

Ve370 Introduction to Computer Organization

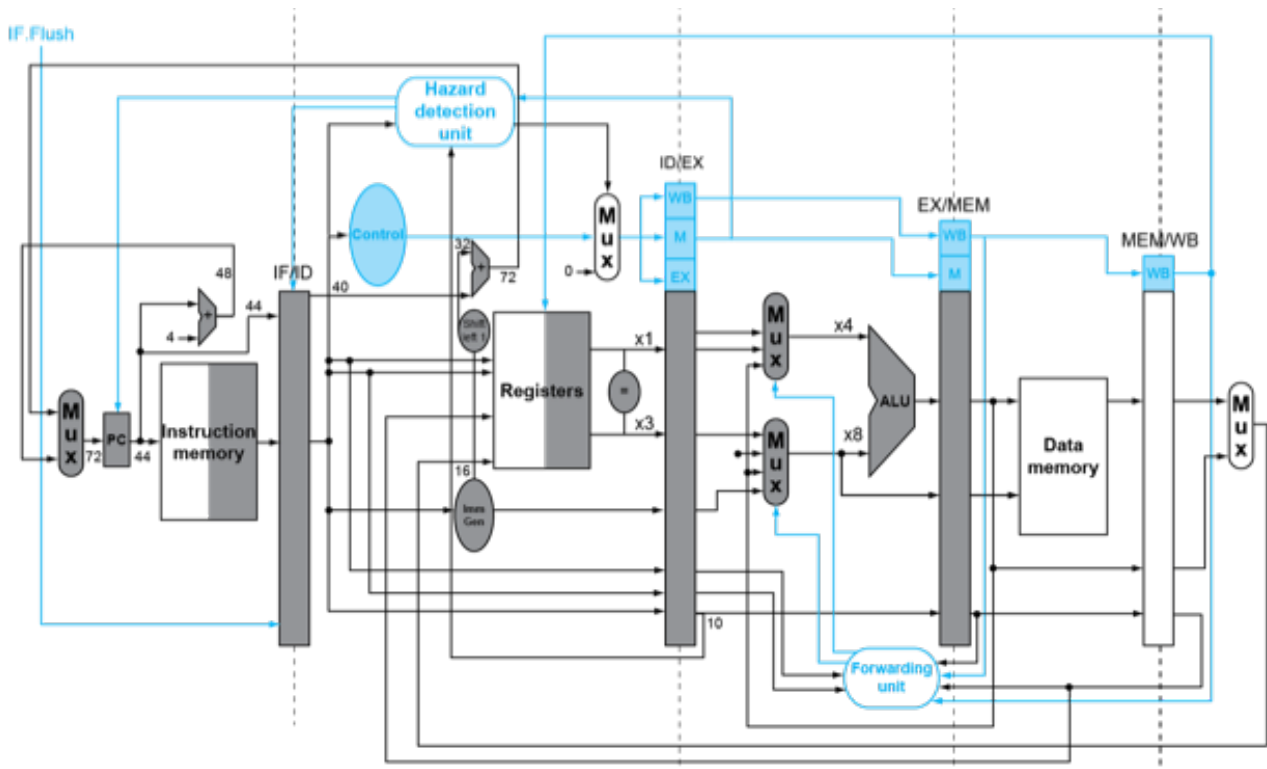
Homework 5

Assigned: June 16, 2022

Due: 4:00pm on July 5, 2022

Submit a PDF file on Canvas

1. (15 points) If we change load/store instructions to use a register (without an offset) as the base address, these instructions no longer need to use the ALU. As a result, the MEM and EX stages can be overlapped and the pipeline has only four stages.
 - (1) How will the reduction in pipeline depth affect the clock cycle time? (5 points)
 - (2) How might this change improve the performance of the pipeline? (5 points)
 - (3) How might this change degrade the performance of the pipeline? (5 points)
2. (25 points) One of the solutions to control hazard is to always stall the instruction following the branch or jump instruction by inserting nop instructions. Using the following diagram as a reference:



- (1) How many nop should be inserted after each beq instruction? (5 points)
 - (2) How can this stall be implemented in hardware rather than in software? Hint: nop instruction is realized as addi x0, x0, 0. (10 points)
 - (3) If the above pipeline is modified to support jal instruction, which would be the earliest stage the jump instruction is identified and jump target is calculated? In that case, how many stalls would have to be inserted? How would the clock cycle time be affected? (10 points)
3. (20 points) Consider the following loop.

```

LOOP: lw x10, 0(x13)
      lw x11, 8(x13)
      add x12, x10, x11
      addi x13, x13, 16
      bne x12, x0, LOOP
  
```

Assume that perfect branch prediction is used (no stalls due to control hazards), that there are no delay slots, that the pipeline has full forwarding support, and that branches are resolved in the EX (as opposed to the ID) stage. Show a pipeline execution (multicycle) diagram for the first two iterations of this loop. Hint: unfold the loop first. Hint : you may use Excel to show the execution diagram.

4. (20 points) The importance of having a good branch predictor depends on how often conditional branches are executed. Together with branch predictor accuracy, this will determine how much time is spent flushing due to mispredicted branches. In this exercise, assume that the breakdown of dynamic instructions into various instruction categories is as follows:

R-type	branch	jal	lw	sw
40%	25%	5%	25%	5%

Also, assume the following branch predictor accuracies:

Always-Taken	Always-Not-Taken	2-Bit
45%	55%	85%

- (1) Stall cycles due to mispredicted branches and jumps increase the CPI. What is the extra CPI due to jumps? What is the extra CPI due to mispredicted branches with the always-



taken predictor? Assume that branch outcomes are determined in the ID stage and that there are no data hazards, and that no delay slots are used. (10 points)

(2) Repeat (1) for the 2-bit predictor. (10 points)

5. (20 points) This exercise examines the accuracy of various branch predictors for the following repeating pattern (e.g., in a loop) of branch outcomes: T, NT, T, T, NT. (T: taken, NT: not taken)

(1) What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes? (5 points)

(2) What is the accuracy of the 2-bit predictor if this pattern is repeated forever? (5 points)

(3) Design a predictor that would achieve a perfect accuracy if this pattern is repeated forever. Your predictor should be a sequential circuit with one output that provides a prediction (1 for taken, 0 for not taken) and no inputs other than the clock and the control signal that indicates that the instruction is a conditional branch. (10 points)