Design HW I

UCLA
Winter 2013
Due Date: Thu January 31, 2013 @ 5PM

Prof. Mani Srivastava

EEM16/CSM51A - Logic Design of Digital Systems

NOTE: This homework is to be done individually using LogiSim.

SUBMISSION PROCEDURE: You need to submit your solution in electronic form at http://nesl.ee.ucla.edu/courses/eem16/2013w/submissions . You would need to use the user id and password handed out in the first lecture.

NO LATE SUBMISSION. Only exceptions would be for medical emergencies based on a certified note from a doctor.

Following table to be filled by the course staff only:

Problem	Maximum Score	Your Score	Comments
Part A			
TempMon correctness	7.5		
TempMon design quality	5		
TestBench1	5		
Testing of Design	5		
Part B			
TempMonWithAlarm correctness	7.5		
TempMonWithAlarm design quality	5		
TestBench2	5		
Testing of Design	5		
Submission of required material	5		
Total	**50**		

In this design homework you have to use Logisim to design a simple temperature monitoring system. The primary input to your system is a temperature reading that comes from an external temperature sensor in the form of a 3-bit binary signal TEMP[2:0] that encodes the temperature according to following table:

TEMP$_2$	TEMP$_1$	TEMP$_0$	Temperature in °C
0	0	0	68
0	0	1	70
0	1	0	72
0	1	1	74
1	0	0	76
1	0	1	78
1	1	0	80

Part A:

Your system needs to display the temperature in °C as a two digit decimal number using two 7-segment displays. Design a circuit named *TempMon* in Logisim that takes TEMP[2:0] as input, and produces two 7-bit control signals, one for left decimal digit and the other for the right digit. We will name these LEFT[6:0] and RIGHT[6:0] respectively. Bit i of LEFT[6:0] and RIGHT[6:0] correspond to the segments in a 7-segment display according to the following pattern:

```
 6666
1    5
1    5
1    5
 0000
2    4
2    4
2    4
 3333
```

When designing this circuit, you must only use NOT, NAND, and NOR gates that are available through the *Gates* library in Logisim. Your goal would be to minimize the cost of the circuit as measured by the number of literals, i.e. the total number of gate inputs in your circuit. Say your circuit used 3 inverters, 1 2-input NAND gate, and 2 3-input NAND gate, then you'd list them

Invert = 3, NAND2 = 1, NAND3 = 2. Then the total number of literals would be: 3x1 + 1x2 + 2x3 = 11 (note: 3x1 comes from 3 inverters of 1 input each; 1x2 comes from 1 NAND gate of 2 inputs each; and 2x3 comes from 2 NAND gates of 3 inputs).

If your circuit is given an invalid input (i.e. one that is not specified in the table, it should display - - in the two displays.

Now to test the circuit that you just designed, create another circuit in Logisim with the name *TestBench1*. In this circuit you should instantiate a copy of the *TempMon* circuit you designed, and then hook it up to input and output devices to test whether it works correctly. Conveniently, Logisim provides a *7-Segment Display* module in its *Input/Output* library. This module has eight input pins, one for each of the seven segments as well as an eight one for the decimal dot at the bottom right of that display. You can leave it unconnected. You would need to instantiate two of the *7-Segment Display* modules and connect their inputs to the LEFT[6:0] and RIGHT[6:0] outputs from the *TempMon* circuit.

To provide test inputs to the TEMP[2:0] inputs of the *TempMon* circuit, simplest for you would be to connect them to input pins, and then use the Logisim feature whereby you can manually alter the logic value at the pin using the poke tool (the one at the top left with the symbol of a hand with a finger poking out). Exhaustively test for all input values.

Part B:

Now, enhance your design to create a new circuit named *TempMonWithAlarm* so that it takes from an operator a second 3-bit temperature input TEMPSETPOINT[2:0] that indicates a temperature threshold, and outputs a binary signal called TOOHOT if the temperature from the sensor is greater than or equal to the temperature threshold. In other words, TOOHOT should be 1 if TEMPSETPOINT[2:0] is <= TEMP[2:0], and 0 otherwise.

(Note: an earlier version of this homework had a typo whereby <= was accidentally written as >=, and in effect implemented a TOOCOLD functionality. If you already implemented the >= functionality, please put a note to that effect and we won't penalize you).

When designing this circuit, you can use any modules available in Logisim's library - i.e. You are no longer restricted to NOT, NAND, and NOR gates.

To test this new circuit, create a new testbench *TestBench2* where you use another set of input pins to control TEMPSETPOINT[2:0] and a LED (Logsim provides LEDs under the *Input/ Output* library) to see the alarm.

Digression:

If you want to have a bit of fun and have an audio alarm instead of LED, you can download from https://sites.google.com/site/immibis/logisim-stuff/sound-library a file called buzzer.jar and then load it into Logisim through the *Project → Load Library → JAR Library…* menu. A dialog box will pop up asking for a class name, and you should enter "immibis.buzzer.Library" (don't enter the double quotes). This will add a couple of new modules to Logisim under a library named *Sound Library*, one of which is called *Simple Buzzer*. You can instantiate it in your *TestBench2* in addition to the LED and connect to TOOHOT using the left pin of *Simple Buzzer*, and then your computer ought to make a buzz sound when TOOHOT=1.

What to submit?

Please follow these instructions carefully.

1. Save your design with the name *dhw1*. This will create a file called *dhw1.circ.*

2. Additionally create a single PDF file named *dhw1.pdf* with the following

- Your name and student id on the top of the first page
- Color screen capture of Logisim showing schematics of each of the following, one per page: *TempMon, TestBench1, TempMonWithAlarm, and TestBench2.*
- Screen capture of Logisim showing *TestBench1* and *TestBench2* with a couple of sample inputs your choice for each of them. Again, one per page. Please annotate with a sentence or two describing what is happening.
- A page with a table that tallies all gates of different types and # of inputs your used for the *TempMon* circuit, and shows the total cost of the *TempMon* as measured by the approach described in the previous page. Basically, the cost of each inverter is 1, and the cost of a k-input NAND or NOR gate is k.

Put both *dhw1.circ* and *dhw1.pdf* (and any other additional file you deem necessary) into a single folder with the name *dhw1_yourlastname_yourfirstname* (e.g. *dhw1_smith_john*), and then zip that folder to get a file named *dhw1_yourlastname_yourfirstname.zip.* Upload this file by logging in at http://nesl.ee.ucla.edu/courses/eem16/2013w/submissions using the password and id that were given in the first lecture.