CENG3420 Homework 3

Due: 11:59 PM, Mar. 21, 2023

All solutions should be submitted to the blackboard in the format of **PDF/MS Word**.

Q1 (25%) In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:

IF	**ID**	**EX**	**MEM**	**WB**
250ps	400ps	200ps	300ps	200ps

1. What is the clock cycle time in a pipelined and non-pipelined (single-cycle) processor?
2. What is the total latency of an lw instruction in a pipelined and non-pipelined (single-cycle) processor?
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?

Q2 (15%) Consider the following sequence of instructions, and assume that it is executed on a five-stage pipelined datapath.

```
lw x12, 4(x11)
lw x14, 4(x13)
and x15, x12, x14
sub x13, x11, x13
```

1. If there is no forwarding or hazard detection, insert NOPs to ensure correct execution. You need to draw a pipeline diagram like we did in the lectures.
2. Does inserting NOPs change the clock cycle time?
3. Does inserting NOPs change the excute time of a program containing this block of code.

Q3 (20%)

Consider the following sequence of instructions, and assume that it is executed on a five-stage pipelined datapath:

```
xor x13, x12, x11
lw x14, 8(x13)
and x10, x15, x14
```

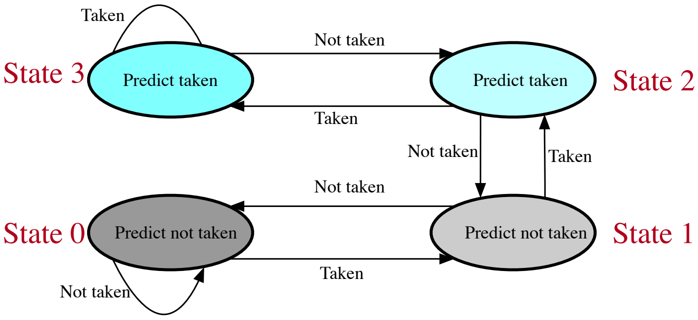
1. If there is no forwarding or hazard detection, insert NOPs to ensure correct execution. You need to draw a pipeline diagram like we did in the lectures.

2. If hazard detection and forwarding are allowed, draw a pipeline diagram with forwarding to ensure correct execution. You may also use NOP(s), but you need to use at least NOPs as possible.

Q4 (20%)

This exercise examines the accuracy of various branch predictors for the following repeating pattern (e.g., in a loop) of branch outcomes: T, NT, NT, NT, T. (T means ’Taken’ and NT means ’Not taken’)

1. What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes?
2. What is the accuracy of a 2-bit predictor if this pattern is repeated forever? You should give your explanations rather than just an answer. The following figure shows the finite-state machine for a 2-bit prediction scheme. Here we assume this predictor starts from ”State 0”.



Q5 (20%) Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (classes A, B, C, and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 2 GHz and CPIs of 2, 2, 2, and 2. A program with an instruction count of 1.0E6 can be divided into classes as follows: 10% class A, 20% class B, 40% class C, and 30% class D.

1. What is the average CPI for each implementation?
2. What is the CPU execution time of this program for each implementation?

Q1 (25%) In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:

IF	ID	EX	MEM	WB
250ps	400ps	200ps	300ps	200ps

1. What is the clock cycle time in a pipelined and non-pipelined (single-cycle) processor?
2. What is the total latency of an lw instruction in a pipelined and non-pipelined (single-cycle) processor?
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?

