

Kohar Thaveri
830962238

ECE 452

Q1

IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

a) clock cycle time(pipelined) = ~~clock time~~
~~longest stage~~ clock time
= clock time of ID
= 350ps,

clock cycle time (non-pipelined) =
 Σ clock time of all stages
= 250+350+150+300+200
= 1250ps,

b) Total latency of Lw (pipelined.) =
longest stage clock time \times
number of stages
= 350ps \times 5 = 1750ps,

Total latency of Lw from each instruction
after that will take 350ps.

Total Latency of LW (non-pipelined) =
 \sum clock time of all stages

$$= 250 + 350 + 150 + 300 + 200$$

$$= 1250 \text{ ops}$$

Each instruction loadword after
will take 1250ops

c) ID stage should be the one which we should split into 2 stages. The reason for that ID has the longest clock time and splitting it would mean the latency of that stage is reduced to half and hence the new total clock cycle time would be the clock time of MEM stage instead of ID stage and hence the total clock cycle time is

reduced from 350 ps to 300 ps,

d) Data memory is utilized only by load word(lw) & store word(sw) instructions. So assuming there are no stalls or hazards, the utilization of data memory would be 20% + 15% = 35%.

e) Write-register port of the "Registers" unit are utilized only by ALU & load word(lw) instructions. So assuming there are no stalls or hazards, the utilization of write-register would be 45% + 20% = 65%.

Q2

Sw $r16, 12(r6)$

lw $r16, 8(r6)$

beq $r5, r4, \text{Label}$ # Assume $r5 = r4$

add $r5, r1, r4$

slt $r5, r15, r4$

Stalls occur when an instruction fetch cannot happen because the memory is accessed by either load word (lw) or store word (sw) instructions. In our case the pipeline execution would be as follows.

	1	2	3	4	5	6	7	8	9	10	11
Sw $r16, 12(r6)$	IF	ID	EX	MEM	WB						
lw $r16, 8(r6)$	IF	ID	EX	MEM	WB						
beq $r5, r4, \text{Label}$	IF	ID	EX	MEM	WB						
add $r5, r1, r4$	stall	stall	IF	ID	EX	MEM					
slt $r5, r15, r4$			IF	ID	EX		MEM	WB			

- ∴ The number of cycles due to one structural hazard is 11 cycles
 - ∴ The Total execution time would be equal to $11 \times 2.00 = 2200 \text{ ps}_{\parallel}$
- \downarrow \downarrow
No. of Total
cycles clock
 cycle time

Unlike data hazards, in structural hazard you cannot add nops to eliminate them as they themselves are fetched from memory hence a stall would occur in any case.

b) \because EX and MEM are combined
 load word (lw) & store word (sw)
 cannot access the memory directly.
 Hence, the modified code would be,
 $\text{addi } r8, r6, 12 \# \text{stores the address for } s_1$
 $\text{addi } r9, r6, 12 \# \text{stores the address for } l_1$
 $\text{sw } r16, r8$
 $\text{lw } r16, r9$
 $\text{beq } r5, r4, \text{Label} \# \text{Assume } r5 = 04$
 $\text{add } r5, r1, r4$
 $\text{slt } r5, r15, r4$

\therefore The pipeline would look like

$\text{sw } r16, r8$	1	IF	ID	EX/MEM	WB		
$\text{addi } r8, r6, 12$		IF	ID	EX/MEM	WB		
$\text{lw } r16, r9$	2	IF	ID	EX/MEM	WB		
$\text{beq } r5, 04, \text{Label}$		3	IF	ID	EX/MEM	WB	
$\text{add } r5, r1, r4$			4	IF	ID	EX/MEM	WB
$\text{slt } r5, r15, r4$				5	6	7	8

Hence, now the ~~new~~^{no} number of cycles are 8 which earlier was 9.

∴ The speedup is $\frac{200 \times 9}{200 \times 8} = 1.125$ //

c.) SW \rightarrow 16, 12(r_6) LW \rightarrow 16, 8(r_6) beq \rightarrow 5, 4, Label branch results add \rightarrow 5, 21, 4 Slt \rightarrow 5, 15, 4

	1	2	3	4	5	6	7	8	9	10
IF	IF	ID	EX	MEM	WB					
IF	IF	ID	EX	MEM	WB					
IF	IF	ID	EX	MEM	WB					
Stall	IF	ID	EX	MEM	WB					
	IF	ID	EX	MEM	WB					

\therefore The number of cycles required to execute is reduced to 10 otherwise it would have been 11 cycles.

