Homework 5

VE370 - Intro to Computer Organization Summer 2022

∗ Name: Huang Yucheng ID: 519021910885

Exercise 1

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?

(2) How might this change improve the performance of the pipeline?

(3) How might this change degrade the performance of the pipeline?

Answer:

(1) Since the critical stage is the ID stage, it will not change the length of ID stage because it only moves MEM and EX stage. So, there will be no reduction.

(2) First, when the two stages are merged, the time required to pass between each other will be reduced, and the lw and sw instructions will not require ALU, which can shorten the time to a certain extent. Second, if EX and MEM execute at the same time, this will reduce some data hazards.

(3) Since the `lw` and `sw` instructions can only operate on the current address, an additional instruction needs to be added to calculate the new address when data offset is required, so the number of instructions will increase and the operation speed will be slowed down.

Exercise 2

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?

(2) How can this stall be implemented in hardware rather than in software? Hint: nop instruction is realized as `addi x0, x0, 0`.

(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?

Answer:

(1) one `nop`

(2) When a B-type is detected, a signal is used in the IF stage that affects the control of PCSrc, which can stop the read of PC, so that it could be stalled. This is to ensure that on the next clock cycle, the PC can load the correct target. At the same time, the signal to clear the IF/ID pipeline register is used and we could put a nop here.

(3) The earliest detected stage can be the IF stage. In this case, there is no need to insert a stall, because the previous instruction has not progressed to the EX stage and can jump directly. But the IF stage may be extended due to judgment and manipulation of jal.

Exercise 3

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.

Answer:

lw x10, 0(x13)	IF	ID	EX	MEM	WB									
lw x11, 8(x13)		IF	ID	EX	MEM	WB								
add x12, x10, x11			IF	ID	nop	EX	MEM	WB						
addi x13, x13, 16				IF	nop	ID	EX	MEM	WB					
bne x12, x0, LOOP					nop	IF	ID	EX	MEM	WB				
lw x10, 0(x13)							IF	ID	EX	MEM	WB			
lw x11, 8(x13)								IF	ID	EX	MEM	WB		
add x12, x10, x11									IF	ID	nop	EX	MEM	WB
addi x13, x13, 16										IF	nop	ID	EX	MEM
bne x12, x0, LOOP											nop	IF	ID	EX

Exercise 4

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.

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

Answer:

(1) Extra CPI for jal: $5\% = 0.05$

Extra CPI for branches: $25\% \times (1 - 45\%) = 0.1375$

(2) Extra CPI for jal: $5\% = 0.05$

Extra CPI for branches: $25\% \times (1 - 85\%) = 0.0375$

Exercise 5

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?

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

(3) Design a predictor that would achieve a perfect accuracy if this pattern is repeated forever. You 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.

Answer:

(1) Always taken: 0.6

Always not taken: 0.4

(2) A 2-bit predictor has a total of four parts. We set the lower left corner to 1, the lower right corner to 2, the upper right corner to 3, and the upper left corner to 4. Initially, we think that the prediction is at the position of 4 in the upper left corner, then the first cycle, first T, the prediction is still 4. Then NT, predict 3, T, predict 4, T, predict 4, NT, predict 3. The prediction of this cycle (44344, the correct case is TFTTF) enters the second cycle, the initial is 3, T, prediction 4, NT, prediction 3, T prediction 4, T prediction 4, NT, prediction 3. Forecast for this cycle (34344, correct case TFTTF). After entering a stable state, each prediction is (34344, which is always predicted to be taken) and the accuracy rate is 0.6.

(3) We're going to use a time-synchronized shift register, which will have 5 digits of 10100, shifted by one digit to the right each time, and it will make perfect predictions because the pattern of branches is the same and repeats all the time