NWEN242 Homework Assignment 3

Due date: Thursday 24 September at 23:59

Total marks: 20

Q1 [6 marks] 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	350ps	150ps	300ps	200ps

Also, assume that instructions executed by the processor are broken down as follows:

alu	beq	lw	sw
45%	20%	20%	15%

a) [1 mark] What is the clock cycle time in a pipelined and single-cycle processor?

b) [1 mark] What is the total latency of an LW instruction in a pipelined and single-cycle processor?

c) [1 mark] Assuming there are no stalls or hazards, what is the utilization of the data memory? [hint: percentage of cycles used for data memory operations]

d) [1 mark] Assuming there are no stalls or hazards, what is the utilization of the write-register port of the “Registers” unit?

e) [2 marks] Instead of a single-cycle organization, we can use a multi-cycle organization where each instruction takes multiple cycles but one instruction finishes before another is fetched. In this organization, an instruction only goes through stages it actually needs (e.g., ST only takes 4 cycles because it does not need the WB stage). Compare the performance of the single-cycle and the multi-cycle organizations against the pipelined organization.

Q2 [2 marks] Consider the following loop.

```
loop: lw r1,0(r1)
      and r1,r1,r2
      lw r1,0(r1)
      lw r1,0(r1)
      beq r1,r0,loop
```

Assume that branches execute in the EX stage (not ID stage) and perfect branch prediction is used (branch is always taken) with the support of branch target buffer (that is, no stalls due to control hazards), and that the pipeline has full forwarding support.

a) [2 marks] Show a pipeline execution diagram for the third iteration of this loop, from the cycle in which we fetch the first instruction of that iteration up to (but not including) the cycle in which we can fetch the first instruction of the next iteration. Show all instructions that are in the pipeline during these cycles (not just those from the third iteration). Hints are provided by the table below. "***" indicates stalls. You may use the table to fill in your answers.

LW	R1, 0(R1)	WB							
LW	R1, 0(R1)	EX	MEM	WB					
BEQ	R1, R0, Loop	ID	***	EX					
LW	R1, 0(R1)	IF							
AND	R1, R1, R2								
LW	R1, 0(R1)								
LW	R1, 0(R1)								
BEQ	R1, R0, Loop								

Q3 [3 marks] This exercise is intended to help you understand the relationship between forwarding, hazard detection, and ISA design. Problems in this exercise refer to the following sequence of instructions, and assume that it is executed on a 5-stage pipelined datapath:

```
add r5,r2,r1
lw r3,4(r5)
lw r2,0(r2)
or r3,r5,r3
sw r3,0(r5)
```

a) [1 mark] If there is no forwarding or hazard detection, insert nops to ensure correct execution.

b) [1 mark] Is there any performance gain by reordering these instructions? Why?

c) [1 mark] If the processor has forwarding, do we need to insert nops or reorder the instructions to ensure correct execution? Explain your answer.

Q4 [3 marks] Consider the following sequence of instructions.

```
lw   $t1, 4($t2)
lw   $t2, 4($t1)
add  $t3, $t1, $t2
sw   $t3, 8($t1)
lw   $t4, 12($t0)
add  $t5, $t1, $t4
sw   $t5, 16($t2)
```

Assume that the pipeline has full forwarding support.

a) [1 mark] Without reordering the instructions, insert nops to ensure correct execution

b) [1 mark] To minimise the number of nops, how would you reorder the instructions without changing the outcome of the execution?

c) [1 mark] For the last three instructions (copied below), give the conditions (discussed in the class) for detecting the hazards (you need specify the register number, using the register number provided in the instructions instead of RegisterRs/Rt/Rd).

```
lw   $t4, 12($t0)
add  $t5, $t1, $t4
sw   $t5, 16($t2)
```

Q5 [3 marks] Consider the following sequence of instructions.

```
16:  lw   $t2, 4($t1)
20:  beq  $t2, $t1, 10
...
xx:  sw   $t2, 16($t2)
```

a) [1 mark] Assuming that the `store` instruction is the target if the branch is taken, what xx should be?

b) [1 mark] Assuming that the pipeline has full forwarding support, but does not have the extra hardware to move the branch decision to ID stage, insert nops to ensure correct execution

c) [1 mark] Assuming that the pipeline has full forwarding support, and has the extra hardware to move the branch decision to ID stage so that branch delay can be reduced, insert nops to ensure correct execution

Q6 [3 marks] This exercise examines the accuracy of various branch predictors for the following repeating pattern (e.g., in a loop) of branch outcomes: T, NT, T, T, NT (T being "taken", NT being "not taken")

a) [1 mark] What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes?

b) [1 mark] What is the accuracy of the two-bit predictor for this pattern for the first time prediction, assuming that the predictor starts off with "predict not taken"? Show which ones are predicted correctly or incorrectly.

c) [1 mark] What is the accuracy of the two-bit predictor if this pattern is repeated forever? Show which ones are predicted correctly or incorrectly.