$$\therefore \text{Speed up} = \frac{[(\text{cycle time}) \times (\text{No. of cycles})]_{\text{old}}}{[(\text{cycle time}) \times (\text{No. of cycles})]_{\text{new}}}$$

$$= \frac{11 \times 200}{10 \times 200} = \frac{11}{10} = 1.10\%$$

d) I.F I.D EX / MEM WB

I.F	I.D	EX / MEM	WB
200ps	120ps	210ps	100ps

In this case the cycle time would be 210ps for 4 stages

Hence, Speedup = $\frac{9 \times 200}{8 \times 210} = 1.0714$

I.F	I.D	EX / MEM	WB
200ps	120ps	210ps	100ps

e) Latency of ID stage is ↑ by 50%
and Latency of EX stage ↓ by 10ps
when branch results are moved
from EX to ID. The pipeline would
remain the same as in (c.)

∴ The pipeline latency would be

IF	ID	EX	MEM	WB
200ps	180ps	140ps	190ps	100ps

But this brings no change in
clock cycle time.

∴ The speedup remains the
same as in (c) part.

f) Now, branch instruction result is moved to MEM stage. Hence the pipeline would look like.

	1	2	3	4	5	6	7	8	9	10	11	12
sw r16, 12(r6)	IF	ID	EX	MEM	WB							
lw r16, 8(r6)	IF	ID	EX	MEM	WB							
beq r5, r5, Label	IF	ID	EX	MEM	WB							
add r5, r1, r4			stall	stall	IF	ID	EX	MEM	WB			
sh r5, r15, r4			stall	stall	IF	ID	EX	MEM	WB			

The pipeline latency would be as follows

IF	ID	EX	MEM	WB
200ps	120ps	130ps	190ps	-100

$$\therefore \text{Speed up} = \frac{11 \times 200}{12 \times 200} = \frac{11}{12} = 0.916 //$$

Hence, in this case it is a slow down as there are 3 stalls and hence the number of cycles increases.

Q 3

R-Type	BEQ	JMP	LW	SW
40 J.	25 J.	5 J.	25 J.	5 J.

15 J.	SS J.	85 J.
Always taken	Always not taken	2-bit

- a) A mispredicted branch will cause 3 stalls for always taken predictor
∴ The ^{additional} ~~new~~ CPI = $0.25 \times (1 - 0.45) \times 3$
 $= 0.4125_{II}$

- b) A mispredicted branch will cause 3 stalls for always not-taken predictor
∴ The ^{additional} ~~new~~ CPI = $0.25 \times (1 - 0.55) \times 3$
 $= 0.3375_{II}$

- c) A mispredicted branch will cause 3 stalls for a 2-bit predictor
∴ The ^{additional} ~~new~~ CPI = $0.25 \times (1 - 0.85) \times 3$
 $= 0.1125_{II}$

d) Let the CPI of branches predicted correctly & incorrectly be ~~0.1~~ 1 CPI without converting the instructions is ~~1.125~~ as calculated in (a) which is 1st additional CPI [From(c)] CPI with conversion will be

$$1 + 0.5 \times 0.25 \times (1 - 0.85) \times 3 = 1.0562$$

$$\therefore \text{Speedup} = \frac{\text{CPI without conversion}}{\text{CPI with conversion}}$$

$$= \frac{1.1125}{1.0562}$$

$$= 1.053311$$

e) After conversion, it requires 2 ALU cycles instructions
∴ The execution time increases by 1 cycle.

$$CPI \text{ without conversion} = 1.1125 \text{ [From d]}$$

$$\begin{aligned} CPI \text{ with conversion} &= 1 + [1 + 0.25 \times \\ &\quad (1 - 0.85) \times 3] \times 0.25 \times 0.5 \\ &= 1.1812 \end{aligned}$$

$$\begin{aligned} \therefore \text{Speedup} &= \frac{1.1812}{1.1125} \quad \frac{1.1125}{1.1812} \\ &= 1.0617 \quad 0.9418 \end{aligned}$$

Hence, here it is a slowdown

f.) Assume The total branch instruction are x .

\therefore Easy to predict branch instruction would be $0.8x$ i.e. 80%.

Out of which $0.85x$ are predicted correctly i.e. 85%.

\therefore There are $0.2x$ difficult to predict branch instructions, i.e. 20%.

Amongst these $0.2x$ i.e. 20%

instructions $0.85x - 0.8x = 0.05x$ are predicted correctly

\therefore The Accuracy of a 2 bit branch predictor would be $\frac{0.05x}{0.2x} = 0.25$

i.e. 25%.