ES215 : Assignment-3

Name: Daniel Cliftson E

Roll No: 20110051

1) a) • Single-cycle processor: An instruction finishes in one clock cycle. ∴ the best clock cycle time is the max-time that an instruction will take to execute. Load word (lw) instructions take the max-time.

Clock cycle time = (200 + 100 + 200 + 400 + 100) ps

⇒ 1000 ps.

Clock frequency ⇒ $\frac{1}{1000 \times 10^{-12}}$ ⇒ 1 GHz

• Multi-cycle processor: An instruction executes in 5 cycles.

∴ The best clock cycle time will be the maximum of time taken by IF, ID, EX, MEM or WB.

Clock cycle time ⇒ max (IF, ID, EX, MEM, WB)

⇒ 400 ps.

Clock frequency ⇒ $\frac{10^{12}}{400}$ ⇒ 2.5 GHz

• Pipelined processor: Similar to multi-cycle processor, an instruction will take 5 cycles to execute and the best clock cycle time will be the maximum of the 5 stages (IF, ID, EX, MEM or WB)

⇒ Clock cycle time ⇒ 400 ps.

Clock frequency = $\frac{10^{12}}{400}$ ⇒ 2.5 GHz

(b) We know that LW ~~takes just one cc~~ executes in "one-cycle" in single-cycle processor and 5 cycles in both multi-cycle and pipelined processors.

$$\therefore (\text{Latency})_{\text{single-cycle}} = 1000 \times 10^{-12} \times 1 \Rightarrow 1000 \text{ ps.}$$

$$(\text{Latency})_{\text{multi-cycle}} = 400 \times 10^{-12} \times 5 \Rightarrow 2000 \text{ ps.}$$

$$(\text{Latency})_{\text{pipelined}} = 400 \times 10^{-12} \times 5 \Rightarrow 2000 \text{ ps.}$$

(c) • Single-cycle processor: We know that every instruction will take only one-cycle to execute. Therefore, ~~sinstruction~~ irrespective of the instruction mix and type of instructions, we can conclude that

$$\Rightarrow (\text{Execution time})_{\text{single-cycle}} \Rightarrow \text{IC} \times (\text{Latency})_{\text{single-cycle}}$$

$$\Rightarrow 10^{6} \times 1000 \times 10^{-12}$$

$$\Rightarrow 1\text{ms}$$

• Multi-cycle processor: The no. of cycles depends on what kind of instruction.

	IC	No. of cycles
ALU	5×10^5	4
BEQ	2×10^5	3
LW	1.5×10^5	5
SW	1.5×10^5	4

$\Rightarrow$ (Execution time)$_{multi\text{-}cycle}$ $\Rightarrow (5\times10^{5}\times4\times400\times10^{-12}) + (2\times10^{5}\times3\times400\times10^{-12}) + (1.5\times10^{5}\times5\times400\times10^{-12}) + (1.5\times10^{5}\times4\times400\times10^{-12})$

$\Rightarrow$ 0.8 ms + 0.24 ms + 0.3 ms + 0.24 ms.

$\Rightarrow$ 1.58 ms

- Pipelined processor:

(Execution time)$_{pipelined}$ $\Rightarrow$ $\underbrace{5(400\times10^{-12})}_{\text{Execution time for 1st instruction}} + \underbrace{(10^{6}-1)(400\times10^{-12})}_{\text{Additional time taken by following instruction}}$

$\Rightarrow (10^{6}+\cancel{4})(400\times10^{-12})$

Since $10^{6} >>> 4$.

$\Rightarrow$ 0.4 ms

- Since (Execution-time)$_{multi\text{-}cycle}$ > (Execution-time)$_{single\text{-}cycle}$, there won't be a speedup achieved by multi-cycle processor w.r.t single cycle processor.

- Speedup (of pipelined w.r.t single-cycle processor) = $\frac{1\times10^{-3}}{0.4\times10^{-3}} \Rightarrow 2.5$ (Since (Execution-time)$_{pipeline}$ < (Execution-time)$_{single\ cycle}$)

d) Since we are assuming intermediate buffers, ~~there will be~~ which consume 1ops; the additional execution time will be $= 10 \times 10^{-12} \times 10^{6} \Rightarrow 0.01$ ms

$\therefore$ (Execution time)$_{\text{pipelined}}$ (in the presence of buffers) $\Rightarrow 0.4 + 0.01$

$\Rightarrow 0.41$ ms

$\Rightarrow$ Speedup (w.r.t single-cycle processor) $= 1/0.41 = 2.439$

$\therefore$ The Speedup achieved by the pipelined processor w.r.t single-cycle processor will decrease due to the presence of intermediate buffers.

2) From the data given in the question, we get.

Stage	IF	ID	EX	MEM	WB
Latency	1000ps	100ps	100ps	1000ps	100ps

Let's assume the no. of instruction in the program

$IC = 20$

a). Single-cycle processor: Each instruction is executed in 1 clock cycle and thus the best clock cycle time will be cummulative ~~total~~ sum of all latencies (5 stages)

⇒ Clock cycle time (T_c) ⇒ 1000 + 100 + 100 + 1000 + 100 ⇒ 2300 ps.

