CLASS TEST

S.No.: 01DLSK_CS_ABC_22052026

MADE EASY
Leading Institute for IES, GATE & PSUs
Delhi | Bhopal | Hyderabad | Jaipur | Pune
Web: www.madeeasy.in | E-mail: info@madeeasy.in | Ph: 011-45124612

COMPUTER ORGANIZATION

COMPUTER SCIENCE & IT

Date of Test : 22/05/2026

ANSWER KEY ➤

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

DETAILED EXPLANATIONS

1. (d)

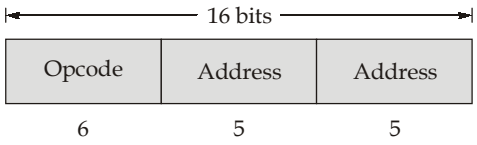
- Since, vertical microprogram encode the control signals hence a signal decoder is needed which decrease the operational speed of vertical micro-programming in comparison with horizontal micro-programming.
- Since, the control signal bits under horizontal microprogram control unit are not encoded. Hence no signal decoder is needed.

2. (c)

- Relative addressing mode and Base addressing modes are suitable for program relocation at run time.
- By first loading the operand's address in the register. Indirect addressing through registers can be used to access global variables.

3. (c)

Address format



Number of operations = $2^6 = 64$

Number of free opcodes after 2-address = 64 – 2 = 62

Number of 1 add instruction = $62 \times 2^5 = 1984$

Free opcodes after 1-address instructions = 1984 – 1024 = 960

Number of 0 add instruction = $960 \times 2^5 = 30720$

4. (b)

- More than one word are put in one cache block to explicit in the spatial locality of reference.
- By the help of virtual memory, programs can exceed from the size of primary memory, hence increases the degree of multi programming.
- Increasing RAM will result in fewer page faults.

Hence only S_2 is the correct statement.

5. (a)

The given address space is of 32 bits. Also, it is given that '11' as the MSB in those 32 bits, refer to IO devices. That means out of 32 '2'-bits are fixed, so total IO address space can be 1×2^{30}.

Similarly, since '11' are reserved hence, the total memory address space will be 3×2^{30}.

6. (c)

- Hardware detect interrupt immediately but CPU acts only after its current instruction. This is followed to ensure integrity of instructions.

7. (b)

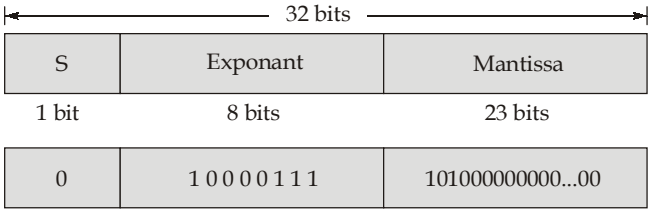
$$ET_{\text{non-pipe}} = \text{Average CPI} \times \text{Cycle time (non-pipe)}$$
$$= 5 \times 0.25\ \mu\text{sec} = 1.25\ \mu\text{sec}$$
$$ET_{\text{pipe}} = \text{Average CPI}_{\text{pipe}} \times \text{Cycle time (pipe)}$$

$$= 1 \times 0.33\ \mu\text{sec} = 0.33\ \mu\text{sec}$$

$$\text{Speed-up} = \frac{ET_{\text{non-pipe}}}{ET_{\text{pipe}}} = \frac{1.25}{0.33} = 3.78 \approx 3.7$$

8. (b)

Format of single precision floating point is



$$\begin{aligned}\text{Value} &= 1.M \times 2^{E-127} \\ &= 1.1010 \times 2^{135\,-\,127} \\ &= (1.1010)_2 \times 2^8 \\ &= 1.625 \times 2^8 \\ &= (416)_{10}\end{aligned}$$

Octal representation

8	416	
8	52	0
	6	4

Octal representation is (640).

9. (a)

Group-1 and 2 are using horizontal micro-programming,
Hence, total bits are:

$$3 + 9 = 12$$

Group-3, 4 and 5 are using vertical micro-programming,
Hence, total bits are:

$$\lceil \log_2 6 \rceil + \lceil \log_2 13 \rceil + \lceil \log_2 10 \rceil = 3 + 4 + 4 = 11$$

Total bits for control word = 12 + 11 = 23 bits

10. (c)

The device generate

$$10 \times 1000 = 10{,}000\ \text{Bps}$$

i.e. 10240 B are transmitted per second

$$1\ \text{Byte} = \frac{1}{10{,}000}\ \text{sec} = 100\ \mu\text{sec}$$

Given, that each interruption take 100 μs

$$\therefore\ \text{Fraction of processor time consumed} = \frac{100\,\mu\text{s}}{100\,\mu\text{s}} = 1$$

11. (b)

Average cycles/instruction = {(0.2 × 0) + (0.2 × 0) + (0.4 × 16) + (0.2 × 12)}
= {6.4 + 2.4} = 8.8 cycles

So, average time = 8.8 nsec

1 operand requires 8.8 nsec

Number of operands fetched in 1 sec

$$\text{Number of operands} = \frac{1\ \text{sec}}{8.8\ \text{nsec}} = 0.113636 \times 10^9\ \text{operand/sec}$$

