Homework #4

Course: CSE4010-01 Computer Architecture

Professor Juho Kim

May 12, 2021

1. Submission

- Submission Deadline: **May 18, 2021, 11:59 pm (Submit at Cyber Campus)** (20% deduction per day. No credit after 3days)
- Write solving processes and answers on a blank white paper, scan it and submit it as a **PDF file.**
- The filename should be **HW4_STUDENT-ID_NAME.pdf**

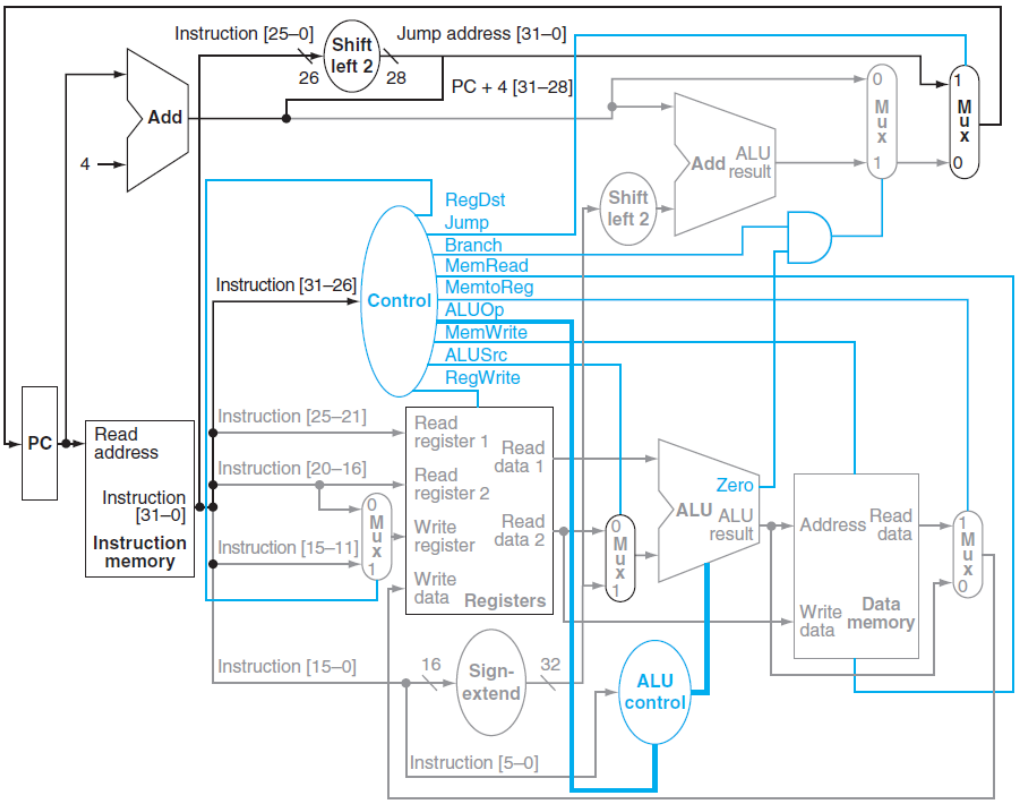
  (ex. HW4_20219999_홍길동.pdf)
- **WARNING**:
  - Students who copy other's homework will get zero point for this assignment.
  - Submission without any solving processes will get zero points.
  - Submission with another form rather than pdf will have a 3 points reduction of the total score.
  - Submission with the wrong file name will have 3 points reduction of total score.
  - Before submission, please check whether your solving processes and answers are clear enough to read. (No credit will be given if your work is illegible)
  - All works must be hand-written.(50% reduction for typed submissions)

2. Reference

- Lecture notes
- Patterson and Hennessy, Computer Organization and Design 4th (ARM Edition), Morgan Kaufmann, 2010

1. Assume that the logic blocks used to implement the datapath have the following latencies. **(10pts)**

I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-extend	Shift-left-2	ALU Ctrl
400ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps	55ps



To avoid lengthening the critical path of the datapath shown above, how much time can the control unit take to generate the MemWrite signal?

