

# CLASS TEST

S.No. : 01 PT\_CS&IT\_A\_060519

Computer Organization



## MADE EASY

India's Best Institute for IES, GATE & PSUs

Delhi | Noida | Bhopal | Hyderabad | Jaipur | Lucknow | Indore | Pune | Bhubaneswar | Kolkata | Patna

Web: [www.madeeasy.in](http://www.madeeasy.in) | E-mail: [info@madeeasy.in](mailto:info@madeeasy.in) | Ph: 011-45124612

**CLASS TEST  
2019-2020**

**COMPUTER  
SCIENCE & IT**

**Subject : Computer Organization**

**Date of test : 06/05/2019**

### *Answer Key*

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

**DETAILED EXPLANATIONS**

1. (c)

PC holds the value of next instruction to be executed. We store the value of PC to MBR and value of stack pointer to MAR. Then store the value of PC which is available in MBR to location addressed by MAR. Atlast vector address return to the PC. This can be done in interrupt subprogram initialization.

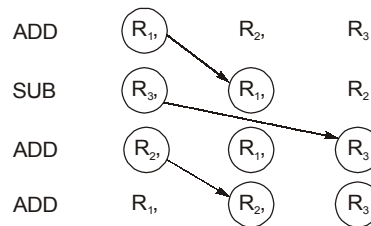
2. (c)

Execution time for pipeline =  $(k + n - 1) \times tp$   
 where  $k$  = Number of stages  
 $n$  = Number of instruction  
 $tp$  = Execution time = Max (all stages)  
 $P_1 = [8 + 500 - 1] \times 8 = 4056$   
 $P_2 = [5 + 500 - 1] \times 5 = 2520$   
 Time saved using  $P_2 = 4056 - 2520 = 1536 \text{ nsec} = 1.536 \mu\text{sec}$

3. (c)

In relative addressing mode content of the program counter is added to the address part of the instruction to get the effective address.

4. (b)



RAW (In-Out) dependency between consecutive instructions is considered only.

5. (b)

For 1 second it take  $10^9$  byte

$$\text{So for 64 kbyte it takes} = \frac{64k}{10^9} = 64 \mu\text{sec}$$

$$\text{Main memory latency} = 64 \mu\text{sec}$$

$$\text{Total time required to fetch} = 64 \mu\text{sec} + 64 \mu\text{sec} = 128 \mu\text{sec}$$

6. (a)

$S_1$ : Cache solely works on principle of locality i.e., temporal locality and spatial locality.

$S_2$ : The performance of a system depends on the indirect proportion of memory accesses satisfied by cache.

7. (a)



$$\text{Opcode bit} = \lceil \log_2 170 \rceil = 8$$

$$\text{Register address bit} = \lceil \log_2 37 \rceil = 6$$

$$\begin{aligned} \text{Memory address bit} &= 32 - (8 + 6) = 18 = 2^{18} \text{ words} = 2^{18} \times 4 \text{ bytes} \\ &= 2^{20} \text{ bytes} = 1 \text{ MB} \end{aligned}$$

8. (d)

2's Complement of -72 is 1 1 0 1 1 1 0 0 0

Append 0 at LSB and start from right end taking pair of two symbols which are encoded as.

00 → 0      10 → -1

01 → +1    11 → 0

**Booth Multiplier**

Actual		Recorded
0	0	0
0	0	0
0	0	0
1	0	-1
1	1	0
1	1	0
0	1	+1
1	0	-1
1	1	0

9. (c)

$$\begin{aligned} \text{Average CPI} &= \sum C_i I_i \\ &= 1 \times 0.5 + 2 \times 0.23 + 3 \times 0.17 + 4 \times 0.1 \\ &= 0.50 \times 0.46 + 0.51 + 0.40 = 1.87 \end{aligned}$$

10. (b)

CISC processor contain less registers and larger instruction set.

11. (d)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
IF	ID	EX	MA	WB									
	IF	ID	EX	MA	WB								
		IF	ID	S	S	S	EX	MA	WB				
			IF	ID	S	S	S	S	S	S	EX	MA	WB

S = Stall

14 cycles are required.

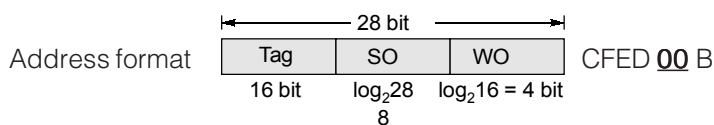
12. (c)

$$\text{Number of lines} = \frac{8K}{16} \Rightarrow 2^9$$

$$\text{Number of sets} = \frac{2^9}{2} \Rightarrow 2^8$$

Physical address size = 28 bits

2 way set associative cache.

