NWEN242 Assignment 3

David Barnett (300313764)

Question 1

part a

Single cycle period $T = 250ps + 350ps + 150ps + 300ps + 200ps = 1250ps$ Single cycle frequency $f = \frac{1}{1250ps} = 800Mhz$

Pipeline period $T = Max(250ps, 350ps, 150ps, 300ps, 200ps) = 350ps$ Pipeline frequency $f = \frac{1}{350ps} = 2857Mhz$

part b

The latency until the memory access is usable after a `LW` instruction. For a single-cycle processor the delay is 0 cycles as it will be available for the next cycle to use the value immediately. The pipeline however can range from 3 cycles to 0 depending on if it writes in the first half of the cycle then reads on second half or has forwarding of the value or even if there is a data hazard at all.

part c

$lw\% + sw\% = 20\% + 15\% = 35\%$ utilisation

part d

$alu\% + lw\% = 45\% + 20\% = 65\%$ utilisation

part e

Multi Cycle

Clock cycle: `350ps` (maximum of all stages)

- alu: $IF + ID + EX + WB = 350ps + 350ps + 350ps + 350ps = 1400ps$
- beq: $IF + ID + EX = 350ps + 350ps + 350ps = 1050ps$
- lw: $IF+ID+EX+MEM+WB = 350ps+350ps+350ps+350ps+350ps = 1750ps$
- sw: $IF + ID + EX + MEM = 350ps + 350ps + 350ps + 350ps = 1400ps$

Average time to complete an instruction: $1400ps * 45\% + 1050ps * 20\% + 1750ps * 20\% + 1400ps * 15\% = 1400ps$

Time to complete 1000 instructions (assuming no NOOP for pipeline)

Single Cycle: $1000 * 1250ps = 1250000ps$

Multi-Cycle: $1000 * 1400ps = 1400000ps$

Pipeline: $(5 + (1000 - 1)) * 350ps = 351400ps$

Pipeline is the fasted to complete 1000 instructions.

Question 2

part a

Instruction	Args	T1	T2	T3	T4	T5	T6	T7	T8
LW	R1, 0(R1)	WB							
LW	R1, 0(R1)	EX	MEM	WB					
BEQ	R1, R0, Loop	ID	***	EX					
LW	R1, 0(R1)	IF	ID	***	EX	MEM	WB		
AND	R1, R1, R2		IF	ID	***	***	EX	MEM	WB
LW	R1, 0(R1)			IF	ID	***	***	EX	MEM
LW	R1, 0(R1)				IF	ID	***	***	***
BEQ	R1, R0, Loop					IF	ID	***	***

Question 3

part a

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

part b

There is no improvement to be made if the instructions were re-ordered because the total number of instructions will be the same

part c

Given full forwarding there will still need to be a nop in between `add r5,r2,r1` and `lw r3,4(r5)` because of `lw`'s dependency on the result of `add`.

Question 4

part a

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

part b

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

part c

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

`lw $t4, 12($t0)`

No Hazards

`add $t5, $t1, $t4`

ID/EX.MemRead && (ID/EX.$t5 != IF/ID.$t5)

Hazard needing to insert NO OP

MEM/WB.RegWrite && (MEM/WB.$t4 != 0) && (MEM/WB == ID/EX.$t4)

`sw $t5, 16($t0)`

EX/MEM.RegWrite && (EX/MEM.$t5 != 0) && (EX/ME == ID/EX.$t5)

Question 5

part a

$XX = PC + 4 + 10 * 4 = 20 + 44 = 66$

part b

```
lw $t2, 4($t1)
nop
beq $t2, $t1, 10
nop
nop
...
sw $t2, 16($t2)
```

part c

```
lw $t2, 4($t1)
nop
beq $t2, $t1, 10
nop
...
sw $t2, 16($t2)
```

Question 6

part a

The accuracy for always taken is 60% and always not taken is 40%

part b

Starting off in "predict not taken" has two possible starting positions, needing one miss to change or two.

Note number in brackets are number of misses needed to swap to the other value

Started with predicting in NT(0)

- T: Miss - predicted next is T(0)
- NT: Miss - predicted next is T(0)
- T: Correct - predicted next is T(1)
- T: Correct - predicted next is T(1)
- NT Miss - predicted next is T(0)

Started with predicting in NT(1)

- T: Miss - predicted next is NT(0)
- NT: Correct - predicted next is NT(1)
- T: Miss - predicted next is NT(0)
- T: Miss - predicted next is T(0)
- NT Miss - predicted next is NT(0)

part c

Note number in brackets are number of misses needed to swap to the other value

First cycle starting with predicting NT(0)

- T: Miss - predicted next is T(0)
- NT: Miss - predicted next is T(0)
- T: Correct - predicted next is T(1)
- T: Correct - predicted next is T(1)
- NT Miss - predicted next is T(0)

Looping forever:

- T: Correct - predicted next is T(1)

- NT: Miss - predicted next is T(0)
- T: Correct - predicted next is T(1)
- T: Correct - predicted next is T(1)
- NT Miss - predicted next is T(0)

Started with predicting in NT(1)

- T: Miss - predicted next is NT(0)
- NT: Correct - predicted next is NT(1)
- T: Miss - predicted next is NT(0)
- T: Miss - predicted next is T(0)
- NT Miss - predicted next is NT(0)

Looping forever:

- T: Miss - predicted next is NT(1)
- NT: Correct - predicted next is NT(1)
- T: Miss - predicted next is NT(0)
- T: Miss - predicted next is T(0)
- NT Miss - predicted next is NT(0)