CLASS TEST

Sl.: 03 SKCS_FGHI_16112023

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 ORGANIZATION

COMPUTER SCIENCE & IT

Date of Test : 16/11/2023

ANSWER KEY

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

DETAILED EXPLANATIONS

1. (c)

Address format

$\longleftarrow$ 16 bits $\longrightarrow$

Opcode	Address	Address
6	5	5

Number of operations = $2^6 = 64$

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

Number of 1 add instruction = $62 \times 32 = 1984$

Free opcodes = 1984 – 1024 = 960

Number of 0 add instruction = $960 \times 32 = 30720$

2. (a)

Since, it uses horizontal micro-programmed that requires 1 bit control / signal.

For 125 control signal, we need 125 bits.

Total number of micro-operation instruction = $215 \times 6 = 1290$

It requires 11 bit.

3. (c)

Multiplier	Pair with (q – 1)	Recorder
0	0	0
1	0	–1
1	1	0
0	1	+1
1	0	–1
0	1	+1
0	0	0
1	0	–1
0	1	+1
1	0	–1
1	1	0
0	1	+1

4. (b)

The address division will be given as:

11	2	3
Tag	Line	Word

Line

$$\text{73B2} \longrightarrow \underbrace{0\,1\,1\,1\,0\,0\,1\,1\,1\,0\,1}\,\underbrace{1\,0}\,\underbrace{0\,1\,0} \longrightarrow 1\,0$$

$$\text{DF45} \longrightarrow \underbrace{1\,1\,0\,1\,1\,1\,1\,1\,0\,1\,0}\,\underbrace{0\,0}\,\underbrace{1\,0\,1} \longrightarrow 0\,0$$

$$\text{8C7C} \longrightarrow \underbrace{1\,0\,0\,0\,1\,1\,0\,0\,0\,1\,1}\,\underbrace{1\,1}\,\underbrace{1\,0\,0} \longrightarrow 1\,1$$

$$\text{07E9} \longrightarrow \underbrace{0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1}\,\underbrace{0\,1}\,\underbrace{0\,0\,1} \longrightarrow 0\,1$$

5. (b)

1 clock is for 1 micro operation

8 clock means 8 micro operations

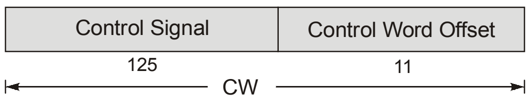
150 instructions × 8 micro operations = 1200

$\Rightarrow$ $2^{10} < 1200 < 2^{11}$

$\Rightarrow$ 11 bits are required

Control address register will use 11 bits

Size of control word (CW) = 125 + 11 = 136 bits



6. (c)

S_1 : Main disadvantage of direct mapping is that cache hit ratio decreases sharply if two or more frequently used blocks map on to same region.

S_2 : Because each and every WRITE operation is done simultaneously on both cache and main memory. WRITE THROUGH results in more cache cycles than WRITE BACK.

7. (a)

Write-After-Read Hazard :

(1) $I_4 - I_2$

(2) $I_4 - I_3$

(3) $I_5 - I_4$

Write-After-Write Hazard :

(1) $I_2 - I_1$

(2) $I_5 - I_3$

So, there are total '3' write after read hazard and '2' write after write hazard.

8. (b)

Increment value is known when the current instruction is decoded within the IR.

9. (c)

The given address space is of 32 bits. Also, it is given that "111" as the MSB in those 29 bits, refer to IO devices. That means out of 32 '3'-bits are fixed, so total IO address space can be 1×2^{29}.

Similarly, since "111" are reserved hence rest of 7 combinations represent memory location (000 to 110). So, the total memory address space will be 7×2^{29}.

10. (b)

Memory mapped I/O uses the same address bus to address both memory and I/O devices the memory and registers of the I/O devices are mapped to address values.

So, when an address is accessed by the CPU, it can depict whether the address range belongs to some I/O device or a memory location.

