Habib University

EE-371/CS-330/CE-321 Computer Architecture – Spring 2024
Instructors: Dr. Tariq Kamal, Dr. Farhan Khan, Dr. Waseem Hassan

Time = 50 minutes **Quiz 03 SOL** **Max Points: 20**

Instructions:

i. ***Smart watches, laptops, and similar electronics are strictly NOT allowed.***
ii. ***Answer sheets should contain all steps, working, explanations, and assumptions.***
iii. Attempt the quiz on clean papers with black/blue ink.
iv. Print your name and HU ID on all sheets.
v. Turn in your question paper along with your answer sheets.
vi. You are not allowed to ask/share your method or answer with your peers. The work submitted by you is solely your own work. Any violation of this will be the violation of HU Honor code and proper action will be taken as per university policy if found to be involved in such an activity.

CLO Assessment:

This assignment assesses students for the following course learning outcomes.

Course Learning Outcomes		CLO Assessed
CLO 1	***Explain*** the role of ISA in modern processors and instruction encodings and assembly language programming	
CLO 2	***Explain*** the architecture and working of a single cycle processor	
CLO 3	***Design*** the architecture to mitigate issues of a pipelined processor	✓
CLO 4	***Analyze the*** performance of cache operations	

Question 1 [6 points]:

Consider the RISC-V assembly code given below:

```
ld  x3, 32(x10)
ld  x1, 8(x10)
add x2, x4, x3
sub x5, x2, x3
bne x2, x0, Ll
sub x3, x3, x4
```

Suppose we modify the pipeline so that it has only one memory that handles both instructions and data. In this case, there will be a structural hazard every time a program needs to fetch an instruction during the same cycle in which another instruction accesses data.

a) [3 points] Draw a pipeline diagram to show where the code above will stall.

b) [3 points] Is it possible to reduce the number of stalls resulting from this structural hazard by reordering code?

Solution:

a) The lines of code have been numbered and this numbering will be used in the multiple-clock-cycle pipeline diagram:

```
1: ld  x3, 32(x10)
2: ld  x1, 8(x10)
3: add x2, x4, x3
4: sub x5, x2, x3
5: bne x2, x0, Ll
6: sub x3, x3, x4
```

Instr. No.	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7	Cycle 8	Cycle 9	Cycle 10	Cycle 11	Cycle 12
1	IF	ID	EX	MEM	WB							
2		IF	ID	EX	MEM	WB						
3			IF	ID	EX	MEM!	WB					
-				Stall	Stall	Stall	Stall	Stall				
-					Stall	Stall	Stall	Stall	Stall			
4						IF	ID	EX	MEM!	WB		
5							IF	ID	EX	MEM!	WB	
6								IF	ID	EX	MEM!	WB

b) In any case, there will be a structural hazard every time a program needs to fetch an instruction during the same cycle in which another instruction accesses data. Reordering the code will simply change the instructions in conflict. Number of stalls won't be reduced since every instruction has to be fetched. This means that data access for ever instruction results in a stall.

Question 2 [4 points]:

A notation that names the fields of the pipeline registers allows for a more precise notation of dependences. For example, "ID/EX.RegisterRs1" refers to the number of one register whose value is found in the pipeline register ID/EX; that is, the one from the first read port of the register file. The first part of the name, to the left of the period, is the name of the pipeline register; the second part is the name of the field in that register.

Using this notation, the two pairs of hazard conditions are:

- Type 1a. EX/MEM.RegisterRd = ID/EX.RegisterRs1
- Type 1b. EX/MEM.RegisterRd = ID/EX.RegisterRs2
- Type 2a. MEM/WB.RegisterRd = ID/EX.RegisterRs1
- Type 2b. MEM/WB.RegisterRd = ID/EX.RegisterRs2

Classify the **data hazards** in the following RISC-V assembly code using the above types:

```
add x22, x11, x31
or x12, x22, x5
and x13, x16, x22
sub x14, x22, x22
sd x15, 100(x22)
```

Solution:
The data hazards are as follows:

■ The add–or is a type 1a hazard.
■ The add–and is a type 2b hazard:
MEM/WB.RegisterRd = ID/EX.RegisterRs2 = x22
■ The two dependences on add–sub are not hazards because the register file supplies the proper data during the ID stage of sub.
■ There is no data hazard between add and sd because sd reads x22 the clock cycle after add writes x22.