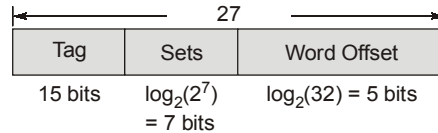


13. (b)

Main m/m size = 128 M byte =  $2^{27}$  byte = 27 bits  
 Cache memory size = 16 kbytes  
 Block size = 32 bytes

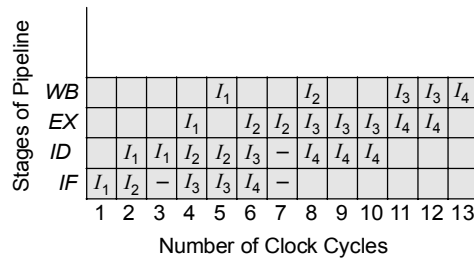
$$\text{Number of lines (N)} = \frac{16 \text{ kB} \times 1024}{32 \text{ kB}} = \frac{2^{14}}{2^5} = 2^9$$

$$\text{Number of Sets (S)} = \frac{N}{P - \text{Way}} = \frac{2^9}{2^2} = 2^7$$



$$\text{TAG bits} = 27 - (7 + 5) = 15 \text{ bits}$$

14. (b)



15. (c)

$$\text{Number of sets} = \frac{2 \times 10}{2} = 10$$

∴ 232 mod 10 = 2  
 Hence the block 232 will map to set 2.

16. (d)

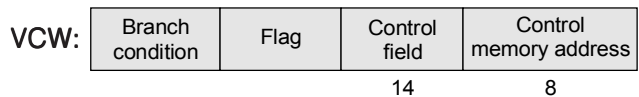
$$\begin{array}{r} 10001110 \\ 10000000 \\ \hline \text{Sum} = 100001110 \end{array}$$

Z = 0, C = 1, O = 1, S = 0

17. (a)

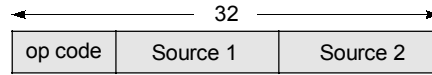
Number of bits for control signals in vertical programming:  
 $\log_2(2) + \log_2(1) + \log_2(4) + \log_2(27) + \log_2(17)$   
 $= 1 + 1 + 2 + 5 + 5 = 14 \text{ bits}$

256 CW = 8 bits

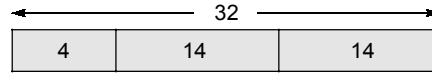


$$\begin{aligned} \text{VCW size} &= 14 + 8 = 22 \text{ bits} \\ \text{Vertical control memory size} &= 256 \times 22 \text{ bits} \\ &= \frac{256 \times 22}{8} \text{ bytes} \\ &= 704 \text{ bytes} \end{aligned}$$

18. (c)



16 K words  $\Rightarrow$  14 bits needed for addressing



Number of op-codes possible =  $2^4 = 16$

Remaining op-codes =  $16 - 12 = 4$

Number of one address instructions =  $4 \times 2^{14} = 64 \text{ K}$

19. (b)

$$\text{Speed up} = \frac{t_n}{t_p} = \frac{CPI_n \times \# \text{ instruction} \times \text{Cycle time}_n}{CPI_p \times \# \text{ instruction} \times \text{Cycle time}_p}$$

$$4 = \frac{CPI_n \times (1 + 2 + 3 + 4 + 5)}{1 \times (5)}$$

$$CPI_n = \frac{20}{15} = \frac{4}{3} = 1.33$$

20. (c)

Choice A is "register" addressing, which is supported by this architecture. Choice B is also typically covered when manufacturers speak of "register" addressing, which is supported by this architecture. Choice C is "immediate" (or "literal") addressing, which is not supported by this architecture. Choice D is "direct" (or "absolute") addressing, which is supported by this architecture. Choice E is "indirect" (or "memory indirect") addressing, which is supported by this architecture.