11. (a)

For non-pipelined processor:

For p-instruction execution time = $\frac{(p \times 5)}{5} = p$ ns

Pipelined processor:

For p-instruction execution time = $\frac{p}{3} = 0.33p$ ns

$$\text{Speed-up} = \frac{p}{0.33p} = 3.03$$

12. (b)

Memory mapped I/O uses the same address bus to address both memory and I/O devices the memory and registers of the I/O devices are mapped to address values.

So, when an address is accessed by the CPU, it may refer to a portion of physical RAM, but it can also refer to memory of I/O device.

13. (b)

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

14. (b)

$$\text{Biased exponent} = 18 + 64 = 82$$

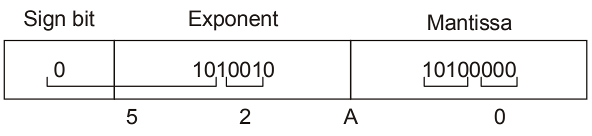
Representing 82 in binary

$$(82)_2 = (1010010)_2$$

Representing mantissa in binary

$$(0.625)_{10} = (0.10100000)$$

Floating point representation is as follows:



15. (a)

The minimum number of clock cycles can be obtained by writing its assembly code. The process in obtaining the outputs Z and R from input line X or Y via the same manner, such that the codes are not much different except there is a line code to execute store command when using input line X. The code is as follows:

	Instruction	Meaning	
I_1:	Load ACC, R	ACC ← (R)	$\{R = A_i\}$
I_2:	Inc, R	R ← (R) + 1	
I_3:	Mul ACC, R	ACC ← (ACC) × (R)	
I_4:	Store R, ACC	R ← (ACC)	

Constructing the table:

	1	2	3	4	5	6	7
I_1	F	D	E	W			
I_2		F	D	E	W		
I_3			F	D	E	W	
I_4				F	D	E	W

So, the minimum clock cycles required to complete one process is 7 clock cycles.

16. (b)

$$\begin{aligned} ET_{\text{non-pipe}} &= \text{Average CPI} \times \text{Cycle time (non-pipe)} \\ &= 5 \times 0.25\ \mu\text{sec} \\ &= 1.25\ \mu\text{sec} \\ ET_{\text{pipe}} &= \text{Average CPI}_{\text{pipe}} \times \text{Cycle time (pipe)} \\ &= 1 \times 0.33\ \mu\text{sec} \\ &= 0.33\ \mu\text{sec} \end{aligned}$$

$$\text{Speed-up} = \frac{ET_{\text{non-pipe}}}{ET_{\text{pipe}}}$$

$$= \frac{1.25}{0.33} = 3.78 \approx 3.7$$

17. (d)

Increasing the cache line size brings in more from memory when a miss occurs. If accessing a certain byte suggests that nearby bytes are likely to be accessed soon (locality), then increasing the cache line essentially prefetches those other bytes. This, in turn, forestalls a later cache miss on those other bytes. If misses occur because the cache is too small, then the designers should increase the size!

Conflict misses occur when multiple memory locations are repeatedly accessed but map to the same cache location. Consequently, when they are accessed, they keep kicking one another out of the cache. Increasing the associativity implies that each chunk of the cache is effectively doubled so that more than one memory item can rest in the same cache chunk.

18. (d)

$$\text{Main memory size} = 32768 \text{ blocks}$$
$$1 \text{ block} = 512 \text{ words}$$
$$= 32768 \times 512 \text{ words} = 2^{15} \times 2^{9} = 2^{24} \text{ words}$$

Main memory takes 24 bits.

Block size = 512 words = 2^9 words

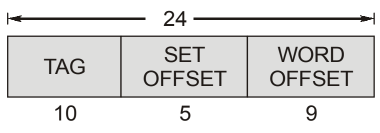
Number of bits for block size = 9 bits.

Number of blocks in set associative = 128

Number of blocks in one set = 4

$$\text{Number of sets in cache} = \frac{128}{4} = 32 = 2^5$$

Number of bits in set offset = 5 bits



Number of TAG bits = 24 – (9 + 5) = 10 bits.

