Computer Architecture

**Fall, 2022**

**Week 12**

**2022.11.28**

**組別：＿＿＿＿＿　簽名：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿**

[group4]

Which of the following statements is (are) true for the forwarding unit used in a five-stage pipelined processor?

1. The forwarding unit is used to bypass the write-back result due to RAW hazards.



1. The forwarding unit is used to forward data to the register file.



1. The forwarding unit compares the source register number of the instructions in the MEM and WB stages with the destination register number of the instruction in EX stage.



1. The forwarding unit can be used to resolve load-use data hazard directly.



[group6]

True or False

1. In MIPs 5-stage pipeline, R-Type-use data hazard can be handled by Forwarding technique.



1. In MIPs 5-stage pipeline, forwarding technique is not enough to handle load-use data hazard since it takes longer time to read memory than register.



1. The main reason why inserting NOP is not preferred is that software runs slower than hardware.



1. Using Forwarding technique, we always need to forward data if EX/MEM.RegisterRD == ID/EX.RegisterRS.



1. There is a register which will be written by instruction 1 and then read by instruction 2. Without forwarding, 3 stalls or NOPs will be inserted in instruction 2 to avoid hazard and to make sure that the ins2. will read the right data from the pipeline register.



1. The main difference between inserting NOP and Stalls is that the previous one is controlled by the compiler and the latter one is controlled by hardware.



[group9]

In MIPS pipeline, which of the following program segments will require a NOP bubble (require stalling)? Please explain why or why not?

Hint:

If (ID/EX.MemRead and

((ID/EX.RegisterRt = IF/ID.RegisterRs) or

(ID/EX.RegisterRt = IF/ID.RegisterRt))):

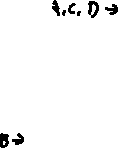
stall the pipeline



(A)



lw $1, 60($2)



add $2, $1, $3



and $1, $2, $3



(B)



add $1, $2, $3



add $1, $1, $2

add $1, $1, $3

(C)



lw $2, 60($1)



add $2, $1, $3

and $1, $2, $3



(D)



lw $0, 60($2)

add $1, $0, $2

sub $2, $0, $1

[group10]

Consider the following MIPS code. Please find ALL the hazards in there and rearrange it to make it right.



lw $t1, 4($t0)



add $t3, $t1, $t2



sub $t6, $t7, $t8



lw $t4, 8($t0)



add $t5, $t1, $t4



and $t8, $t6, $t6

[group2]

Fred set a list of MIPS instructions.

(1) add $s0, $s1, $s2

(2) and $s3, $s0, $s1

(3) lw $s4, 12($s3)

(4) addi $s5, $s4, 2

Please help him answer the following question:

(a) List all the data dependencies between the above instructions. (Ex: (1)、(2) -> $s0)



(b) What kind of hazard may occur in the list of instructions?



(c) Please insert NOP between instructions to solve the hazard. (Assume the hardware Fred is using has no forwarding design, but it has internal forwarding in the register file.)



[group1]

Please select the correct options and explain the incorrect ones.

* 1. We can always resolve hazards by waiting.



* 1. The hazard, which tries to make a decision before evaluating a condition, always occurs in branch instructions.



* 1. In MIPS, if there is data dependency, there must be a data hazard.



* 1. By using more than one memory we can solve the data hazard.



* 1. The forwarding logic requires extra units to achieve.



[group5]

Which of the following statements are correct?

1. We don’t care “write after read” and “write after read” hazards in MIPS.



1. At the MEM stage, if EX/MEM.RegRd = ID/EX.RegRs, it always forwards.



1. If both WB and MEM can forward , we let MEM forward.



1. We don’t have to forward if the destination register is $0.



1. When stalling, PC still updates.



1. the insertions of Nops can speed up the pipeline