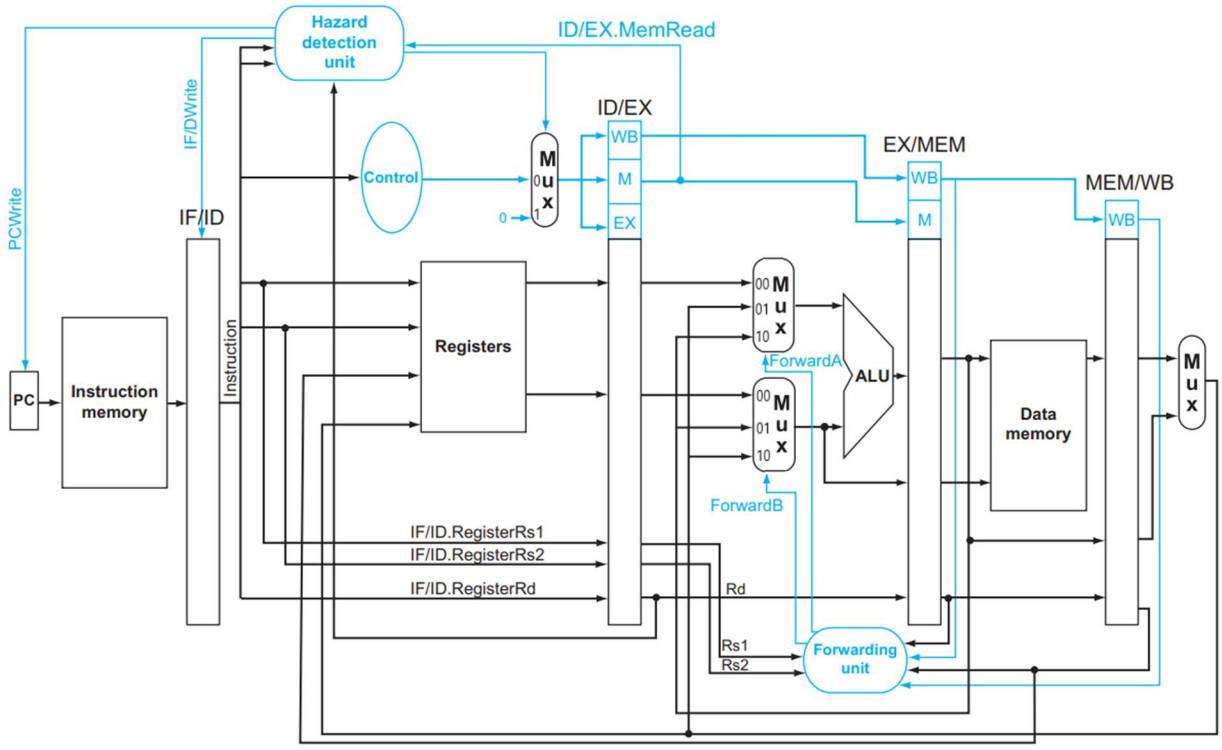
Question 3 [6 points]:

Assume that the following sequence of RISC-V assembly instructions is executed on a five-stage pipelined RISC-V processor:

```
or x18, x19, x20
ld x31, 88(x18)
ld x30, 80(x8)
add x31, x31, x1
sd x31, 72(x18)
```

a) [3 points] If there is no forwarding or hazard detection, insert NOPs to ensure correct execution.

b) [3 points] If the processor has the forwarding unit implemented, but not the hazard detection unit, what happens when the original code executes on the processor?



Solution:

a) The lines of code have been interspersed with nops.

```
or x18, x19, x20
nop
nop
ld x31, 88(x18)
ld x30, 80(x8)
nop
```

```
add x31, x31, x1
nop
nop
sd x31, 72(x18)
```

b) A hazard detection unit is needed when a load-use data hazard occurs. All the other hazards can be catered to by the forwarding unit. Since there is no load-use data hazard, i.e., for two consecutive load instructions, the second load instruction does not use the destination register of the first one, absence of hazard detection does not disrupt the execution of the given code sequence.

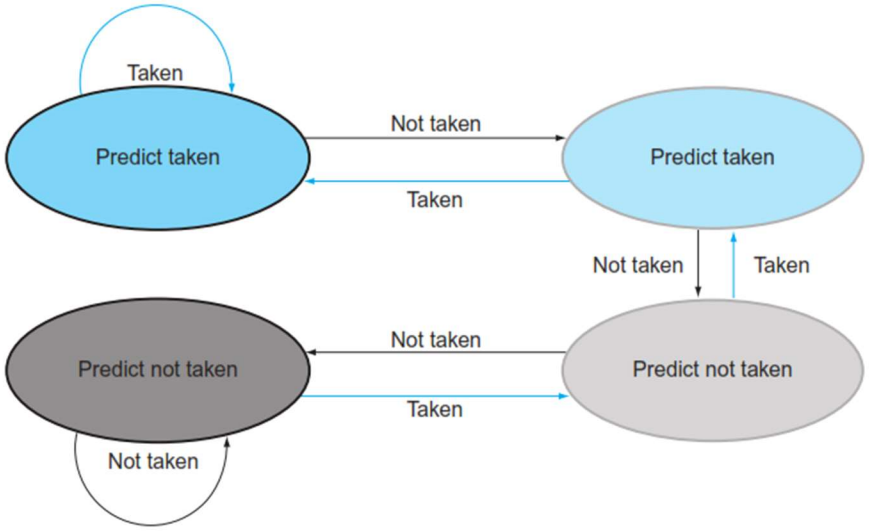
Question 4 [4 points]:

Consider the following repeating pattern of branch outcomes (e.g., in a loop):

T, NT, T, T, NT

a) [2 points] What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes?

b) [2 points] What is the accuracy of the 2-bit predictor for the first four branches in this pattern, assuming that the predictor starts off in the state "predict not taken"?



Solution:

a) The accuracy of always-taken predictor can be calculated from the following:

PREDICTION	T	T	T	T	T
ACTUAL	T	NT	T	T	NT
OUTCOME	Correct	Incorrect	Correct	Correct	Incorrect

Accuracy (T) = 3/5 = 0.6

The accuracy of always-not-taken predictor can be calculated from the following:

PREDICTION	NT	NT	NT	NT	NT
ACTUAL	T	NT	T	T	NT
OUTCOME	Incorrect	Correct	Incorrect	Incorrect	Correct

Accuracy (NT) = 2/5 = 0.4

b)

PREDICTION	NT	NT	NT	NT
ACTUAL	T	NT	T	T
OUTCOME	Incorrect	Correct	Incorrect	Incorrect

Accuracy (DP) = 1/5 = 0.25