19. (b)

DMA is in burst mode i.e. the DMA interface gains bus mastership prior to the start of a block transfer and maintains control of the bus until the whole block is transferred.

$$\text{Actual transfer time needed} = \frac{128\text{B}}{30\text{KBps}}$$
$$= 4.27 \text{ msec}$$

Added to this is the time to transfer bus control at the beginning and end of transfer, which is

$$500 + 500 = 1000 \text{ nsec}$$

This additional time is negligible.

So, the transfer time can be considered as 4.27 msec.

20. (b)

Considering each statement :

S_1 : Delayed control transfer, also known as delayed branching, is an attempt to cope with control hazards.

S_2 : The branch target stores the previous target address for the current branch, other algorithms for branch prediction also exist.

S_3 : For any given instruction set architecture implemented on a N-stage pipelined processor, N registers probably is not enough registers to completely prevent structural hazards involving a shortage of register hardware.

21. (b)

D_1 : Number of stages, $k = 5$

$$\tau_i = 3, 2, 4, 2, 3 \text{ ns}$$
$$i = 1 \text{ to } 5$$
$$\tau = \max(\tau_i) + d \text{ here } d \text{ is negligible}$$
$$\tau = 4 \text{ ns}$$

D_2 : Number of stages, $k = 8$

$$\tau_i = 2 \text{ ns each}$$
$$i = 1 \text{ to } 8$$
$$\tau = \max(\tau_0) = 2 \text{ nsec}$$

Number of instructions, $n = 100$

D_1: $T_k = (n + k - 1)\tau$
$$= (100 + 5 - 1) \times 4$$
$$= 104 \times 4 = 416 \text{ ns}$$

D_2: $T_k = (n + k - 1)\tau$
$$= (100 + 8 - 1) \times 2$$
$$= 107 \times 2 = 214 \text{ ns}$$

Total time saved $= 416 - 214 = 202$ ns

22. (a)

Average Memory Access time

$$= \text{Hit Time } L_1 + \text{Miss rate } L_1 \times (\text{Hit time } L_2 + \text{Miss rate } L_2 \times \text{Miss Penalty } L_2)$$

$$= 1 + 4\% \ (10{+}50\% \times 100) \left[\text{Miss rate } L_1 = \frac{40}{1000} \times 100 = 4\%\right]$$

$$= 1 + 4\% \times 60 \left[\text{Miss rate } L_2 = \frac{20}{40} \times 100 = 50\%\right]$$

$$= 3.4 \text{ clock cycles}$$

Alternate

$$T_{avg} = \text{Hit time}_{L1} + (\text{Miss rate}_{L1} * \text{Miss penality}_{L1})$$
$$\text{Miss penality}_{L1} = \text{Hit time}_{L2} + (\text{Miss rate}_{L2} * \text{Miss penality}_{L2})$$
$$= 10 \text{ cycles} + 50 \text{ cycles} = 60 \text{ cycles}$$

$$T_{avg} = \left(1 + \left(\frac{40}{1000} \times 60\right)\right) = 3.4 \text{ cycles}$$

23. (d)

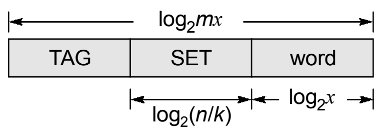
	IF	ID	OF	PD & WB
I_1:	1 m-ref	3 m-ref	2 m-ref	—
I_2:	1 m-ref	1 m-ref	1 m-ref	—
I_3:	1 m-ref	—	1 m-ref	2 cycles
I_4:	1 m-ref	—	—	2 m-ref
I_5:	1 m-ref	3 m-ref	—	2 m-ref
I_6:	1 m-ref	—	—	—

$$\text{Total time} = 80 \text{ cycles} \times 0.5 \text{ ns} = 40 \text{ ns}$$

24. (a)

$$\text{Number of sets} = n/k$$

Let there are x words per block in main memory.



$$\text{No. of TAG bits} = \log_2(mx) - \left(\log_2\frac{n}{k} + \log_2 x\right)$$

$$= \log_2(mx) - \log\frac{nx}{k} = \log_2\frac{mk}{n}$$

25. (a)

