20F-COMSCIM151B-1 Homework 4

NEVIN LIANG

TOTAL POINTS

73 / 75

QUESTION 1

Dependencies 10 pts

1.1 Forwarding 4 / 5

✓ **+ 2 pts** **i1 - i2 : RAW on x3**

✓ **+ 2 pts** **i1 - i3 : RAW on x3**

+ 1 pts i3 - i4 : RAW on x6

- 1 pts Approximate description of the dependencies

+ 0 pts Incorrect

1.2 Stall 5 / 5

✓ **+ 5 pts** **Correct**

+ 2.5 pts Party correct

+ 0 pts Incorrect

QUESTION 2

Slow memory 20 pts

2.1 Pipeline diagram 10 / 10

✓ **+ 10 pts** **Correct**

+ 7 pts Almost correct

+ 0 pts Incorrect

2.2 New hazards and stalls 10 / 10

✓ **+ 3.5 pts** **2-cycle stall for load then use**

✓ **+ 3.5 pts** **1-cycle stall for load then use 2 instructions later**

✓ **+ 3 pts** **Forwarding needed when instructions are further apart**

+ 0 pts Incorrect

- 2 pts Not the most efficient solution

QUESTION 3

Dependencies 15 pts

3.1 All dependencies 5 / 5

S1

✓ **+ 1.25 pts** **i1 - i3 : RAW on x1**

✓ **+ 1.25 pts** **i2 - i3 : RAW on x6**

✓ **+ 0 pts** **i1 - i2 : WAR on x6**

+ 0 pts Incorrect

S2

✓ **+ 1.25 pts** **i1 - i2 : RAW on x5**

✓ **+ 1.25 pts** **i1 - i3 : RAW on x5**

✓ **+ 0 pts** **i1 - i3 : WAW on x5**

✓ **+ 0 pts** **i2 - i3 : WAR on x5**

+ 0 pts Incorrect

- 0.8 pts Approximate description of dependencies

3.2 Without forwarding 4 / 5

S1

✓ **+ 1 pts** **Zero nop after i1**

✓ **+ 1.5 pts** **Nops after i2**

- 0.5 pts Too many nops after i2

✓ **- 1 pts** **Too few nops after i2**

+ 0 pts Incorrect

S2

✓ **+ 1.5 pts** **Nops after i1**

✓ **+ 1 pts** **Zero nop after i2**

- 0.5 pts Too many nops after i1

- 1 pts Too few nops after i1

+ 0 pts Incorrect

3.3 With forwarding 5 / 5

S1

✓ **+ 1.25 pts** **Zero nop after i1**

✓ **+ 1.25 pts** **Zero nop after i2**

+ 0 pts Incorrect

S2

✓ + 1.5 pts Nop after i1

✓ + 1 pts Zero nop after i2

- 1 pts Too many nops after i1

+ 0 pts Incorrect

QUESTION 4

Branch prediction 20 pts

4.1 Always taken 5 / 5

✓ + 1 pts Branch 1 - correct

✓ + 1 pts Branch 2 - correct

✓ + 1 pts Branch 3 - correct

✓ + 1 pts Branch 4 - correct

✓ + 1 pts Branch 5 - correct

+ 0 pts Incorrect

4.2 Never taken 5 / 5

✓ + 1 pts Branch 1 - correct

✓ + 1 pts Branch 2 - correct

✓ + 1 pts Branch 3 - correct

✓ + 1 pts Branch 4 - correct

✓ + 1 pts Branch 5 - correct

+ 0 pts Incorrect

4.3 1-bit predictor 5 / 5

✓ + 1 pts Branch 1 - correct

✓ + 1 pts Branch 2 - correct

✓ + 1 pts Branch 3 - correct

✓ + 1 pts Branch 4 - correct

✓ + 1 pts Branch 5 - correct

+ 0 pts Incorrect

4.4 2-bit predictor 5 / 5

✓ + 1 pts Branch 1 - correct

✓ + 1 pts Branch 2 - correct

✓ + 1 pts Branch 3 - correct

✓ + 1 pts Branch 4 - correct

✓ + 1 pts Branch 5 - correct

+ 0 pts Incorrect

QUESTION 5

5 Branch delay slots 10 / 10

✓ + 10 pts Correct

+ 8 pts One too many

+ 4 pts Correct if issue width is 1

+ 0 pts Incorrect

QUESTION 6

6 Late submission penalty 0 / 0

✓ - 0 pts On time

- 18.75 pts Late submission

Homework Set – IV

Session: Fall 2020 **Instructor: Prof. P. Gupta**

Total Points: 75 **Due Date: Nov 5 (before class starts)**

Instructions

1. **There are two sections in this homework set. The questions under Section – I are for your practice. Solutions to these problems are already provided on the course web page. Students are encouraged to solve the problems before looking at the solution provided.**
2. **Problems under Section – II are to be submitted.**

