MID SEM EXAMINATION - 2019 (1)

COMPUTER ARCHITECTURE

6th SEM, ELECTRICAL ENGG

Q1.

a) LOCALITY OF REFERENCE

CPU performs READ or WRITE operation only on cache memory. Cache memory maintains image of main memory. This property is called as principle of locality of reference.

b) SUBROUTINE

A subroutine is a sequence of program instructions that performs a specific task.

INTERRUPT SERVICE ROUTINE

Interrupt service routine is a software process invoked by an interrupt request from a hardware device. It handles the request and sends it to CPU, interrupting the current process.

c) SYSTEM BUS ARBITRATION

It refers to process by which current bus master accesses and then leaves the control of bus and passes it to another bus requesting processor unit. There are two approaches of to bus arbitration

→ Centralized bus arbitration
→ Distributed bus arbitration

d) 128 X 8 RAM.

→ Memory capacity = 2048

No of chips required = $\frac{2048}{128} = 16.$

e)

		4	4	4	5	5	5	6
	3	3	3	2	2	2	4	4
2	2	2	6	6	6	1	1	1
*	*	*	*	*	*	*	*	*

No of hits = 3

Hit ratio = $\frac{3}{12}$

Opt → iv) None of these.

---

f.) 64 B cache lines

$64\ B = 2^6$

opt c) ⇒ 6 bits

Q2. a) PAGE FAULT : The processor is trying to access the page which is not currently available in the main memory. This situation is called page fault.

The fault notifies the operating system that it must locate data in virtual memory, then transfer it to system RAM.

Different types of page replacement algorithms are:

i) FIFO algorithm : In the event of page fault, replace the page which has been used in the earliest stage.

ii) LRU algorithm : In the event of page fault, replace the page which is least recently used.

iii) MRU algorithm : In this event of page fault, replace the page which is most recently used.

iv) Optimal page replacement algorithm : In the event of page fault, replace the page which is not used for longest duration of time.

b) No. of frames = $\frac{4K}{1K} = 4$.

i) FIFO

			1	1	1	1	3	3	3
		0	0	0	0	2	2	2	2
	2	2	2	2	4	4	4	4	7
4	4	4	4	6	6	6	6	5	5
*	*	*	*	*	*	*	*	*	*

page fault = 10
Available page in main memory
⇒ 5, 7, 2, 3

ii) LRU

			1	1	1	1	1	1	1	1	1	1	5	5
		0	0	0	0	0	4	4	4	4	4	3	3	3
	2	2	2	2	2	2	2	0	0	0	0	0	0	7
4	4	4	4	4	6	6	6	6	6	6	2	2	2	2
*	*	*	*		*		*	*			*	*	*	*

Available page in main memory
⇒ 5, 3, 7, 2

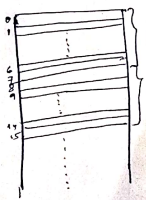
Q3. a) CACHE MISS : There are three types :

i) Compulsory miss → It always happen at the starting time of mapping process.

ii) Capacity miss → This miss happens due to memory overflow.

iii) Conflict miss → Same type of blocks are fetched into same line no.

b)



B0 (0-7)
B1 (8-15)
B2 (16-23)
B3 (24-31)
B4 (32-39)
B5 (40-47)
B6 (48-55)
B7 (56-63)
B8 (64-71)
B9 (72-79)
B10 (80-87)
B11 (88-95)
B12 (96-103)

Memory adds	Block #	Hit/ miss
5	B_0	Miss
17	B_2	Miss
64	B_8	Miss
18	B_2	Hit
26	B_3	Miss
16	B_2	Hit
68	B_8	Hit
74	B_9	Miss
80	B_{10}	Hit

Memory adds	Block #	Hit/Miss
84	B_{10}	Miss
92	B_{11}	Miss
100	B_{12}	Miss
64	B_8	Hit
18	B_2	Hit
26	B_3	Hit
16	B_2	Hit

K mod S = 2

04 mod 4 = 0
2 mod 4 = 2
8 mod 4 = 0
3 mod 4 = 3

9 mod 4 = 1
10 mod 4 = 2
11 mod 4 = 3

$$\% \text{ hit} = \frac{\text{Total no of hit}}{\text{Total no of access to cache memory}} \times 100\%$$

$$= \frac{8}{16} \times 100\%$$

$$= 50\%$$

0	B_{12}	B_8
1	B_9	
2	B_2	B_{10}
3	B_3	B_{11}

Q4. $Y = (A+B) * (C+D)$

Three address

Load	r_0	A	$r_0 \leftarrow M[A]$
Load	r_1	B	$r_1 \leftarrow M[B]$
Load	r_2	C	$r_2 \leftarrow M[C]$
Load	r_3	D	$r_3 \leftarrow M[D]$
~~Load~~	~~r_4~~		
ADD	r_4	r_0 r_1	$r_4 \leftarrow r_0 + r_1$
ADD	r_5	r_2 r_3	$r_5 \leftarrow r_2 + r_3$
MUL	r_6	r_4 r_5	$r_6 \leftarrow r_4 * r_5$
STORE	Y	r_6	$M[Y] \leftarrow r_6$.

Two Address Instruction

MOV	r_0	A	$r_0 \leftarrow M[A]$
ADD	r_0	B	$r_0 \leftarrow r_0 + M[B]$
MOV	r_1	C	$r_1 \leftarrow M[C]$
ADD	r_1	D	$r_1 \leftarrow r_1 + M[D]$
MUL	r_0	r_1	$r_0 \leftarrow r_0 * r_1$
MOV	Y	r_0	$M[Y] \leftarrow r_0$

1 - Address Instruction

load	A	Acc ← M[A]
ADD	B	Acc ← Acc + M[B]
STORE	T	M[T] ← Acc
Load	C	Acc ← M[C]
ADD	D	Acc ← Acc + M[D]
MUL	T	Acc ← Acc * M[T]
STORE	Y	M[Y] ← Acc.

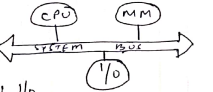
Qs:

I/O TRANSFER MODES

* I/O transfer modes are used to transfer the data from I/O devices to the CPU and memory.
* There are three types of I/O transfer modes.

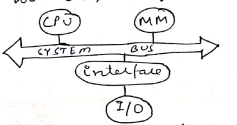
a) PROGRAMMED I/O

→ In this mode no high speed interface logic is used which implies that I/O are directly connected with CPU, therefore, processor utilisation is inefficient.



b) INTERRUPT DRIVEN I/O

→ In this mode, high speed interface logic is used b/w CPU and I/O devices, therefore processor utilisation is efficient.



c) DMA (DIRECT MEMORY ACCESS)

* In this mode of transfer, bulk amount of data is transferred from I/O devices to main memory through DMA without involvement of CPU.