Amrita Vishwa Vidyapeetham

Amrita School of Engineering, Coimbatore

Department of Computer Science and Engineering

Topic: Pipelining-Hazards

Sub Code: 19CS211 **Sub Title: COA**

Roll No: CB.EN.U4CSE19453 **Name: R.ABHINAV**

Lab Evaluation No: Term2-A1 **Date: 05-03-2021**

SNo	Code Snippets	Situation	Action to be taken
1	**LW $s0, 45($s1)** **ADD $s2, $s3, $s4** **SUB $s5, $s3, $s4** **OR $s6, $s3, $s4**	no data hazard, as the instructions are independent of the other instructions	no action to be taken

2	**LW $s0, 45($s1)**
	ADD $s2, $s0, $s4
	SUB $s5, $s3, $s4
	OR $s6, $s3, Ss4
		Data Hazard, because ADD requires &s0 before write back of LW
		accessing s0 in the next instruction will not give the updated value. hence a buble must be included(time consuming) or reordering will get rid of the data hazzard
			LW $s0, 45($s1)
			SUB $s5,$s3,$s4
			ADD $s2,$s0,$s4
			OR $s6,$s3,$s4

3	**LW $s0, 45($s1)** **ADD Ss2, $s3, $s4** **SUB $s5, $s0, $s4** **OR $s6, $s3, $s4**	s0 will be updated only in the 4th cycle by the time SUB istruction will be in 2nd stage. but the updated s) will be avilable only in the 5th stage	forwarding from MEM to ID could be used to avoid stalling.
4	**LW $s0, 45($s1)** **ADD $s2, $s3, $s4** **SUB $s5, $s3, $s4** **OR $s6, $s0, $s4**	during instruction fetch of OR s0 will not be updated hence data hazard is caused.	introducing a bubble in before OR operation will make s0 available.

5. Explain about the situation where forwarding is not only sufficient to overcome the hazard with typical code snippets.

A. Forwarding (in a pipeline processor) –

- Pipeline forwarding involves routine of data (called packet or packets) from one pipeline stage to another. In particular, PF is optimal from various points of view:

1. High efficiency in utilization of network resources, which enables accommodating a larger amount of traffic on the network, thus lowering operation cost and being the foundation for accommodating the exponential growth of modern networks.
2. Low implementation complexity, which enables the realization of larger and more powerful networking systems at low cost, thus offering further support to network growth.
3. High scalability, which is an immediate consequence of the above two features.

B. Data Hazard (in a pipeline processor)

Data hazards occur when instructions that exhibit data dependence modify data in different stages of a pipeline.

- If the result can be moved from where the ADD produces it (EX/MEM register), to where the SUB needs it (ALU input latch), then the need for a stall can be avoided.
-

Using this observation , forwarding works as follows:

- The ALU result from the EX/MEM register is always *fed back* to the ALU input latches.
-

If the forwarding hardware detects that the previous ALU operation has written the register corresponding to the source for the current ALU operation, *control logic* selects the forwarded result as the ALU input rather than the value read from the register file.

a) The paths correspond to a forwarding of:
(a) the ALU output at the end of EX,
(b) the ALU output at the end of MEM, and
(c) the memory output at the end of MEM.

5) lw r_1, 0(r_2)
sub r_4, r_1, r_6
and r_6, r_1, r_7
or r_8, r_1, r_9

IF | ID | EX | MEM | WB
IF | ID | EX | MEM | WB
IF | ID | EX | MEM | WB
IF | ID | EX | MEM | WB

- from above code; we can see that the data is not loaded until after MEM stage to the next instruction cycle.
  ∴ We have to stall & insert bubble to overcome this data hazard.

- IF | ID | EX | MEM | WB
  IF | ID | stall | EX | MEM | WB
  IF | stall | ID | EX | MEM | WB
  stall | IF | ID | EX | MEM | WB