)

Be sure to use hierarchy in the implementation of the 12 bit adder, i.e., component instantiation. For example, the top-level of the ripple carry adder should be implemented using 12 full adders. The logic for a full adder can be obtained from any basic digital logic text (or on the web)

I am currently working out how you will actually perform in-class demonstration using your Digilent boards (lab #4). This is likely to involve the use of the serial port. I've put UART slides up on my website for those who have time to forge ahead.

Laboratory Report Requirements:

1) Turn in a commented copy of your VHDL code.
2) Turn in the top-level schematic diagram that represents the synthesized schematic of the code.
3) Run a simulation that shows the inputs and output behavior of the circuit.
4) Include a set of ‘sample’ waveforms in your report.

Grading:

The lab grade is derived from your lab report. 75 points (of the 100 pts) will be given for simulation, which includes the test bench and waveform graphics with proper annotation. 20 pts will be given according to how well the VHDL code is written, e.g., indentation and coding style, and 10 pts will be given for comments. Bonus points will be given to any implementation feature that goes above and beyond the requirements.