Operand fetch rate = 113.636 million words/sec

12. (d)

One element = 8 B

Block size = 128 B

$$\therefore \quad \text{One block holds} = \frac{128}{8} = 16\ \text{elements}$$

1 row contains = 512 elements

∴ 32 block are required to carry row

Cache memory size = 32 KB

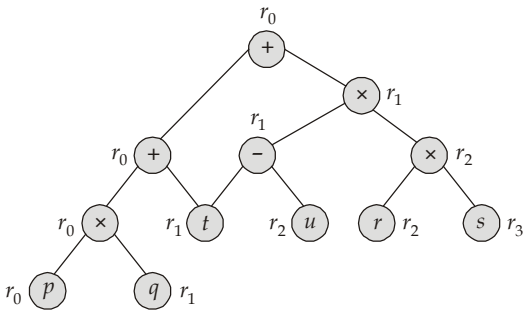
Block size = 128 B

$$\text{Number of lines} = \frac{2^{15}}{2^7} = 256$$

One row accessing = 32 miss operations

Total miss = 32 × 512 = 16384

13. (c)



Load r_0, p

Load r_1, q

MUL r_0, r_1 [store result in r_0]

Load r_1, t

Load r_2, u

ADD r_0, r_1 [store result in r_0]
SUB r_1, r_2 [store result in r_1]
Load r_2, r
Load r_3, s
MUL r_2, r_3 [store result in r_2]
MUL r_1, r_2 [store result in r_1]
ADD r_0, r_1 [store result in r_0]

So, 4 registers are required.

14. (b)

(i) It is not true, because doubling line size do not effect the number of tag bits in cache but this halves the number of lines i.e.

Main memory size = 2^{30} B
Cache memory size = 2^{20} B
Initially size of line = 256 B

10	12 bits	8 bits
Tag bits	Lines	Line offset

After doubling line size = 512 B

10	11 bits	9 bits
Tag bits	Lines	Line offset

So false.

(ii) Consider

Main memory size = 2^{30} B
Cache memory size = 2^{20} B
Initially size of line = 256 B
Initially associativity = 4 way

12 bits	10 bits	8 bits
Tag bits	Set	Line offset

After doubling associativity i.e. 8 way

13	9 bits	8 bits
Tag bits	Set	Line offset

So true.

(iii) Doubling line size decrease the compulsory misses. Since during miss more items are brought into memory in comparison to without doubling line size. Chance of miss are less.

So true.

15. (a)

Prog. IO : CPU time depends on IO speed.

i.e., 1 MB ..... 1 sec

8 B(1 word) ..... ?

$$ET_{Prog\ IO} = \frac{8B}{1\,MB}\,sec = 8\,\mu sec$$

INT-IO : CPU time depends on interface latency.

$$ET_{INT\text{-}IO} = 2\,\mu sec$$

$$S = \frac{ET_{Prog\text{-}IO}}{ET_{INT\text{-}IO}} = \frac{8}{2} = 4$$

16. (d)

With operand forwarding

	1	2	3	4	5	6	7	8	9	10
I_1	IF	ID	EX	MA	WB					
I_2		IF	ID	EX	MA	WB				
I_3			IF	ID	X	EX	MA	WB		
I_4				IF	X	ID	EX	MA	WB	
I_5						IF	ID	EX	MA	WB

10 cycles are required.

17. (b)

- Pointer are implemented using indirect addressing mode.
- Global variable are using direct addressing mode.
- Array are implemented using base index addressing mode.
- Constant are implemented using immediate addressing mode.

18. (d)

$$\begin{aligned} T_{avg} &= h_1 t_1 + (1 - h_1)h_2 (t_2 + t_1) + (1 - h_1)(1 - h_2)(t_3 + t_2 + t_1) \\ &= 0.65 \times 0.02 + 0.35 \times 0.45 \times 0.22 + 0.35 \times 0.55 \times 2.22 \\ &= 0.013 + 0.03465 + 0.42735 \\ &= 0.475 = 475\ \mu sec \end{aligned}$$

19. (b)

$$P_1 \text{ CPU time} = \frac{[1 \times 0.1 + 2 \times 0.1 + 3 \times 0.5 + 4 \times 0.3]}{1.5 \times 10^9}$$

$$= 2 \times 10^{-9}\ sec = 2\ nsec$$

$$P_2 \text{ CPU time} = \frac{[2 \times 0.1 + 2 \times 0.1 + 2 \times 0.5 + 2 \times 0.3]}{2.5 \times 10^9}$$

$$= 0.8 \times 10^{-9}\ sec = 0.8\ nsec$$

$\therefore$ P_2 is faster than P_1 processor.

