CSE4372/5392 (Spring 2025)
Lab #7

This lab is due by April 4, with a 20% penalty per week day for being late.

In this lab, you will add support for jumps and branches.

The following steps will guide you through this process:

1. Start with the project from Lab 6.

2. Add these signals to the rv32_if_top port list:

```
// from id
input jump_enable_in,
input [31:0] jump_addr_in,
```

3. Alter the register used to generate the PC so that the input to the register is either formed from PC+4 or jump_addr_in depending on the value of the jump_enable_in signal.

4. In the ID stage, generate the unregistered signals that drive jump_enable_in needed by the IF stage. Connect the signals from ID to IF at the project top.

A jump is enabled for the following instructions:
- JAL
- JALR
- BEQ, BNE, BLT, BGE, BLTU, and BGEU when the condition is true

Note that when reading rs1 and rs2 to evaluate the condition for Bx, make sure to use the data forwarded values from Lab 6 as needed.

5. In the ID stage, generate the unregistered signals that drive jump_addr_in needed by the IF stage. Connect the signals from ID to IF at the project top.

The jump address is based on one of the following:
- pc + 2*signex(i[12:1])
- pc + 2*signex(i[20:1])

- rs1 + signex(i[11:0])

Note that when reading rs1 to generate the jump address for JALR, make sure to use the data forwarded value from Lab 6 as needed.

6. Add logic to ID so that the instruction word sent to the EX stage is a either the current instruction word (jump not enabled) or a NOP (jump is enabled). Effectively this eliminates the instruction in the branch delay slot.

7. Write, assemble, and load a simple assembly language program that uses the branch instructions (with and without the branch being asserted), JAL, and JALR. You should be able to do this without any NOPs between the instructions.

8. Demonstrate operation of the instructions through the pipeline showing the expected register results to the TA. Send your Verilog source .v or .sv files (don't send the entire project) to the TA for credit.