CLASS TEST

S.No.: 01SKEC_HIJKLMNOPQ_6925

MADE EASY

Leading Institute for IES, GATE & PSUs

Delhi | Bhopal | Hyderabad | Jaipur | Pune | Kolkata

Web: www.madeeasy.in | **E-mail:** info@madeeasy.in | **Ph:** 011-45124612

COMPUTER ORGANISATION & ARCHITECTURE

ELECTRONICS ENGINEERING

Date of Test : 06/09/2025

ANSWER KEY

1.	(a)	7.	(d)	13.	(a)	19.	(a)	25.	(d)
2.	(c)	8.	(a)	14.	(b)	20.	(c)	26.	(d)
3.	(d)	9.	(c)	15.	(a)	21.	(b)	27.	(d)
4.	(c)	10.	(a)	16.	(d)	22.	(b)	28.	(c)
5.	(b)	11.	(b)	17.	(a)	23.	(a)	29.	(a)
6.	(c)	12.	(d)	18.	(c)	24.	(b)	30.	(d)

Detailed Explanations

1. **(a)**

$++i$; is pre-incremented operator i.e. value of variable i incremented by 1, the value of 'i' stored in variable C.

2. **(c)**

An opcode is the portion of a machine language instruction that specifies the operation to be performed.

3. **(d)**

An interrupt is a signal from a device attached to a computer or from a program within the computer that causes the main program that operates the computer to stop and figure out what to do next. After the interrupt signal is sensed, it may change state of interrupted process to 'blocked' and schedule another process.

4. **(c)**

Number of bits for virtual address = 32
Number of bits for pages = 4 KB = $\log_2 (2^{12}) = 12$
So, number of bits for pages = 32 – 12 = 20

Number of sets = $\frac{128}{4} = 32$

Number of bits for sets = $\log_2 (32) = 5$
So, minimum tag size = 20 – 5 = 15 bits

5. **(b)**

MDR register needed to read or written data into or from memory location.

6. **(c)**

Size of instructions 24 bits.
Starting address of the program is 300. The size of instruction is 3 byte long so that the address is always the multiple of 3 byte, next address is 600, it is also the next instruction of the program.

7. **(d)**

8. **(a)**

Interrupt vector gives the branch address of an interrupting device.

9. **(c)**

Indirect addressing → Passing array as parameter
Indexed addressing → Array implementation
Base register addressing → Relocatable code

10. **(a)**

Hazard causes a delay in the execution process of the processor.

11. **(b)**

$$\text{Speed-up} = \frac{\text{Execution time of non-pipeline}}{\text{Execution time of pipeline}}$$

$$\text{Speed up factor} = \frac{4}{1+0.42\times 3} = \frac{4}{2.26} = 1.76$$

12. **(d)**

16 bit

opcode	R_1	R_2	Immediate
	4	4	

No. of bits for register = $\lceil \log_2 10 \rceil = 4$

No. of bits for opcode = $\lceil \log_2 7 \rceil = 3$ bit

So, no. of bits for immediate operand field = 16 – (4 + 4 + 3) = 5 bits
Thus maximum possible value of immediate operand = $2^5 - 1 = 31$.

13. **(a)**
Configurations for CPU in decreasing order of operating speeds:
Hardwired control > Horizontal micro programming > Vertical Micro-programming

14. **(b)**
Load and store take 2 memory access, 1 for IF and 1 for loading/storing
= 100(IF) + 1 × 0.25 × 100 (loading/storing)
= 125 memory access.

$$\text{Avg. memory access/instruction} = \frac{125}{100} = 1.25 \text{ memory access/instruction}$$

Cycles per instruction for handling cache misses.
= (Memory access per instruction × miss rate × Cycles per miss)
= (1.25 × 0.02 × 102) = 2.55 cycles per instruction
Cycles per instruction for handling cache hits.
= Memory accesses per instruction × Hit rate × Cycles per hit
= 1.25 × 0.98 × 2 = 2.45 cycles per instruction
Effective CPI = Cycles for hits + Cycles for misses.
= 2.55 + 2.45 = 5

15. **(a)**
Interrupt vector gives the branch address of an interrupting device.

