Computer Architecture Homework Assignment #1

TA in charge: Myeongjae Jang
E-mail: myeongjae0409@kaist.ac.kr
Office: N1 - #922

I. Submission and grading

- ✓ Due date: March, 26th (Tue.) 23:59:59. There was a typo in the lecture note. Please check the due date again.
- ✓ Submit your homework as a hardcopy into HW box prepared near the office room #922 in N1 building. Do not submit by KLMS or TA's e-mail.
- ✓ Late submissions will not be accepted. Please keep the submission due date.
- ✓ Each score has been written at the end of each question. The total score is 100.
- ✓ You will be given 0 point for any kind of cheating.
- ✓ Please explicitly denote number of each question and its answer. Otherwise, it can be considered as not be done.
- ✓ Please give detailed process how you solve questions. Otherwise, no point will be charged.

II. Questions

✓ Please solve every questions as below **(Total 100 points)**:

Q1. Use the following code fragment:

```
Loop:   LD      R1, 0(R2)       ;load R1 from address 0+R2
        DADDI   R1, R1, #1      ;R1=R1+1
        SD      R1,0,(R2)       ;store R1 at address 0+R2
        DADDI   R2,R2,#4        ;R2=R2+4
        DSUB    R4,R3,R2        ;R4=R3-R2
        BNEZ    R4,Loop         ;branch to Loop if R4!=0
```

Assume that the initial value of R3 is R2+396.
(For *Figure C.5* and *Figure C.6*, you can refer to Appendix page C-15 and C-17 in the textbook. *Figure C.5* and *Figure C.6* are given in the next page.)

A. Data hazards are caused by data dependences in the code. Whether a dependency causes a hazard depends on the machine implementation (i.e., number of pipeline stages). List all of the data dependences in the code above. Record the register, source instruction, and destination instruction; for example, there is a data dependency for register R1 from the LD to the DADDI. **(8 points)**

B. Show the timing of this instruction sequence for the 5-stage RISC pipeline without any forwarding or bypassing hardware but assuming that a register read and a write in the same clock cycle "forwards" through the register file, as shown in *Figure C.6*. Use a pipeline timing chart like that in *Figure C.5*. Assume that the branch is handled by flushing the pipeline. If all memory references take 1 cycle, how many cycles does this loop take to execute? **(8 points)**

C. Show the timing of this instruction sequence for the 5-stage RISC pipeline with full forwarding and bypassing hardware. Use a pipeline timing chart like that shown in *Figure C.5*. Assume that the branch is handled by predicting it as not taken. If all memory references take 1 cycle, how many cycles does this loop take to execute? **(10 points)**

D. Show the timing of this instruction sequence for the 5-stage RISC pipeline with full forwarding and bypassing hardware. Use a pipeline timing chart like that shown in *Figure C.5*. Assume that the branch is handled by predicting it as taken. If all memory references take 1 cycle, how many cycles does this loop take to execute? **(10 points)**

