Programming Assignment 4-3: Jumps and Branches

CS141 Spring 2019

Due April 19, 2019 at 5pm

Instructions

In this programming assignment, we will finish our CPU! To do this, you will implement the rest of the instructions

- `j`
- `jr`
- `jal`
- `beq`
- `bne`

Again, an incremental approach here is best. First implement the `j`, and `jr`, and `jal` functions. Note that you may need to introduce new states to your state machine beyond what is listed in the book to achieve this. Finally implement the `beq` and `bne` functions, as many students find those more difficult. Yet again, keeping your assembly tests short will really help debug until you are sure your datapath state machine modifications are working. Consider writing tests that implement classic programming control structures (if, else, while, for, etc.)

Deliverable (April 19, 2019 at 5pm)

Similar to the previous part, in lab you will have to demonstrate your CPU with working jump and branch instructions. You will also need to show the assembly files you wrote to test your partial datapath.

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	10
Testing methodology	15
j	15
jr	15
jal	15
beq	15
bne	15