Out: September 15th, 2016
Due: September 22nd, 2016

Goal

To integrate your EE 431.3 microprocessor into the CME 435 design methodology. To develop an initial test plan for verification of your microprocessor.

Procedure

Copy your EE 431.3 microprocessor Verilog files to a subdirectory of your CME_435 directory on your cabinet drive (i.e., H: drive). Set up a directory structure identical to that used previously for your RAM and ROM modules. Reuse code from these exercises to save some of your development time.

Use the included Data_Memory_RAM.v, which is the same working RAM that was provided for use in Lab #1.

In addition to these files you will be using a third-party Program_Memory_ROM.sv. The complete SystemVerilog source code for this file will be generated by the modified version of the CME 431 microprocessor assembler that is included in this assignment package.

To use this modified assembler, open in a web browser the EE431AssemblerApplet.html file under the cme435_mod_assembler directory in this assignment package.

NOTE: The default Java security settings are constantly being tightened. You may need to drop your Java security to the lowest level in order to run this assembler app because it is not digitally signed. On Windows, type ‘configure java’ into the start menu search bar, click on the Security tab in the config dialog, and set the security level to Medium. On Mac, click on the Apple icon in the upper left, go to System Preferences, click on the Java icon to access the Java Control Panel. **Remember to restore your Java security settings after you are done using the assembler app.**

After compiling an assembly language program you will SystemVerilog code for the Program_Memory_ROM module in the bottom window. Cut and paste this code into a Program_Memory_ROM.sv file in your directory. Every time you compile an assembly language program remember sure to copy and paste the new Program_Memory_ROM code into the .sv file, as it is re-generated with each assembly compile.

Study the Program_Memory_ROM.sv code to satisfy yourself that it is correct. It should be <u>similar</u> to the Verilog ROM module that was provided for use in Lab #1. If in doubt, set up a subdirectory in your CME_435 directory, change the initialization block to match your Lab #1 ROM’s initialization, and verify its operation.

For the purposes of this assignment we will be looking at very low level functionality of your microprocessor. Namely, **can you read the data on the i_pins and output it to the o_reg?**

Create an assembly language program called Simple_IO.asm to accomplish this. Additionally the following question will arise:

> How do you synchronize the changing of data on the i_pins input in relation to your microprocessor code? Recall that the EE 431.3 microprocessor will execute an instruction **every** clock cycle after coming out of reset. We would like to have your verification test run for the shortest possible time while at the same time testing all possible i_pins input data.

Create a test plan spreadsheet based on the included CME_435_Testplan_Template.xls. Identify the testable device features and specifications, and write them in the Specification/Feature column. These can be copied from the specification document verbatim if they are already appropriately descriptive. In the reference column, add the section and/or page number in which the specification was found. In the Test Scenarios column, describe the stimulus that you would apply to exercise the identified specification. This may be a high-level or low-level description depending on the scenario you are creating. If you have more than one scenario, go to a new row in the same column. The Expected Results column should then be used to describe what you would expect to observe to know if the test passed or failed.

Additionally specify any operating environment requirements that are applicable, unless otherwise noted. For example, "The CLK signal shall be driven at 1MHz for the duration of the tests". You can write these general environmental assumptions on a separate worksheet in the same spreadsheet.

Note that at this point, you do NOT need to specify verification technologies or SystemVerilog code/modules that need to be written.

Make sure that you have a test plan in place for the test(s) described before you attempt to perform the actual verification on the modules. This verification will be done in Lab #2 on September 23rd. As well, additional tests will be done at that time.

Deliverable

Your assembly language program and your test plan spreadsheet for this aspect of your EE 431.3 microprocessor.

Package the required files into a zip archive that includes your name in the filename, and email to the course marker Mehedi Hasan (mehedi.hasan@usask.ca) by midnight on September 22nd.