16. **(d)**

$$\begin{aligned}\text{Average CPI} &= \Sigma C_i P_i \\ &= (1 \times 0.5) + (2 \times 0.20) + (3 \times 0.23) + (4 \times 0.07) \\ &= 0.5 + 0.4 + 0.69 + 0.28 = 1.87\end{aligned}$$

17. **(a)**

$$\text{Size of memory} = 32 \text{ K bytes}$$
$$\text{Chip size} = 4 \times 2^{12} \text{ byte}$$

So, $$\text{chips needed} = \frac{2^{15} \text{ bits} \times 2^3}{2^{12} \times 4 \text{ bits}} = 16$$

18. **(c)**

|← 64 bit →|

Opcode 32 bit	Addr 1 16 bit	Addr 2 16 bit

2-address instruction

- $(2^{32} - 256)$ instruction left after 2-address instruction.
- Number of 1-address instruction = $((2^{32} - 256) \times 2^{16})$
- Number of 0-address instruction = $((2^{32} - 256) \times 2^{16} - 102) \times 2^{16}$

19. **(a)**

FIFO policy for page replacement used.

Access 100 distinct pages by taking some example: 2 3 4 5 6 7 8 9

So by loading it get

$$\begin{array}{|c|}\hline 5\\\hline 4\\\hline 3\\\hline 2\\\hline\end{array} = 4 \text{ page fault}, \quad \begin{array}{|c|}\hline 9\\\hline 8\\\hline 7\\\hline 6\\\hline\end{array} = 4 \text{ page fault}$$

and now access these page in reverse so

$$\underbrace{9\quad 8\quad 7\quad 6}_{\text{So no page fault for these}} \qquad \underbrace{5\quad 4\quad 3\quad 2}_{\text{page fault for these}}$$

$$\begin{array}{|c|}\hline 9\\\hline 8\\\hline 7\\\hline 6\\\hline\end{array} \quad \begin{array}{|c|}\hline 5\\\hline 4\\\hline 3\\\hline 2\\\hline\end{array} \; 4 \text{ page fault}$$

So, total = 4 + 4 + 4 = 12 page fault

For 8 pages = 2 × 8 – 4 = 12

So, for n pages = 2n – 4

So, for 100 pages = 2 × (100) – 4 = 196

20. **(c)**

(III) and (IV) are true.

21. **(b)**

- Group G_1 and G_2 use horizontal micro-programming.
  Total bits are : 32 + 45 = 77
- Group G_3, G_4, G_5 and G_6 are using vertical micro-programming.
  Hence, total bits are

$$= \lceil \log_2 66 \rceil + \lceil \log_2 33 \rceil + \lceil \log_2 13 \rceil + \lceil \log_2 21 \rceil$$

$$= 7 + 6 + 4 + 5 = 22$$

Total bits for control word = 77 + 22 = 99

22. (b)

	Instruction	Instruction size	Location
I_1	Load r_0, 300	2 word	2000-2003
I_2	MOV r_1, 5000	2 word	2004-2007
I_3	MOV r_2, (r_1)	1 word	2008-2009
I_4	Add r_0, r_2	1 word	2010-2011
I_5	MOV 6000, r_0	2 word	2012-2015
I_6	HALT	1 word	2016-2017

∴ Since 1 word is of 2 bytes.

If an interrupt occurs, the CPU has been halted after executing the HALT instruction, the return address 2016 is saved in the stack.

23. (a)

$$\text{Block size} = 16 \text{ bytes} = 2^4 \text{ bytes}$$

$$\text{Offset} = 4 \text{ bits}$$

$$\text{Index} = 12 \text{ bits}$$

$[\because \text{ number of lines} = 2^{12}]$

$$\text{Main memory size} = 2^{20} \text{ bytes}$$

$\Rightarrow$ Physical address = 20 bits

(4)	(12)	(4)
TAG	INDEX	OFFSET

20 bits

E	2	0	1	F
1110	0010	0000	0001	1111

∴ E is tag and 201 is line address (index).

24. (b)

Execution time for D1 pipeline:

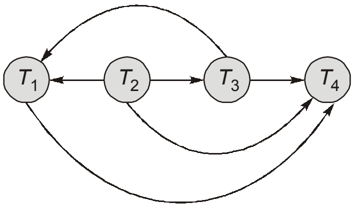
$$\begin{aligned} \text{Pipeline execution time [D1]} &= (K + n - 1) \times T_P \\ &= (100 + 5 - 1) \times T_P \\ T_P &= \max \text{ (all stages time)} \\ &= \max (3, 2, 4, 2, 3) \text{ nsec} \\ &= 4 \text{ nsec} \\ \text{[D1]} &= (100 + 5 - 1) \times 4 \text{ nsec} \\ &= (104) \times 4 \text{ nsec} \\ &= 416 \text{ nsec} \end{aligned}$$