⇒ Execution time ⇒ $IC \times CPI \times T_c$ ⇒ (20)(1)(2300)

⇒ 46,000 ps

• Multi-cycle processor: Each stage of the datapath is executed in 1 cycle. ∴ $CPI = 5$. ∴ The best clock cycle time will be the time taken to execute the longest stage in the datapath.

T_c (Clock cycle time) = 1000ps.

Execution Time ⇒ 20 × 5 × 1000ps ⇒ 1,00,000 ps

b) Five-stage pipeline design:

Let's assume there are no hazards in the given program.

→ Clock cycle time (T_c) ⇒ max (IF, ID, EX, MEM, WB) + Overhead

⇒ 1000 + 100 ⇒ 1100 ps.

⇒ ~~CPI = 5~~

Execution time = 5 (1100 ps) + (20 − 1)(1100 ps).

⇒ 5500 ps + 20900 ps.

⇒ 26400 ps

c) • We can implement more forwarding / bypassing units in our pipeline design to reduce stalling during occurences of hazards. (Price / Power will increase)

• We can reduce the clock cycle time by splitting the longest stages of the ~~dat~~ pipeline datapath (Here IF and MEM). For eg: Let's split IF and MEM into 2 stages each of 500ps. ∴ Now there will be 7 pipeline stages but the Clock cycle time will be 500 + 100 = 600ps.

Thus Execution time = 7(600ps) + 19(600ps).

→ 15,600 ps

. (Considering there are no hazards or at max 1 ~~RAW~~ hazard as there is atmost one forwarding unit)

∴ In this case, the performance is better than the previous 5-stage pipeline ~~&~~ processor.

3) Instructions:

I_1: add $s0 $t0 $t1
I_2: add $s0 $s0 $t2
I_3: add $s1 $s1 $s0

(a) Case 1:

	1	2	3	4	5	6	7	8	9	10	11	12	13
I_1:	IF	ID	EX	MEM	WB								
		(1-5)											
Stall													
Stall													
Stall													
I_2:						IF	ID	EX	MEM	WB			
						(5-9)							
Stall													
Stall													
Stall													
I_3:										IF	ID	EX	MEM WB
										(9-13)			

(Execution time)$_1$ $\Rightarrow$ 13 (T_c) $\rightarrow$ clock-cycle time.

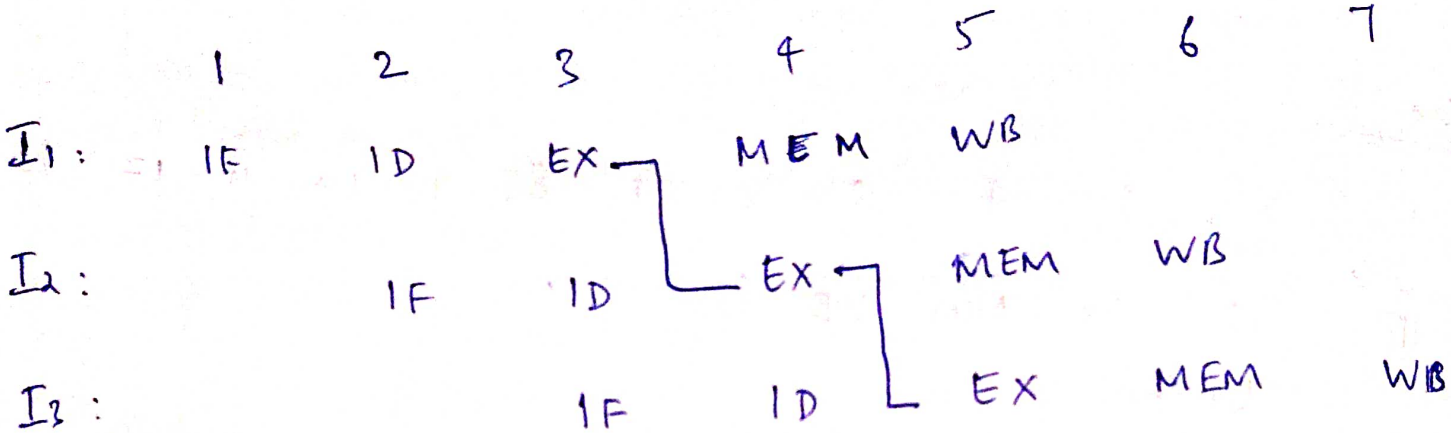
(b) Case 2: Since WB finishes register update in the first half cycle and ID can read the register in the second half of the cycle, there will be only "2 stalls" in b/w $I_1 - I_2$ and $I_2 - I_3$ (2 each).

I_1: 1-5 I_3: 7-11
I_2: 4-8

Execution time

$(\text{Execution time})_2 = 11\,T_c$

(c) Case 3: (Forwarding)



$(\text{Execution time})_3 = 7\,T_c$

(d) $(\text{Speedup})_{2 \text{ w.r.t } 1} \Rightarrow \dfrac{(\text{Execution time})_1}{(\text{Execution time})_2}$

$\Rightarrow \dfrac{13\,T_c}{11\,T_c} \Rightarrow 1.182$

$(\text{Speedup})_{3 \text{ w.r.t } 1} \Rightarrow \dfrac{(\text{Execution time})_1}{(\text{Execution time})_3}$

$= \dfrac{13\,T_c}{7\,T_c} \Rightarrow 1.857$