Computer Organization and Architecture

A4-COE

Assignment :

Neeraj Sharma

2K19/ CO/244

Ans:1 $n = 500$ instructions

stages $m = 5$

Probability. instruction being branch $p = 0 \cdot 3$

$$\text{speed up} = \frac{m * n}{n + m - 1 + p * m (n-1)}$$

$$= \frac{500 \times 5}{5 + 499 - 1 + 0 \cdot 3 (500 * 4)}$$

$$= 2 \cdot 26$$

for $5n = \frac{2500}{504 + 2000p} \geq 4$

Hence $p \leq \frac{625 - 504}{2000}$

or $p \leq 0 \cdot 0605$

$\therefore$ number of branch instructions

$\leq 0 \cdot 0605 \times 500$

$\leq 32 \cdot 5$

$$5n = \frac{2500}{504 + 2000p} \geq 5$$

2K19/C0/244

$\Rightarrow p \le -\frac{4}{2000}$

∴ speed up of 5 is impossible

Ans. 2 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5

(a) FIFO:

	PF	PF	PF	PF	PF	PF	PF	PF	PF
1	2	3	4	5	1	2	3	4	5
	1	1	1	1	5	5	5	5	4
		2	2	2	2	1	1	1	1
			3	3	3	3	3	2	2
				4	4	4	4	3	3

1	2	3	4	5	
4	4	4	3	3	3
5	5	5	5	4	4
2	1	1	1	1	5
3	3	2	2	2	2
PF	PF	PF	PF	PF	PF

Hit ratio = 0%.

(b) LRU:

1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	
	1	1	1	1	5	5	5	5	4	4	4	4	3	3	3
		2	2	2	2	1	1	1	1	5	5	5	5	4	4
			3	3	3	3	3	2	2	2	1	1	1	1	5
				4	4	4	4	3	3	3	3	2	2	2	2

Hit ratio = 0%.

Ans: 3

4, 7, 5, 7, 6, 7, 10, 4, 8, 5, 6, 8, 11, 4, 9, 5, 9, 6, 9, 12, 4, 7, 5, 7

Using LRU technique

4	~~7~~	5	7	6	7	10	4	8	5	8	6
				6	6	6	6	8	8	8	8
		5	5	5	5	5	4	4	4	4	4
	7	7	7	7	7	7	7	7	5	5	5
4	4	4	4	4	4	10	10	10	10	10	6

8	11	4	9	5	9	6	12	4	7	5	7
8	8	8	8	5	5	5	5	4	4	4	4
4	11	11	11	11	11	6	6	6	7	7	7
5	5	4	4	4	4	4	12	12	12	12	12
6	6	6	9	9	9	9	9	9	9	5	5

Number of page faults = 17.

Ans. 4

ta = aresage time

t_{mm} → main memory-access time = 100ns

t_c = cache access time = 20ns

n = interleaving factor.

n = interleaving factor

$t_a < 60$

$t_c + \frac{t_{mm}}{n} \leq 60ns$

$20 + \frac{100}{n} < 60$.

$\Rightarrow \quad n > \frac{100}{40}$

$\Rightarrow \quad n > 2.5$

Since n is usually a power of 2

$\therefore n = 4$

Average access time of resulting system

$= 20 + \frac{100}{4} = 45ns$. .

Ans: 5

t_c = access time of cache memory = 20ns

$t_{mm} \rightarrow$ access time of semiconductor.

main memory = 100ns

t_{dsm} = access time of disk secondary

memory = 1ms

$= 10^6 ns.$

Average access time

$h_c t_c + (1-h_c)[h_m t_m + (1-h_m) t_d]$

$= (0.9 \times 20) * (0.1)[(0.95 \times 100) + (0.05 \times 10^6)]$

$= 18 + (0.1)(95 + 50000)$

$= 5027.5$ ns.

