Programming Assignment 4-2: CPU I-Types and Memory

CS141 Spring 2019

Due April 14, 2019 at 11:59pm

I-types and Memory

In this section of the lab we will implement most of our I-types and also loads and stores:

- `andi`
- `ori`
- `xori`
- `slti`
- `addi`
- `lw`
- `sw`

Again, an incremental approach here is best. First implement the `andi`, `ori`, `xori`, `slti`, and `addi` instructions - this will make it easy to load arbitrary values into the register file. Next, test the `sw` function to be able to write data to memory. Finally, implement the `lw` function to get values from memory back. Keeping your assembly tests simple will really help debug until you are sure your datapath state machine modifications are working.

Deliverable (April 14, 2019 at 11:59pm)

Similar to the previous part, in lab you will have to demonstrate your CPU with working I-type instructions, memory instructions, and jump instructions. You will also need to show the assembly files you wrote to test your partial datapath. For this part, your testing methodology is particularly important, because loads and stores might be more involved to validate.

As in previous labs, please submit a description of your testing methodology. Describe the methods you utilized to ensure your CPU is working. What sorts of assembly programs did you write? How did you check the results? What sorts of corner cases did you design the tests for?

Rubric (100 points)

Functionality/feature	Points
Schematic of MIPS Datapath/FSM	15
Testing methodology	15
Simple I-types (andi,ori,xori,slti,addi)	30
Load (lw)	20
Store (sw)	20