Computer Architecture

Assignment 5

Design a controller-datapath system to compute the n^{th} number in the Fibonacci sequence. Don't worry about overflow; assume your system can handle whatever size number is required. You may use any datapath component: register, mux, alu, and comp.

Go through all steps of the design process: (1) write out the algorithm in pseudocode, (2) draw the datapath, being sure to label all control signals and status signals, (3) draw the state diagram, (4) build the state table from the diagram and datapath, and (5) write the next state and output equations (you do not need to minimize them or draw the gate-level diagram.)

I've posted the recent boardwork to Sakai so you can see the full worked out notes I've written during class. The filename is architecture 2.pdf.

Sample Problems

Due to our time constraints and the relatively fast turnaround for this assignment, the following problems are NOT assigned. However, they are indicative of the kinds of things you may see on an exam (something like Problem 3 will 100% be on the exam) and since it's hard to find practice problems for this material I've put these here in case you want to take a stab at them. I plan to go over these problems on Tuesday in review for the exam.

Sample Problem 1

In class, we designed a Full Adder (FA). The FA takes three bits as input and outputs their sum and a carry out. A Half Adder (HA) takes two bits as inputs and outputs their sum and carry out.

(a) Design a Half-Adder. Build the truth table, find the SOP form of the output equations, simplify to the minimal sum, and give the gate-level implementation.

(b) Show how to build an FA out of two HA's.

Sample Problem 2

Design a device to detect whether overflow has occurred in two's complement arithmetic. Assume you have a 32-bit adder/subtractor unit so that your available inputs to your device are all the bits of the addends, all the bits of the sum, as well as the carry out of the adder/subtractor unit (note: you will not want to use all these inputs, so choose which are needed to determine whether overflow occurred).

Show all the steps of the design process: truth table, SOP equations, simplified sum, and gate-level implementation.

Sample Problem 3

Give the IEEE single-precision floating point format for the decimal number 127.25.