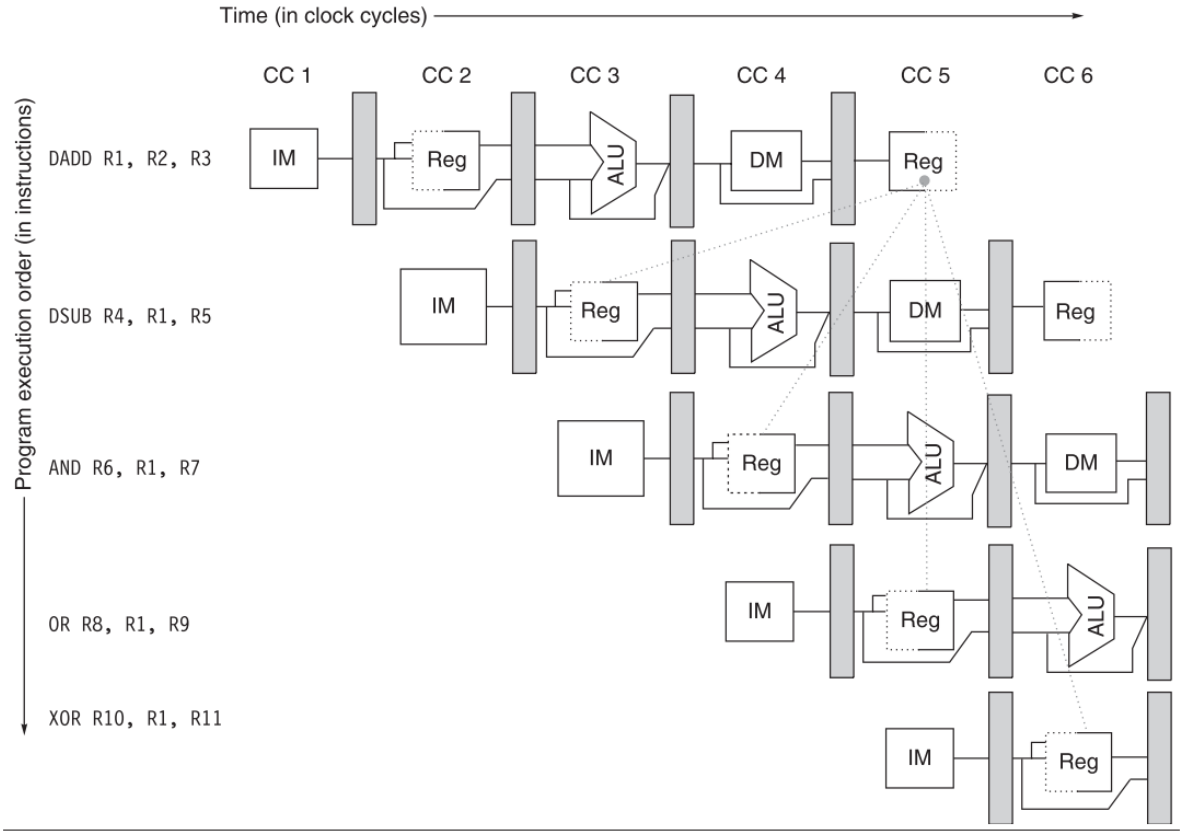

Figure C.6 The use of the result of the DADD instruction in the next three instructions causes a hazard, since the register is not written until after those instructions read it.

	Clock cycle number									
Instruction	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Load instruction	IF	ID	EX	MEM	WB					
Instruction $i + 1$		IF	ID	EX	MEM	WB				
Instruction $i + 2$			IF	ID	EX	MEM	WB			
Instruction $i + 3$				Stall	IF	ID	EX	MEM	WB	
Instruction $i + 4$						IF	ID	EX	MEM	WB
Instruction $i + 5$							IF	ID	EX	MEM
Instruction $i + 6$								IF	ID	EX

Figure C.5 A pipeline stalled for a structural hazard—a load with one memory port. As shown here, the load instruction effectively steals an instruction-fetch cycle, causing the pipeline to stall—no instruction is initiated on clock cycle 4 (which normally would initiate instruction $i + 3$). Because the instruction being fetched is stalled, all other instructions in the pipeline before the stalled instruction can proceed normally. The stall cycle will continue to pass through the pipeline, so that no instruction completes on clock cycle 8. Sometimes these pipeline diagrams are drawn with the stall occupying an entire horizontal row and instruction 3 being moved to the next row; in either case, the effect is the same, since instruction $i + 3$ does not begin execution until cycle 5. We use the form above, since it takes less space in the figure. Note that this figure assumes that instructions $i + 1$ and $i + 2$ are not memory references.

Q2. Suppose the branch frequencies (as percentages of all instructions) are as follows:

Conditional branches	15%
Jumps and calls	1%
Taken conditional branches	60% are taken

A. We are examining a four-deep pipeline where the branch is resolved at the end of the second cycle for unconditional branches and at the end of the third cycle for conditional branches. Assuming that only the first pipe stage can always be done independent of whether the branch goes and ignoring other pipeline stalls, how much faster would the machine be without any branch hazards? **(15 points)**

B. Now assume a high-performance processor in which we have a 15-deep pipeline where the branch is resolved at the end of the fifth cycle for unconditional branches and at the end of the tenth cycle for conditional branches. Assuming that only the first pipe stage can always be done independent of whether the branch goes and ignoring other pipeline stalls, how much faster would the machine be without any branch hazards? **(15 points)**

Q3. We begin with a computer implemented in single-cycle implementation. When the stages are split by functionality, the stages do not require exactly the same amount of time. The original machine had a clock cycle time of 7 ns. After the stages were split, the measured times were IF, 1 ns; ID, 1.5 ns; EX, 1 ns; MEM, 2 ns; and WB, 1.5 ns. The pipeline register delay is 0.1 ns.

A. What is the clock cycle time of the 5-stage pipelined machine? **(7 points)**

B. If there is a stall every 4 instructions, what is the CPI of the new machine? **(7 points)**

C. What is the speedup of the pipelined machine over the single-cycle machine? **(10 points)**

D. If the pipelined machine had an infinite number of stages, what would its speedup be over the single-cycle machine? **(10 points)**