Execution time for D2 pipeline:

$$\begin{aligned} \text{Pipeline execution time [D2]} &= (K + n - 1) \times T_P \\ &= (100 + 8 - 1) \times T_P \\ T_P &= \max \text{ (all stages time)} \\ &= \max (2, 2, 2, 2, 2, 2, 2, 2) \text{ nsec} = 2 \text{ nsec} \end{aligned}$$

$$\begin{aligned}[D2] &= (100 + 8 - 1) \times T_P \\ &= (107) \times 2 \text{ nsec} \\ &= 214 \text{ nsec} \\ \text{Time saved by design D2} &= [D2] - [D1] \\ &= (416 - 214) \text{ nsec} \\ &= 202 \text{ nsec}\end{aligned}$$

25. **(d)**



No cycle, so conflict serializable.

In given schedule no dirty read present so schedule is recoverable also.

26. **(d)**

Using call by value: In call by value, actual value are passed to formal parameter and operation applied on formal parameter so no change made to actual parameter.

Global variance i $\boxed{\cancel{0}\ \cancel{50}}$ 100
1000

main( ) j $\boxed{60}$
2000

i = 50; refer to global variable, since not declared.
call f(i, j) = call f(50, 60)

f(x, y)
x $\boxed{\cancel{50}}$ 10 y $\boxed{\cancel{60}}$ 160
3000 4000
i = 100; refer to global variable, since not declared
x = 100;
y = y + i; ⇒ y = 60 + 100 = 160

Control return

PF(i, j)
i = 100
j = 60

Using call by reference: In call by reference address of actual parameter are passed to call function and any operation performed on it reflect to actual parameter.

Global variance i $\boxed{\cancel{0}\ \cancel{50}\ \cancel{100}}$ 10
1000

main( ) j $\boxed{\cancel{60}}$ 70
2000

i = 50; refer to global variable, since not declared.
call f(&i, &j) = call f(1000, 2000)

f(*x, *y)
*x $\boxed{1000}$ *y $\boxed{2000}$
3000 4000
i = 10; refer to global variable, since not declared
*x = 10;
*y = *y + i; ⇒ *y = 60 + 10 = 70

Control return

PF(i, j)
i = 10
j = 70

27. **(d)**

28. **(c)**

Process	Arrival Time	B.T
A	0	3
B	1	6
C	4	4
D	6	2

(i) **FCFS:**

A	B	C	D

0 3 9 13 15

$$\text{Avg TAT} = \frac{3+8+9+9}{4} = \frac{29}{4} = 7.25$$

(ii) **Non-peemptive SJF:**

A	B	D	C

0 3 9 11 15

$$\text{Avg TAT} = \frac{3+8+5+11}{4} = 6.75$$

(iii) **SRTF:**

A	B	D	C	B

0 3 4 8 10 15

$$\text{Avg TAT} = \frac{3+14+4+4}{4} = \frac{25}{4} = 6.25$$

(iv) **RR:**

A	B	A	C	B	D	C	B

0 2 4 5 7 9 11 13 15

$$\text{Avg TAT} = \frac{5+14+9+5}{4} = \frac{33}{4} = 8.25$$

$\therefore$ SRTF has lowest turn around time.

29. **(a)**

Gantt chart:

P_2	P_3	P_2	P_4	P_5	P_4	P_1

0 2 5 8 10 12 30 40

Process ID	Turn Around Time
P_1	40 – 0 = 40
P_2	8 – 0 = 8
P_3	5 – 2 = 3
P_4	30 – 5 = 25
P_5	12 – 10 = 2
	Average = $\frac{78}{5}$ = 15.6

30. **(d)**

Let's take 100 instructions are $I_1, I_2, \ldots\ldots, I_{98}, I_{99}, I_{100}$.

Instruction	1	2	3	4	5	6	7	8	9	10	11	12
I_1	IF	ID	EX	MO								
I_2		IF	ID	EX	MO							
I_3			IF	ID	EX	MO						
I_4				IF	ID	EX	MO					
I_5					IF	ID	EX					
I_6						IF	ID					
I_{98}							IF	ID	EX	MO		
I_{99}								IF	ID	EX	MO	
I_{100}									IF	ID	EX	MO

$$\begin{aligned} n &= 9 \\ \text{Total time required} &= (k + n - 1)t_p \qquad [t_p = 6 + 3 = 9 \text{ ns}] \\ &= (4 + 9 - 1)9 \text{ ns} \\ &= 108 \text{ ns} \end{aligned}$$

■■■■