Ans: 6. (a)	Horizontal Microinstructions		Vertical Micro instructions.
①	It supports longer control word	①	It supports shorter control word
②	It allows higher degree of parallelism.	②	It allows low degree of parallelism.
③	No additional hardware is required.	③	Additional hardwares like decoders are required
④	Faster than vertical microprogrammed control unit	④	Slower than horizontal microprogrammed control unit

⑤ It is less flexible	⑤ It is more flexible
⑥ Energy bit in the control field attaches to a control line	⑥ Code is used for Each action and the decoder translates this code into individual control signals.

b) Hardwared Control Unit	Microprogrammed Control Unit
① It generates the control signals needed for the procesor using logic circuits	① It generates the Control signals with the help of micro-instructions stored in control memory.
② It is faster as the required control signals are generated with the help of hardwares	② It is slower than the other as micro instructions are used for generating signal here

3.	Different to modify as the control signals that need to be generated are hard wired	3	Easy to modify as the modification only need to be done at the instruction level
4.	It cannot handle complex instructions as the circuit design for it becomes complex	4.	It can handle complex instructions

Ans: 7 Structural Hazards

These hazards arise due to resource conflict in the pipeline. A resource conflict is a situation where more than one instruction tries to access the same resource in the single cycle..

A resource can be a register, memory or ALU.

Example	1	2	3	4
I_1	If (Mem)	ID	EX	Mem
I_2		If	ID	EX
I_3			If	ID
I_4				If / Mem

- Solution: Resource duplication can be a possible sol^n. We divide the memory into two individual modules used to store instruction and data seperately.

- Control Hazards

These hazards occur during the transfer of control instructions such as BRANCH, CALL etc.

Due to this unwanted instructions are fed to pipeline.

Example :

Data Hazard :

These Hazards occurs when instructions that exihibit data dependence.

Let there be two instructions. I, J.

Read after write occurs.

when J tries to read before I has written

I : $R_2 \leftarrow R_1 + R_3$

J : $R_4 \leftarrow R_2 + R_3$.

write after read

occurs when J tries to write before I reads it

I : $R_2 \leftarrow R_1 + R_3$

J : $R_3 \leftarrow R_4 + R_5$

write after write

occurs when J tries to write output before I

I : $R_2 \leftarrow R_1 + R_3$

J : $R_2 \leftarrow R_4 + R_5$.

	1	2	3	4	8
I_1	IF	ID	EX	MEM	WB
I_2		IF	ID(250)	EX	MEM
I_3			IF	ID	EX
			IF		

∴ I_3 is partially executed here.

→ Flushing: It is the worst case scenario either we remove the computed data or we don't work at all

→ N.OP(No - operation): Meaning compiler won't execute any instructions

→ Code rearrangement: or delayed load.

Here with smart compilers we can first execute some instructions which are independent from the current logic

→ Estimation.

Here we can use either 2 stage item then policy change

Solution :

→ We can use code movement or code relocation

→ Also we can use operator forwarding using which we can directly access the result after execution instead of waiting that it gets stored in the memory

Ans: 8 Operand forwarding : In this technique, the value of operand is given to the concerned stage of dependent instruction before it is used.

I_2 is dependent on I_0 and I_1 and

I_3 is dependent on I_2

Operand forwarding ($R_2 + R_5$)

Clock cycles.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I_0	IF	ID	OF	PO	PO	PO	WO								
I_1		IF	ID	←	→	OF	PO	PO	PO	PO	PO	PO	WO		
I_2			IF	ID	←						→	OF	PO	WO	
I_3				IF	ID	←						→	OF	PO	WO

Stall cycles

Operand R_5/R_4 forwarding

2K19/CO/244

Here I0 is a MUL operation which takes 3 clock cycles in PO stage and 1 cycle for rest.

→ I1 is a DIV operation which takes 6 cycles in PO stage, and 1 cycle for rest

→ I2 is an ADD operation which takes cycle of CPU for all stages but is a dependent operation.

→ I3 is again uses operand forwarding

Hence, total no. of cycle = 15 option (B).

Ans: 9

1	2	3	4	5	6	7	8	9	10	11	12	13
IF	OF	PO	PO	WB								
	IF	OF		PO	PO	PO	PO	PO	PO	WB		
		IF		OF						PO	WB	
			IF	OF							PO	WB

Total no. of cycles = 13.

2K19/CO/244.

Ans:10, Here each stage takes 1 CPI on an average.

In 1st case 80% take 1 clock & 20% take clocks.

$\therefore$ total time

$P = (0.8 \times 1 + 0.2 + 3) \times 2.2$

$= 3.08$

$q = (0.8 \times 1 + 0.2 \times 6) \times 1$

$= 2$

$\frac{p}{q} = 3.08/2 = 1.54$