EECS 470 Fall '25
Homework 2

Due Wednesday, September 17th by 11:59 pm on Gradescope.com.

Name: _____ uniqname: _____

Upload a PDF of your legible answers to the course Gradescope page. This is an individual assignment; all work should be your own.

- Please state any assumptions.

ILP scheduling. (based on code taken from H&P).
Consider the following code sequence (destination register specified last):

Functional units & latencies:	
1 Load	2
1 Store	1
1 Integer ALU	1
1 Branch	1
1 FP Adder	3
1 FP Multiplier	5
1 FP Divider	10

```
Loop: LD 0(R1),F2
I0:   ADDD F0,F2,F4
I1:   DIVD F2,F0,F6
I2:   LD 8(R1),F2
I3:   MULTD F2,F6,F10
I4:   ADDD F2,F2,F2
I5:   ADDD F2,F4,F2
I6:   SD 8(R1),F10
I7:   ADDI R1,#16,R1
I8:   SUB R4,R1,R20
I9:   BNZ R20,Loop
```

1. Determine how long the loop takes to complete on an in-order machine where an instruction may not issue (start execution) until the preceding instruction is writing back its result, whether they are dependent or not. (Ignore instruction fetch). Create a table showing when each instruction finishes decode, execute, and writeback. Report the total number of cycles between decoding "Loop" of one iteration until the decoding of "Loop" in the second. We have filled in the first few rows of the table below: [4]

Instruction		Decode (D)	Execute (X)	Writeback (W)
Loop: LD 0 (R 1) , F2	(2)	1	3	4
I0 : ADDD F 0 , F 2 , F 4	(3)	4	7	8

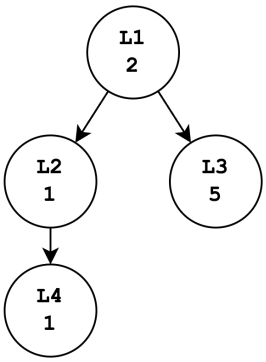
2. Now, consider the execution of this code on a machine with scoreboard scheduling. Create a table showing when each instruction completes the dispatch, issue, execute, and writeback stages, like the table shown in lecture, for a single iteration. Each time there is a stall, indicate the reason for the stall (structural hazard on FU, WAW, WAR, or RAW dependency on a register). Assume that instructions after a branch can be dispatched when the branch reaches the writeback stage. Report the number of cycles it takes in between dispatching "Loop" across successive iterations. Here are the first three rows of the table after 5 cycles. For this problem, assume that it is a scalar machine, meaning no stage can have more than q instruction in a given cycle. [18]

Instruction		Dispatch (D)	Issue (S)	Execute (X)	Writeback (W)
Loop: LD 0 (R 1) , F2	(2)	1	2	4	5
I0 : ADDD F 0 , F2 , F4	(3)	2	5 (RAW F2)		
I1 : DIVD F 2 ,F 0 ,F 6	(10)	3	6 (RAW F2)		

3. Now, draw a *dependence graph*, where the vertices correspond to instructions and the edges correspond to true control and data dependences among the instructions. (Because this graph shows the data flow of an instruction sequence, it is sometimes called a *dataflow graph*). Annotate each instruction with its latency. An example of a dependence graph for a different program is below. [5]

```
L1: r1 = lw 0(r1)
L2: r3 = r1 + r2
L3: r4 = r1 * r4
L4: r5 = r3 + r1

ld: 2 cycles
 *: 5 cycles
 +: 1 cycle
```



4. Now, using your answer from Question 3, imagine an ideal out-of-order processor with infinite issue width, an unbounded number of functional units and an ideal register renaming implementation (i.e., the renaming hardware eliminates all false dependencies). What is the latency to complete the loop iteration (only consider the latencies listed in Question 3)? [3]