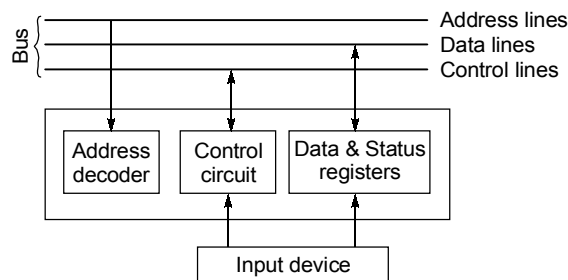
21. (b)

$$\text{Time taken by I/O device} = \frac{16 \text{ MB}}{128 \text{ kB}} = 128 \text{ sec}$$

$$\text{Percentage time CPU is busy} = \frac{128}{128 + 28} \times 100 = 82.05$$

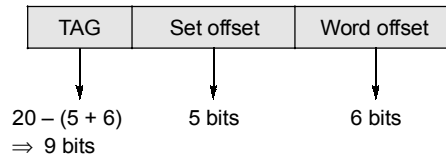
22. (d)

- (i) The address decoder enables the device to recognize its address when this address appears on the address lines.
- (ii) Control circuitry is required to coordinate I/O transfers.
- (iii) The data register holds the data being transferred to or from the processor. The status register contains information relevant to the operation of the I/O device.



23. (b)  
The required probability =  ${}^{10}C_3 (0.35)^3 (0.65)^7 = 0.252$

24. (a)  
4 way set associative  
Number of lines = 128  
Number of sets =  $\frac{128}{4} = 32$   
Block size = 64 words



A : 1111 0101 1 **010 00** 11 1111  
In addresses bit numbers (10 – 14) will decide the set number in cache memory.  
∴ A – 8, B – 16, C – 19, D – 10

25. (b)  
Average Number of stalls per instruction =  
(# misses per instruction in  $L_1$  × Hit time in  $L_2$ ) + (# misses per instruction in  $L_2$  × Miss penalty of  $L_2$ )  
2.5 memory references per instruction ⇒  $\frac{1000}{2.5}$  instructions for 1000 references = 400 instructions.  
∴ Average number of stalls per instruction

$$= \left( \frac{250}{400} \times 40 \right) + \left( \frac{120}{400} \times 250 \right)$$

$$= 25 + 75 = 100 \text{ cycles.}$$

26. (c)  
1. Since the cache line size is 8 bytes, the smallest unit of data transfer into cache from L2 cache or memory is 8 bytes. So if we have a miss for A[0], both A[0] and A[1] get fetched into cache.  
2. The cache is addressed by the lower bits of the address. However the address is byte address, and since a cache line can hold 8 bytes, the lower three bits of the address are used to address bytes inside a cache line. Since the cache is 2K bytes large, it has  $2K/8 = 256$  cache lines, which are addressed by 8 bits. Hence:  
bits 0-2 form the "offset", which is used to address inside a cache line  
bits 3 through 10 of the address form the cache line address.  
Bits 11-32 form the TAG. (assuming a 32 bit architecture)  
Now, Consider the sequence : (2 iterations of the loop )  
load A[0] → causes A[0] and A[1] at cache line 0  
load B[0] → \*also addresses cache line 0\* - so overwrites A[0] & A[1] above  
store A[0] → Nothing happens to cache (no write allocate) -> 8 bytes are written (8 is the unit of transfer)  
load A[1] → Accesses the SAME cache line as A[0] So we load A[0] and A[1] again into line 0  
load B[2] → addresses cache line 1 - load B[2] and B[3] into cache line 1  
store A[1] → Again nothing happens as above.  
So the pattern might be obvious :

At every iteration of the loop,  $B[2*i]$  accesses a new cache line, and A overwrites the cache lines every two iterations. Since the loop is 256 iterations, B will just reach cache line 255 when the loop will finish. Since A has been erasing B half as fast, we would have A in the top half of the cache and B in the bottom half. Thus the cache contains :  $A[0]-A[255]$  (In the top half) and  $B[256]- B[511]$  in the bottom half.

Also, since the cache is write through, the entries in the cache will always be the freshly written entries. Since this is write through, we have to write 256 words = 1024 bytes (all of A) back to the next level (L2 cache or memory).

Assuming that the minimum transfer from a cache to a lower level is a cache line, this translates to 2048 bytes.

27. (b)  
I/O ports are placed at addresses on bus and are accessed just like other memory location in computers that uses memory mapped I/O.
28. (a)  
Increase in the associativity leads to increase in the number of tag comparisons. Hence it leads to increase in cache access time.
29. (d)  
 $S_1$ : Separate I/O address space does not necessarily mean that I/O address lines are physically separated from the memory address lines. A special signal on the bus indicates that the requested read or write transfer is an I/O operation.  
 $S_2$ : The address decoder, the data and status register and control circuitry required to coordinate I/O transfers constitute the interface circuit (Hence true).
30. (b)  
1. For a single instruction time taken on pipeline CPU is always greater than or equal to the non-pipeline.  
2. When all stages have same delay and buffer latency is zero then for a single instruction execution time of pipeline CPU is equal to the execution time of non-pipeline CPU.