Section – I

1. Consider executing the following code on the pipelined datapath of **Figure 1**:

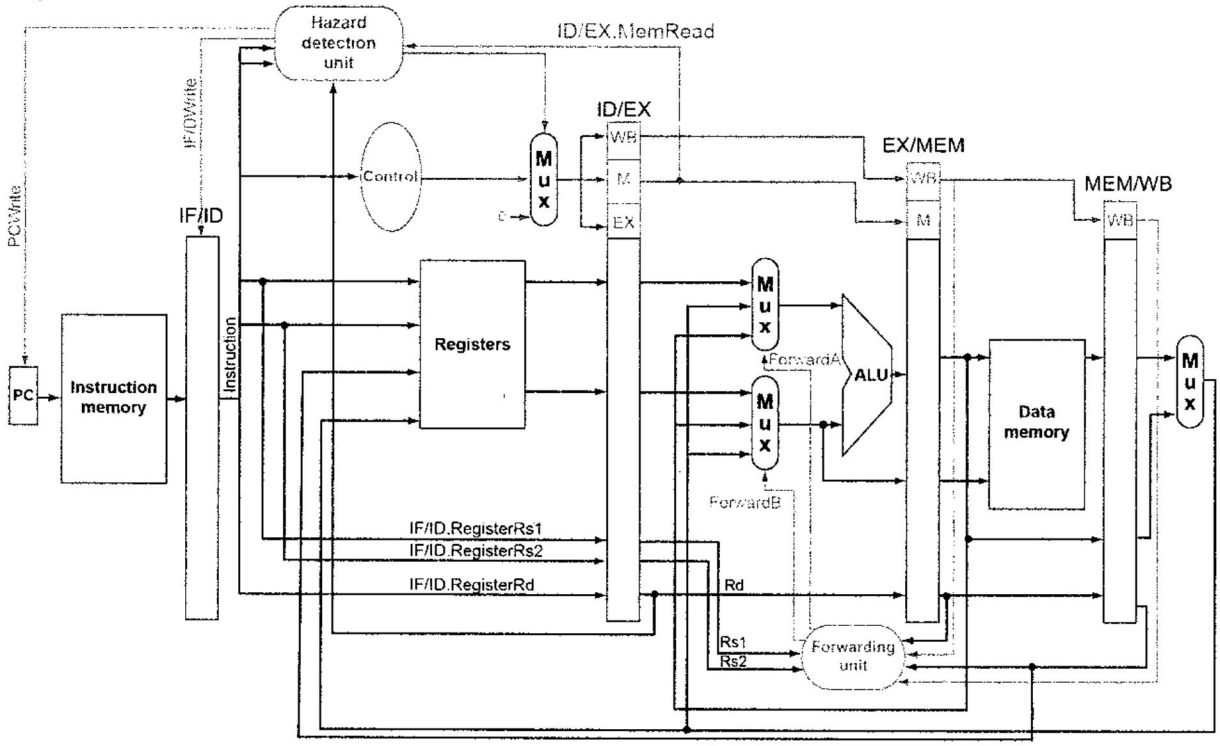
```
add  x2, x3, x1
sub  x4, x3, x5
add  x5, x3, x7
add  x7, x6, x1
add  x8, x2, x6
```

   At the end of the fifth cycle of execution, which registers are being read and which register will be written?

2. With regard to the program in problem 1 above, explain what the forwarding unit is doing during the fifth cycle of execution. If any comparisons are being made, mention them.

3. Again with regard to the program in problem 1, explain what the hazard detection unit is doing during the fifth cycle of execution. If any comparisons are being made, mention them.

Figure 1.


Hazard detection unit
ID/EX.MemRead
IF/DWrite
PCWrite
ID/EX
WB
M
EX
EX/MEM
MEM/WB
Control
Mux
IF/ID
Registers
PC
Instruction memory
Instruction
ALU
Data memory
ForwardA
ForwardB
IF/ID.RegisterRs1
IF/ID.RegisterRs2
IF/ID.RegisterRd
Rd
Rs1
Rs2
Forwarding unit

Section – II

1. Identify all of the data dependencies in the following code. Which dependencies are data hazards that will be resolved via forwarding? **[5+5 points]**

```
I1:  add   x3,  x4,  x2
I2:  sub   x5,  x3,  x1
I3:  lw    x6,  200(x3)
I4:  add   x7,  x3,  x6
```

I1 and I2 has a data hazard.
I1 ~~I2~~ and I3 also has a hazard that can be solved by forwarding.