1) pipelined = 400ps x 5 = 2000ps //

non-pipelined = 250+400+200+300+200 = 1350ps //

2) In an lw instruction

pipelined = 2000 ps //

non-pipelined = 1350 ps //

3) ID, 400ps to 200ps X 2

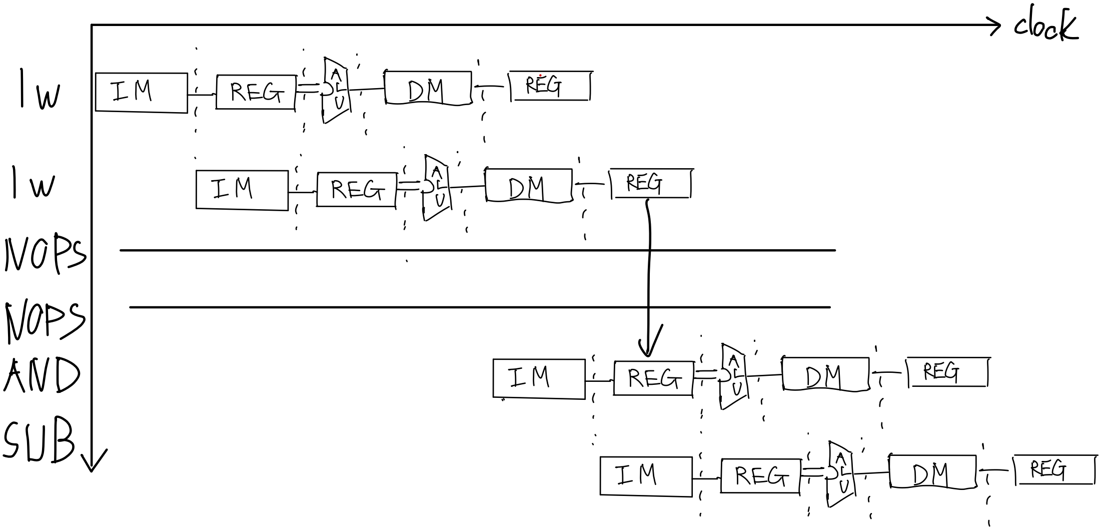
max Latency will be MEM = 300ps

new pipelined Latency = 300 X 6 = 1800 ps //

Q2 (15%) Consider the following sequence of instructions, and assume that it is executed on a five-stage pipelined datapath.

```
lw x12, 4(x11)
lw x14, 4(x13)
and x15, x12, x14
sub x13, x11, x13
```

1. If there is no forwarding or hazard detection, insert NOPs to ensure correct execution. You need to draw a pipeline diagram like we did in the lectures.
2. Does inserting NOPs change the clock cycle time?
3. Does inserting NOPs change the excute time of a program containing this block of code.



2) NO

3) YES, inserting will increase the excute time of the program

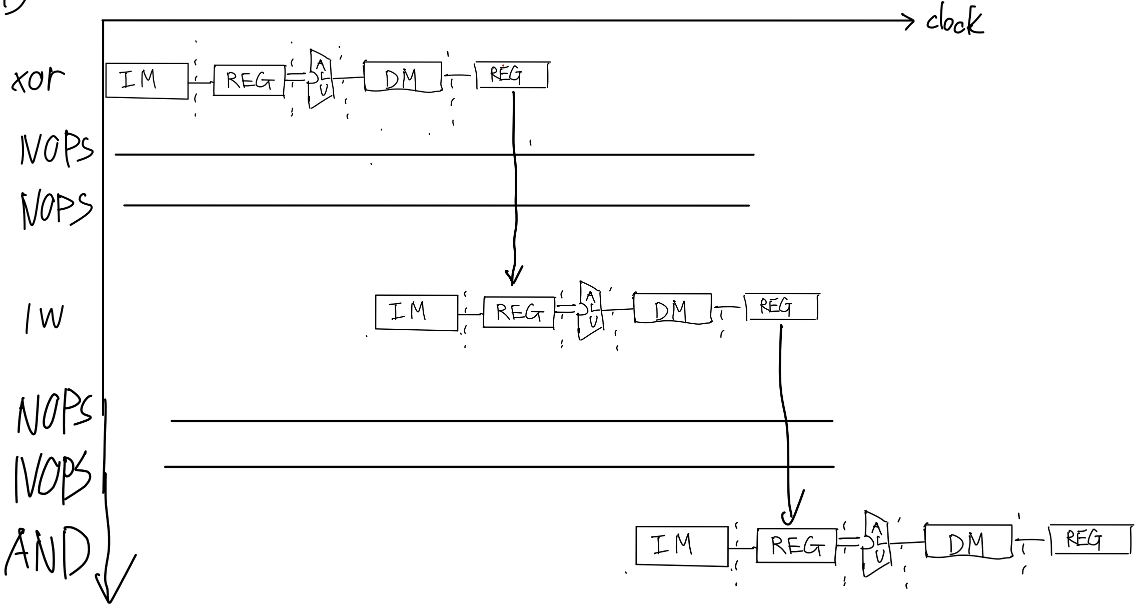
Q3 (20%)

Consider the following sequence of instructions, and assume that it is executed on a five-stage pipelined datapath:

