CSCE 611
Lab 3
MIPS Register File
Due Date: 10/10 (by midnight)

Objective

In this lab you will design a 32 x 32 bit register file having the following characteristics:

- two asynchronous read ports
- one rising-edge-active write port
- writes bypass reads: the register file will return the value being written during a simultaneous read and write to the same register
- register 0 always contains the value 0
- the most recently-written value of register 30 is a separate output
- when register 30 is read, the result will match the value of a separate 32-bit input
- I/O specification:
  - **`input [4:0] readaddr1, readaddr2, writeaddr`**
  - **`input clk, we`**
  - **`input [31:0] writedata, reg30_in;`**
  - **`output [31:0] readdata1, readdata2, reg30_out`**

Design a self-checking test bench for the register file that will test the following cases (at a minimum):

1. Write-then-reads (read after write) of a single register (excluding register 0 and 30) on different cycles
2. Write register 30
3. Read register 30
4. Write register 0, then read register 0 (should stay at zero)
5. Write-and-read of the same register on the same cycle involving read port 1
6. Write-and-read of the same register on the same cycle involving read port 2

The testbench must use testvectors, applied once per cycle. The test bench should drive the clk input.

Once you've tested your register file, test it on the DE2 board. Use the following procedure:

- instance eight 7-segment decoders (developed in Lab 1) and connect their inputs to each group of four consecutive bits of the "reg30_out" output
- connect the outputs of the eight 7-segment decoders to the eight HEX digits on the DE2 board (output signals HEX0[6:0], HEX1[6:0], …)
- connect the RegFile's input clk to CLOCK_50
- hardwire the RegFile's we to 1
- hardwire the RegFile's readaddr1 input to 0
- hardwire the RegFile's readaddr2 to 30
- hardwire the RegFile's writeaddr to 30
- connect the RegFile's readdata2 output to its writedata input

You should be able to set the hex files using the 18 switches. Note you'll only be able to set the low-order 5 HEX values, and the most significant digit will only display values 0-3.

Submit your projects through the course Moodle site (http://dropbox.cse.sc.edu). Upload all Verilog source and testvector files.