I1 + I2 has data hazard. x3, Read after write

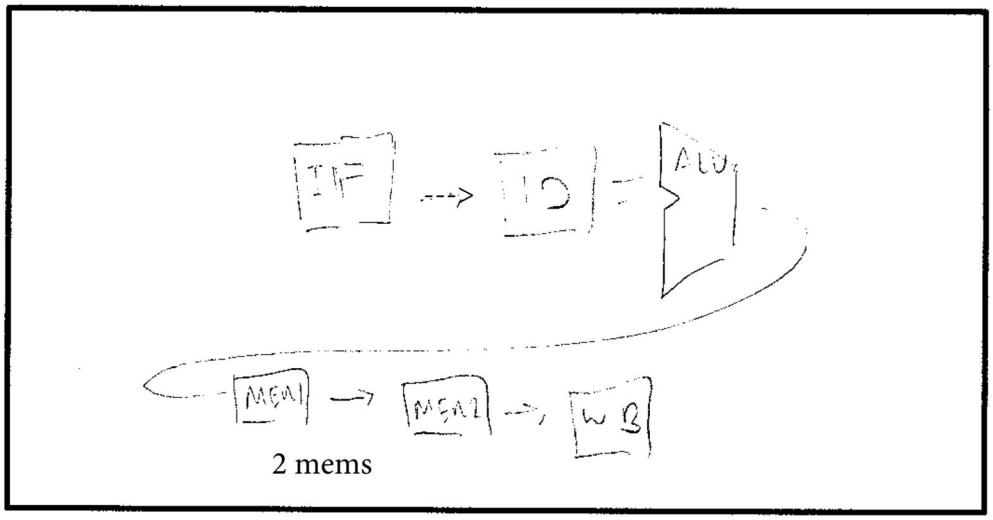
I1 + I3 has data hazard, x3, read after write

Which dependencies are data hazards that will cause a stall?

I3 and I4 has a hazard caused by a stall.

I3, I4 data hazard. x6, read after write

2. The performance advantage of the pipelined designs is limited by the longer time required to access memory versus use of the ALU. Suppose the memory access became 2 clock cycles long. Draw the modified pipeline. **[10 points]**



List all the possible new forwarding situations and all possible new hazards and required stall length. **[10 points]**

if lw:

if used in first instruction -> 2 cycle stall

second -> 1 cycle stall

third -> forward

R-type r-type

if used in first instruction -> forward

second -> forward

third -> forward.

3. We examine how data dependencies affect execution in the basic five-stage pipeline. Problems in this exercise refer to the following sequence of instructions: **[15 points]**

	Instruction Sequence
S1.	I1: lw x1, 40(x6) I2: add x6, x2, x2 I3: sw x6, 50(x1)
S2.	I1: lw x5, -16(x5) I2: sw x5, -16(x5) I3: add x5, x5, x5

a. Indicate dependencies in the above instruction sequence.

S1:

I1 + I3: x1, read after write.

I2 + I3: x6, read after write.

I1 - (I2, I3): x6, write after read.

S2:

I1 + (I2, I3): x5, RAW

(I1, I2) + I3: x5, WAR

I1 + I3: x5, WAW

b. Assume there is no forwarding in this pipelined processor. Indicate hazards and add nop instructions to eliminate them.

S1:

```
lw $1,40($6)
add $6,$2,$2
nop

sw $6,50($1)
```

S2:

```
lw $5,-16($5)
nop
nop
sw $5,-16($5)
add $5,$5,$5
```

c. Assuming there is full forwarding, indicate hazards and add nop instructions to eliminate them.

S1:

```
lw $1,40($6)
add $6,$2,$2
sw $6,50($1)
```

S2:

```
lw $5,-16($5)
nop
sw $5,-16($5)
add $5,$5,$5
```

4. We have a program core consisting of five conditional branches. The program core will be executed thousands of times. Below are the outcomes of each branch for one execution of the program core (T for taken, N for not taken).

Branch 1: T - T - T

Branch 2: N - N - N – N

Branch 3: T – N – T – N – T – N

Branch 4: T – T – T – N – T

Branch 5: T – T – N – T – T – N – T

Assume the behavior of each branch remains the same for each program core execution. For dynamic schemes, assume each branch has its own prediction buffer and each buffer initialized to the same state before each execution. List the prediction accuracy for the following branch prediction schemes: **[20 points]**

a. Always taken

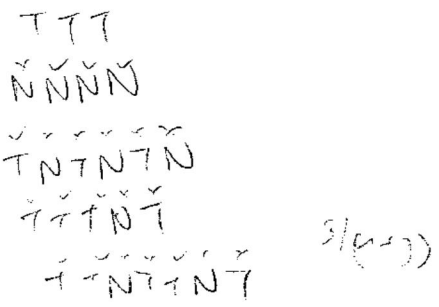
5/7

100%
0%
50%
80%
71.4%

b. Always not taken

0%
100%
50%
20%
28.6%

c. 1-bit predictor, initialized to predict taken



100 %
75 %
16 %
60 %
42 %

d. 2-bit predictor, initialized to weakly predict taken

TTNTTNT $\frac{5}{7} = 71\%$.

NNNN [75%]

100 %
75 %
50 %
80 %
71 %

5. Assume that we have a multiple-issue pipelined processor with the following number of pipeline stages, instructions issued per cycle, stage in which branch outcomes are resolved, and branch predictor accuracy:

	Pipeline Depth	Issue Width	Branches execute in stage	Branch Predictor accuracy	(%) of branches in program
a.	10	4	7	80%	20%
b.	25	2	17	92%	25%

Control hazards can be eliminated by adding branch delay slots. How many delay slots must follow each branch if we want to eliminate all control hazards in this processor? **[5+5 = 10 points]**

a) $4 \times 7 - 1 = 27$

b) $2 \times 17 - 1 = 33$

27 27
33 33