CLASS TEST

Sl. : 01 SK_EC_ABCD_040523

MADE EASY
India's Best Institute for IES, GATE & PSUs

Delhi | Bhopal | Hyderabad | Jaipur | Pune | Bhubaneswar | Kolkata

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

COMPUTER ORGANISATION

ELECTRONICS ENGINEERING

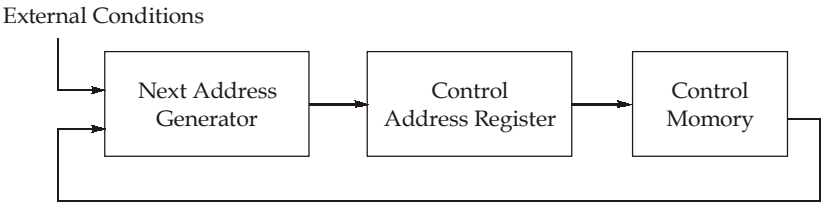
Date of Test : 04/05/2023

ANSWER KEY ➤

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

Detailed Explanations

1. (b)



2. (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.

4. (d)

Instruction pipelining is a technique that implements a form of parallelism called instruction level parallelism with a single processor. It therefore allows faster CPU throughput.

6. (b)

It is using immediate addressing mode hence the value stored in the location is added with 52.

7. (d)

This instruction is in base with offset addressing mode.

8. (c)

Number of cycles per instruction = 4

Total number of micro-instruction = 400 × 4 = 1600

Number of bit for CAR = $\lceil \log_2 (1600) \rceil \simeq 11$ bit

Branch conditions	Flag	Control Signal	CM Address
5-bit	6-bit	8-bit	11-bit

Number of bit for CDR = 5 + 6 + 8 + 11 = 30 bit

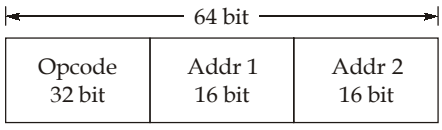
9. (c)

Based addressing mode and relative addressing mode are suitable for program relocation at runtime.

10. (a)

Interrupt vector gives the branch address of an interrupting device.

11. (c)



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 = $((2^{32} - 256) \times 2^{16} - 102) \times 2^{16}$

© Copyright: MADE EASY

12. (b)

$$L_1 \text{ miss rate} = \frac{40}{1000} = 4\%$$

L_2 miss rate (we need to take local miss rate) $= \frac{10}{40} = 25\%$

Average access time = Hit time (L_1) + Miss rate (L_1)

Where, Miss rate (L_1) = Hit time (L_2) + Miss rate (L_2) × Miss penalty

Average access time = 1 + 4% [15 cc + 25% × 200 cc]

= 1 + 0.04[15 cc + 50 cc]

= 3.6 cc

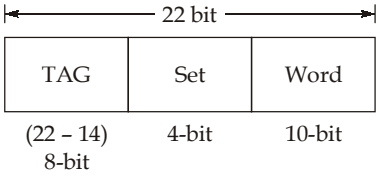
13. (b)

$$\text{Number of sets} = \frac{128}{8} = 2^4 = 4 \text{ bits}$$

Number of blocks in MM = $4k = 2^{12}$

Total MM size = $2^{12} \times 1024$ words

= $2^{12} \times 2^{10}$ words = 2^{22} words



TAG bits = Total – (Set + word) bit

= 22 – (10 + 4) = 8-bit

14. (c)

Size of cache = 32 kB

= 32×2^{10} byte

= $2^5 \times 2^{10}$ byte

= 2^{15} byte $\Rightarrow$ 15 bits

Size of tag = 32 – 15 = 17 bits

Cache indexing size = 10 bits

15. (d)

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
S_4						I_1		I_2	I_3	I_3	I_4
S_3					I_1	I_2	I_2	I_3	I_4	I_4	
S_2		I_1	I_1	I_1	I_2	I_3	I_3	I_4			
S_1	I_1	I_2	I_2		I_3	I_4	I_4				

$$\text{Throughput} = \frac{\text{Number of task completed}}{\text{Total time taken to process the tasks}} = \frac{4}{11} \text{cycles}$$

16. (a)

Configurations for CPU in decreasing order of operating speeds:

Hardwired control > Horizontal micro programming > Vertical Micro-programming

17. (b)

In horizontal micro-programming signals are present in decoded form. So number of signal bits:

$$= 25 + 30 + 38 + 27 + 20 = 140 \text{ bits}$$

In vertical micro-programming signals are present in encoded form. So, number of signal bits:

$$= \lceil \log_2(25) \rceil + \lceil \log_2(30) \rceil + \lceil \log_2(38) \rceil + \lceil \log_2(27) \rceil + \lceil \log_2(20) \rceil$$

$$= 5 + 5 + 6 + 5 + 5 = 26 \text{ bit}$$

So, total bits saved using vertical micro-programming = 140 – 26 = 114

18. (a)

