Name: Joseph Angelo

HW 2: Pipeline Tracing

1) Given the snippet of assembly code below on a LOAD-STORE ISA, for each line of code, identify instructions (by line number) for which there are any possible data dependencies that have any potential to result in a data hazard... i.e., for line X, identify all lines that have to have already written their output before X can move into the Operand Fetch stage. Remember that conditional branches are based on what's in the flag registers (tricky, tricky). If a line isn't dependent on anything we know of, just write "none". As a starting point, I've already written the answer for lines 3 and 4.

Line no.	Code	Data Dependencies
1	set r3, 0	none
	!top	
2	lod r2, [ABC]	none
3	add r3, r2	1, 2
4	mpy r3, r1	3, 7
5	cmp r2, 5	2
6	jl !middle	5
7	lod r1, [DEF]	none
8	sub r3, r2	4,5
9	str [ABC], r3	8
10	jmp !top	none
	!middle	
11	cmp r3, r1	7,8
12	jg !bottom	11
13	str r3, [DEF]	
14	cmp r1, 0	11
15	je !middle	14
16	add r2, r3	2,13
	!bottom	

1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8

= cycles + (stages - 1)

Name: ______________________________

2) Assuming each instruction on a specific CPU must complete **7 stages** from Fetch through Write Back that are one cycle each, (A) how many cycles would it take to complete 8 instructions **without** a pipeline, and (B) how many cycles would it take to complete the same 8 instructions **with an idealized pipeline (no hazards)** of those same 8 stages?

Ⓐ 8 × 7 = 56 Ⓑ 8 + 6 = 14

3) Below is a short bit of assembly code with all data dependencies identified. There is a 5-stage pipeline (Fetch, Decode, Operand Fetch, Execute, and Write Back). Complete the first 20 cycles of the pipeline trace (on the next page).

- There will be NO structural hazards.
- We'll use register forwarding such that an instruction (X) waiting on a data hazard from instruction (Y) can enter the OF stage when Y begins its WB. The same goes for conditional jump instructions waiting on data hazards of flags from CMP instructions. The conditional jump can enter the OF stage when the CMP begins its WB.
- Conditional jumps will flush the pipeline and replace the PC during the E stage… the new PC will begin to be Fetched when the conditional jump begins the WB stage.
- Unconditional jumps will be detected in the D stage, flushing the pipeline and replacing the PC at that time. The new PC will begin to be Fetched when the unconditional jump begins the OF stage.

Line no.	Code	Data Dependencies
1	`set rA, 2`	none
2	`set rB, 1`	none
	`!top`	
3	`sub rA, rB`	1, 2, 6
4	`cmp rA, 0`	3
5	`jle !done`	4
6	`add rB, 1`	2
7	`jmp !top`	none
	`!done`	

Name: ____________________

F	1	2	3	4	4	5	5	6	6	7
D	—	1	2	3	3	4	4	5	5	6
OF	—	—	1	2	—	3	—	4	—	5
E	—	—	—	1	2	—	3	—	4	—
WB	—	—	—	—	1	2	—	3	—	4
Time	1	2	3	4	5	6	7	8	9	10

F	—	6	7	—	3	4	5	5	6	6
D	—	—	6	7	—	3	4	4	5	5
OF	—	—	—	6	7	—	3	—	4	—
E	5	—	—	—	6	7	—	3	—	4
WB	—	5	—	—	—	6	7	—	3	—
Time	11	12	13	14	15	16	17	18	19	20

A = ~~2~~ 1
B = 1