20. (c)

Memory format:

TAG	SET offset	Word offset

$$\begin{aligned}\text{Word offset} &= 1 \text{ block}\\ &= 64 \text{ words}\\ &= 64 \times 16 \text{ bits}\\ &= 64 \times 2\text{B}\\ &= 128 \text{ B}\\ &= \log_2 128\\ &= 7 \text{ bits}\\ \text{CM size} &= 2^{13} \text{ B}\end{aligned}$$

$$\text{Number of lines} = \frac{2^{13}}{2^7} = 2^6$$

$$\text{Number of sets} = \frac{2^6}{2^2} = 2^4$$

$$\begin{aligned}\text{TAG} &= 21 - (7 + 4)\\ &= 21 - 11\\ &= 10 \text{ bits}\end{aligned}$$

21. (c)

We know that,

$$\text{Block number in main memory} = \frac{\text{Byte address}}{\text{Bytes per block}}$$

$$= \left\lfloor \frac{1216}{32} \right\rfloor = \lfloor 38 \rfloor = 38$$

Now, block number 38 will mapped to line number 38 mod 32 = 6

22. (b)

	CC_1	CC_2	CC_3	CC_4	CC_5	CC_6	CC_7	CC_8	CC_9	CC_{10}	CC_{11}
I_1	F	D	E	E	E	W					
I_2		F	D	///	///	E	W				
I_3			F	///	///	D	E	W			
I_4				///	///	F	D	E	E	E	W

Minimum clock cycles = 11[CC = Clock cycle]

23. (d)

Window size = Local register + (in register + out register) + Global register
= L + 2C + G
= 20 + (2 × 10) + 30
= 20 + 20 + 30 = 70

Register file size = W (L + C) + G
= 4 (20 + 10) + 30
= 4 (30) + 30 = 150

Difference = 150 – 70 = 80

24. (a)

8-way set associative

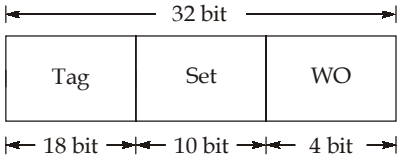
Cache memory size = 128 KB

Block size = 16 bytes

$$\text{Number of lines} = \frac{128\ \text{KB}}{16\ \text{B}} = 8\ \text{K} = 2^{13}$$

$$\text{Number of sets} = \frac{2^{13}}{2^{3}} = 2^{10} = 1\ \text{K}$$

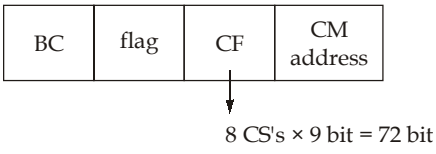
32 bit virtual address



Tag memory size = Number of lines in cache × Number of tag bits
= 2^{13} × 18 = 144 K bits

25. (b)

- Default micro-programmed control unit is vertical control unit.
- Number of control signals = 400.
- $\frac{\text{Number of bits}}{\text{Control signal}} = \log_2^{400} = 9\ \text{bit}$.
- μ instruction design with 8 control signals



26. (c)

DMA transfer character at rate of 19200 bpsec

8 bit = 1 character

So, 192000 bit = 2400 character

So, 1 sec = 2400 character

$$1\ \text{character (X)} = \frac{1}{2400} = 416.7 \times 10^{-6}\ \text{sec}$$

= 416.6 μsec

Processor fetch rate is 2 MIPS

$$1 \text{ MIPS} = 1 \text{ sec}$$

$$1 \text{ Instruction (Y)} = \frac{1}{2\times10^6} = 0.5\,\mu\text{sec}$$

$$\% \text{ slow down using DMA} = \left(\frac{Y}{X+Y}\right)\times100$$

$$= \left(\frac{0.5}{416.6+0.5}\right)\times100 = 0.11\%$$

27. (c)

The decimal number is = (– 48.625)

Binary number representation of (– 48.625)

Normalization form = 1.10000101×2^5

Mantissa field is = 10000101000000000000

Exponent field is = 5 + 127 = 132 = $(10000100)_2$

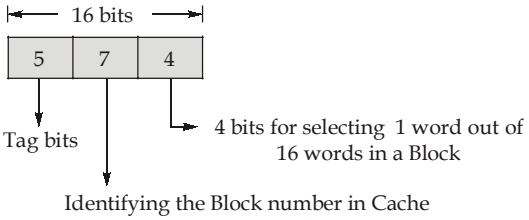
Value in given format is

Sign(s)	Exponent (E)	Mantissa (M)
1	100 00100	100 00101000000000000000
	C 2	4 2 8 0 0 0

So the hexadecimal representation is $(C2428000)_{16}$.

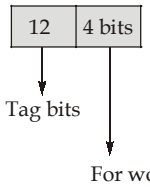
28. (d)

Direct Mapping



Tag memory = 5 × 128 blocks = 640 bits

Associative Mapping



Tag memory = 12 × 128 blocks = 1536 bits

Tag memory size difference = 1536 – 640 = 896 bits

29. (d)

Jump instruction is at address 2056 and is 4 bytes. Therefore PC will point to 2060 on execution of this instruction.

Branch target address = PC + (–15)
= 2060 – 15 = 2045

30. (b)

- The ISR does not push the value of PC and CPU register into stack but processor does this before the ISR is executed. So False.
- An Interrupt Vector is the starting address of an Interrupt Service Routine is true.
- The processor does not save these registers, as the Reset will initialize these values. So False.

■■■■