Anti data dependency → Write after Read Hazard (WAR)

True data dependency → Read After Write Hazard (RAW)

	WAR Hazards		RAW Hazards (Adjacent)
1.	$I_3 - I_1$ (R_1)	1.	$I_3 - I_2$ (R_2)
2.	$I_4 - I_3$ (R_2)	2.	$I_4 - I_3$ (R_1)
		3.	$I_5 - I_4$ (R_2)

19. (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

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

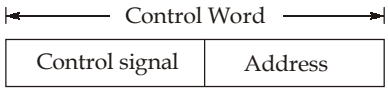
$$= 7 + 6 + 4 + 5 = 22$$

Total bits for control word = 77 + 22 = 99

20. (d)

The fetch ends with the instruction getting decoded and being placed in the IR and the PC getting incremented.

21. (b)



Number of control address = Number of instruction × Number of clock cycles

$$= 150 \times 8 = 1200$$

Number of bits for (CAR) = $\lceil \log_2 1200 \rceil = 11$ bits

Since horizontal micro-programming is used. So number of bits for control signal = 130 bits.

So, control word size in bits = 130 + 11 = 141

22. (a)

$$S = \left[(1-F) + \frac{F}{S}\right]^{-1}$$

$$S = \left[(1-0.6) + \frac{0.6}{4}\right]^{-1}$$

$$S = [0.4 + 0.15]^{-1}$$

$$S = 1.81$$

© Copyright: MADE EASY

23. (c)

$$\text{Speed-up} = \frac{\text{Pipeline depth}}{1 + \text{Branch frequency} \times \text{Branch penalty}} \geq 3$$

$$\frac{6}{1+P\times 3} \geq 3$$

$$9P + 3 \leq 6$$

$$P \leq \frac{1}{3}$$

$$P \leq 0.33$$

24. (c)

I_1	:	4000 – 4005	→	Data Transfer
I_2	:	4006 – 4010	→	ALU operation
I_3	:	4011 – 4015	→	ALU
I_4	:	4016 – 4021	→	Data transfer
I_5	:	4022 – 4026	→	ALU
I_6	:	4027 – 4031	→	ALU
I_7	:	4032 – 4034	→	Branch Instruction
I_8	:	4035 – 4039	→	ALU
I_9	:	4040 – 4045	→	Data Transfer

Return address (4032) is pushed onto the stack.

25. (c)

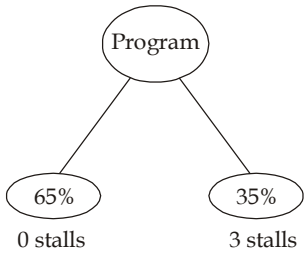
(III) and (IV) are true.

26. (c)

$$\text{Number of stalls/Instruction} = 0.65 \times 0 + 0.35 \times 3$$
$$= 1.05$$

$$\text{Speed-up (s)} = \frac{\text{No. of stages in pipelines}}{(1 + \text{No. of stalls/Instruction})}$$

$$= \frac{5}{1+1.05} = \frac{5}{2.05} = 2.44$$



27. (b)

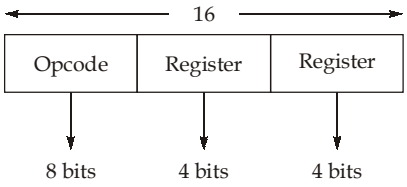
R_1	← u	.....	$(u = 1)$
R_2	← v	.....	$(v = 10)$
R_3	← w	.....	$(w = 20)$
R_1	← $R_1 + R_2$	.....	$(x = u + v)$
R_1	← $R_3 + R_1$	.....	$(y = w + x)$
R_2	← $R_3 + R_1$	.....	$(z = w + y)$
R_3	← $R_3 + R_1$	.....	$(v = w + y)$
R_1	← $R_2 + R_3$	.....	$(y = v + 2)$
R_3	← $5 + R_1$	.....	$(x = 5 + y)$
return $(R_2 + R_3)$		.....	return $(x + 5)$

hence 3 register needed only.

28. (c)

$$\begin{aligned} T_{avg} &= (1 + \#\text{ stalls/instruction}) \times \text{Cycle Time} \\ &= 1 + (10\% \times 1) + (10\% \times 2) + (5\% \times 2) \\ &= 1 + 0.1 + 0.2 + 0.1 \\ &= 1.4 \text{ cycles} \end{aligned}$$

29. (c)



Let, total number of 2-address instructions = x.

Given that number of 1-address instruction = 256.

Therefore, $(2^8 - x) \times 2^4 = 256$

$$2^8 - x = 2^4$$

$$x = 256 - 16 = 240$$

30. (d)

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

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*

$$n = 9$$

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

$$= (4 + 9 - 1)9 \text{ ns}$$

$$= 108 \text{ ns}$$

■■■■

© Copyright: MADE EASY