CDA5155/CDA4150 - Spring 2026
Assignment 3
Detecting Pipeline Hazards

Objectives: Learn how both single cycle and multicycle instructions proceed through a pipeline in a single-issue, in-order processor.

Below is a sequence of RISC-V instructions. The pipeline diagram shows when each instruction will be in each stage of the pipeline. In this diagram, each row represents a cycle and each column represents a different pipeline stage. All instructions go through the IF and ID stages. Integer instructions go through the EX, MEM, and WB stages. Floating-point load instructions go through EX, MEM, and FWB stages. Floating-point add and subtract operations go through the FADD and FWB stages, do not go through MEM stage, and require 4 cycles during execution. Store instructions do not perform a write-back stage.

```
1. fadd.s  f4,f0,f2
2. lw      x3,0(x1)
3. add     x4,x4,x3
4. sw      x4,0(x1)
5. addi    x1,x1,-8
```

cycle	IF	ID	EX	MEM	WB	FADD	FWB
1	1						
2	2	1					
3	3	2				1	
4	4	3	2			1	
5	stall	stall		2		1	
6	5	4	3		2	1	
7		5	4	3			1
8			5	4	3		
9				5			
10					5		

Below is another sequence of instructions. Fill in the pipeline diagram using the same assumptions described for the previous pipeline example. Also assume traditional forwarding where it is possible. Explain the reason for each stall, if any occur. You should upload your solution for this assignment in Canvas before the beginning of class on March 5.

```
1. flw     f0,0(x1)
2. fadd.s  f4,f0,f2
3. sub     x5,x5,x1
4. flw     f6,8(x1)
5. fsw     f4,0(x1)
6. fsw     f6,16(x1)
7. addi    x1,x1,-8
```

cycle	IF	ID	EX	MEM	WB	FADD	FWB
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							