CSE 141 Lab 2: Designing the CPU Internals

• Please turn in hard copies of Lab 1 up front NOW
• Make sure all lab partners names and ID numbers are on the written lab

What to do next

• Read Lab 2
  – Especially the last questions
• Do the my list of suggested actions by THURS! (in a few slides)
• Read the webboard
  – New Lab 2 Questions section
• Come to class or TA office hours to retrieve comments on first lab
  – You can’t change your ISA at all unless we tell you to!
  – But you can make a section in your lab of “things we wish we could change”
Goals of Lab 2

- Design and implement CPU internals including
  - Computation components (ALU)
  - Internal storage components (registers?)
- Test individual instruction behavior
  - Decode instruction
  - Fetch necessary data from internal storage
  - Compute
  - Store back to internal storage
- Due in 3 weeks! Monday Feb 14

What you DON’T have to do for Lab 2

- Increment the Program Counter
  - You DO need to test any PC calculation instructions, but you can “hand feed” in some sample PC value
  - The PC does NOT have to appear in your Lab 2 design
What you DON’T have to do for Lab 2

- Test “external” memory (instruction or data memory)
  - That will be Lab 3, so no load or store
  - Since your loads and stores can do NOTHING but load or store, you shouldn’t need to test them at all, right?

How do we do that? BEFORE THURS:

- 1) Install Xilinx on your computer (or go to APM 2444 and login and make sure you have access).
  - This is a multi-step process and takes at LEAST 30 min.
- 2) Do the xilinx tutorial provided on the class web site. (Don’t worry that it says version 6.2i, use 6.3)
  - Give yourself 2 hours for this. Do it as a team. Bring snacks.
- 3) Be sure you understand the schematic and simulation (waveform) preparation (I’ll show examples in a minute).
Schematic

- A schematic is a “circuits and wires” diagram that determines the functionality of your processor
  - You are not going to design your processor in one big schematic file.
  - Plan a hierarchy of components that each have
    - Meaning
    - Defined inputs and outputs
    - That are modular and independent (self testable)

High level example
(yours may differ significantly)
So is that all I turn in? NO

- You will turn in a LOT of schematics
- There should be text introducing the schematic section of your writeup that tells the name of each and the relationship of the hierarchy
- Each schematic should have
  - A name
  - Description of what it does
  - Annotations on each part saying what it does
  - Annotations on all inputs and outputs
  - Descriptions of exceptional conditions, etc

How do I test each instruction for Lab 2?

- Since we are not running the whole program, have to provide testing input
  - A way to put a value into the registers so that you can test an add, for example
- You should test MULTIPLE cases for each instruction
  - Boundary cases, etc
- “Run” each instruction by creating a waveform diagram
  - Change the inputs (manually) to test a different instruction each “cycle”
How does one debug a schematic?

- Put in extra output lines that output values “intermediate” to the final results
- Look at them on the waveform diagram…

Waveform Diagrams

- How you test your schematic implementation.
- Set up cycle times and input values
- Check output values to see if answers are correct
- ANNOTATE YOUR DIAGRAM (with pen is fine)
Waveform Timing Diagram

More on this next lab

- Specifically, come in with questions on
  - what you need to build
  - How to organize it
  - How to test it

- Start looking at the questions you need to answer, so you can “think about them” as you are doing your implementing.
USE THE WEBBOARD!