2. Assume that individual stages of the datapath have to following latencies.

IF	ID	EX	MEM	WB
250ps	170ps	150ps	240ps	190ps

(20pts)

2-1 What is the clock cycle time in a pipelined and nonpipelined processor?.

2-2 What is the total latency of a lw instruction in a pipelined and nonpipelined processor?

2-3 If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?

ALU	beq	lw	sw
30%	25%	25%	20%

2-4 Assuming there are no stalls or hazards, what is the utilization(% of cycles used) of data memory?

2-5 Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?

3. (15pts)

Instruction sequence

lw $5, –16($5)
sw $4, –16($4)
lw $3, –20($4)
beq $2, $0, Label ; Assume $2 != $0
add $5, $1, $4

For this problem, assume that all branches are perfectly predicted (this eliminates all control hazards) and that no delay slots are used. If we only have one memory (for both instructions and data), there is a structural hazard every time we need to fetch an instruction in the same cycle in which another instruction accesses data. To guarantee forward progress, this hazard must always be resolved in favor of the instruction that accesses data. What is the total execution time of this instruction sequence in the five-stage pipeline that only has one memory?

4. (20pts)

Instruction sequence

add $1,$2,$3
sw $2,0($1)
lw $1,4($2)
add $2,$2,$1

4-1 Find all data dependences in this instruction sequence

4-2 Find all hazards in this instruction sequence for a five-stage pipeline with and then without forwarding

5.

Instruction sequence

add $1,$5,$3
sw $1,0($2)
lw $1,4($2)
add $5,$5,$1
sw $1,0($2)

(5pts)

If there is no forwarding or hazard detection, insert nops to ensure correct execution.

6. Assume that the following MIPS code is executed on a pipelined processor with a five-stage pipeline, full forwarding, and a predict-taken branch predictor **(30pts).**

```
        add $1, $5, $3
Label1: sw $1, 0($2)
        add $2, $2, $3
        beq $2, $4, Label1    ; Not taken
        add $5, $5, $1
        sw $1, 0($2)
```

6-1 Draw the pipeline execution diagram for this code, assuming there are no delay slots and that branches execute in the EX stage

1	2	3	4	5	6	7	8	9	10	11	12	13	14

6-2 Repeat 6-1, but assume that delay slots are used. In the given code, the instruction that follows the branch is now the delay slot instruction for that branch.

1	2	3	4	5	6	7	8	9	10	11	12	13	14

7. Consider the instruction block given below where $t1, $t2, and $t4 all points to different addresses**(20pts).**

```
addi    $t1,    $t10,    -10
lw      $t2,    0($t1)
lw      $t3,    0($t2)
sw      $t3,    0($t4)
sub     $t3,     $zero,   $t3
addi    $t3,    $t3,     1
```

where *$zero* is a register containing a value of zero always and *addi* is adding one register and one immediate operand.

7-1 Identify each of the dependencies in the code if executed on a pipelined processor by circling them. You should draw a circle around both registers that exhibit the dependency and connect them with a line.

7-2 Using the same code block as in part 7-1, show each of the stages of execution through a pipelined processor that provides forwarding. Show all required forwarding by drawing an arrow from the source to the destination. Indicate pipeline stalls by a dashed line(--).
(Hint: a stall cycle is inserted after an instruction is decoded in ID/RF stage and dependency is detected as a result of instruction decoding. All instruction following a stalled instruction are also stalled. E.g. if an instruction A is stalled by one clock cycle, the pipelined diagram is shown as:

Inst A IF | ID/RF | -- | EX | MEM | WB |
Inst B | IF | -- | ID/RF | EX | MEM | WB)

1	2	3	4	5	6	7	8	9	10	11	12	13	14