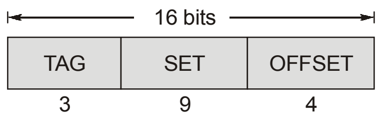
$$\text{Block size} = 16 \text{ byte} = 2^4 \text{ byte} = 4 \text{ bits}$$

$$\text{Blocks in main memory} = 2^{10}$$

$$\text{So number of sets} = \frac{2^{10}}{2^1} = 2^9 \Rightarrow 9 \text{ bits}$$

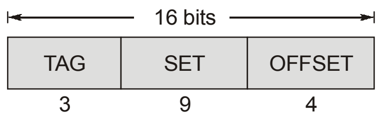
# of bits in physical address = 2^{16} byte

$\Rightarrow$ 16 bits



[E 0 1 F]

$$\underbrace{111}\,\underbrace{000000001}_{\text{SET}}\,\underbrace{1111}$$



[E 2 0 8]

$$\underbrace{111}\,\underbrace{000100000}_{\text{SET}}\,\underbrace{1000}$$

$$\text{SET value}_1 = 000000001 = \text{Decimal value} = (1)_{10}$$

$$\text{SET value}_2 = 000100000 = \text{Decimal value} = (32)_{10}$$

$$\text{Difference} = \text{SET}_2 - \text{SET}_1$$

$$= 32 - 1 = (31)_{10}$$

26. (d)

Initially cache was empty

0 is miss and block containing 0th word would have been brought into cache. Again 2 is hit means 2nd address of memory available in same block as 0th address. But again 4 miss occurs.

$\therefore$ 0th, 1st, 2nd words are surely in same block.

Since size to be power of '2' than $2^2 = 4$ bytes (Block size)

0-miss $\rightarrow$ 128-miss $\rightarrow$ 0H $\rightarrow$ 128H $\rightarrow$ 64M $\rightarrow$ 0M

64-miss affected 0 $\Rightarrow$ 64 maps in the same set as 0 $\Rightarrow$ 128 maps in the same set as zero. But the miss of 128 did not affect the presence of '0'.

$\Rightarrow$ At least 2-way set associative.

If it was more than 2-way associative then 64 would not have evicted 0.

$\Rightarrow$ Exactly 2-way set associative.

27. (c)

–63 : 11000001

Multiplier	Pair with		
1	0	(–1)	LSB
0	1	(+1)	
0	0	(0)	
0	0	(0)	
0	0	(0)	
0	0	(0)	
1	0	(–1)	
1	1	(0)	MSB

$\boxed{0-10000+1-1}$

28. (c)

Memory format:

TAG	SET offset	Word offset

$$\begin{aligned}
\text{Word offset} &= 1 \text{ block} \\
&= 64 \text{ words} \\
&= 64 \times 16 \text{ bits} \\
&= 64 \times 2\text{B} \\
&= 128 \text{ B} \\
&= \log_2 128 \\
&= 7 \text{ bits} \\
\text{CM size} &= 2^{13} \text{ B} \\
\text{Number of lines} &= \frac{2^{13}}{2^7} = 2^6 \\
\text{Number of sets} &= \frac{2^6}{2^2} = 2^4 \\
\text{TAG} &= 21 - (7 + 4) \\
&= 21 - 11 \\
&= 10 \text{ bits}
\end{aligned}$$

29. (a)

$$T_1\text{(Effective Access Time)} = 20 + (0.08)(60) = 24.8 \text{ ns}$$

If size after changing the size is 2 times

$$\text{Miss rate} = 0.08 - 0.08 \times 0.3 = 0.056$$

Since cache size is 4 times then again miss rate reduces to

$$\text{Miss rate} = 0.056 - 0.056 \times 0.3 = 0.0392$$

$$T_2\text{(Effective Access Time)} = 20 + 60 \times 0.0392 = 22.352 \text{ ns}$$

$$\text{Expected improvement} = \frac{(24.8 - 22.352)}{24.8} = 0.0987$$

$$\text{Expected improvement (\%)} = 0.0987 \times 100 = 9.87$$

30. (a)

Prog. IO : CPU time depends on IO speed.

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

8 B(1 word) .... ?

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

INT-IO : CPU time depends on interface latency.

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

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

■■■■