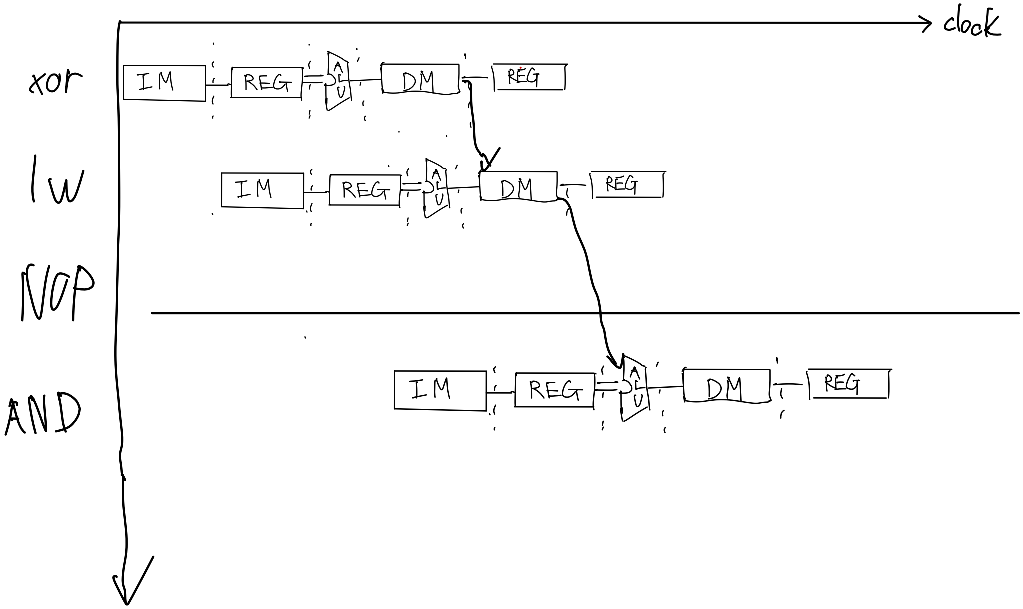
```
xor x13, x12, x11
lw x14, 8(x13)
and x10, x15, x14
```

1. If there is no forwarding or hazard detection, insert NOPs to ensure correct execution. You need to draw a pipeline diagram like we did in the lectures.

1)



2. If hazard detection and forwarding are allowed, draw a pipeline diagram with forwarding to ensure correct execution. You may also use NOP(s), but you need to use at least NOPs as possible.

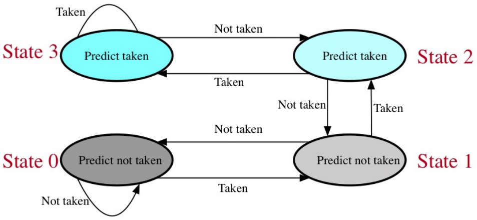


1 NOP need.

Q4 (20%)

This exercise examines the accuracy of various branch predictors for the following repeating pattern (e.g., in a loop) of branch outcomes: T, NT, NT, NT, T. (T means 'Taken' and NT means 'Not taken')

1. What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes?
2. What is the accuracy of a 2-bit predictor if this pattern is repeated forever? You should give your explanations rather than just an answer. The following figure shows the finite-state machine for a 2-bit prediction scheme. Here we assume this predictor starts from "State 0".



1) taken = $\frac{2}{5}$ ×100% = 40%

Not taken = $\frac{3}{5}$ ×100% = 60%

b)

	i1					i2				
Branch output	T	NT	NT	NT	T	T	NT	NT	NT	T
state	0	1	0	0	0	1	2	1	0	0
Predict	NT	NT	NT	NT	NT	NT	T	NT	NT	NT
Change?	✓	✓	×	×	✓	✓	✓	✓	×	✓
Accurate	×	✓	✓	✓	×	×	×	✓	✓	×

'The Branch output will change from stage 1 to 2. and reapted itself at the last.

accuracy = $\frac{5}{10}$ = 0,5

= 50%

Q5 (20%) Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (classes A, B, C, and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 2 GHz and CPIs of 2, 2, 2, and 2. A program with an instruction count of 1.0E6 can be divided into classes as follows: 10% class A, 20% class B, 40% class C, and 30% class D.

1. What is the average CPI for each implementation?
2. What is the CPU execution time of this program for each implementation?

1)

A	0.1
B	0.2
C	0.4
D	0.3

$$P1 = 0.1\times1 + 0.2\times2 + 0.4\times3 + 0.3\times3$$

$$= 2.6$$

$$P2 = 0.1\times2 + 0.2\times2 + 0.4\times2 + 0.3\times2$$

$$= 2 //$$

2) $\text{Time} = I \times CPI \times CC$

$$P1 = 1.0E6 \times 2.6 \times \frac{1}{2.5G} = 1.04\,ms$$

$$P2 = 1.0E6 \times 2 \times \frac{1}{2} = 1.0\,ms //$$