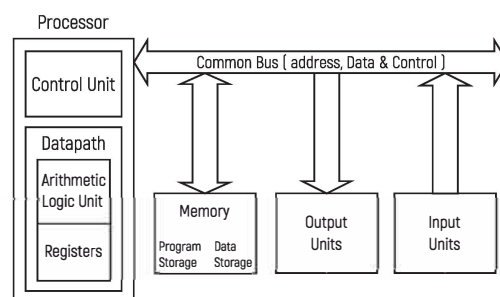


## COMPUTER SCIENCE & INFORMATION TECHNOLOGY

### Computer Organization

(Text Book : Theory with worked out Examples  
and Practice Questions)



## 1. Computer Arithmetic

01. Ans: (b)

Sol: 128 64 32 16 8 4 2 1  
 1 1 0 0 0 1 1 1  
 0 -1 0 0 +1 0 0 -1

02. Ans: (d)

Sol: Sign extension is used for converting smaller size signed data to larger size by padding the sign bit to left.

03. Ans: (a)

Sol:

1111100001111011000  
 Guard C 3 D 8  
 bits

04. Ans: (d)

Sol: In non normalized form,  $0.239 \times 2^{13}$  is

$$0.00111101_2 \times 2^{13}$$

$$e = 13, b = 64, \text{ so } E = 77 = 1001101$$

$$M = 00111101$$

$$0100110100111101 = 4D3D$$

$$S \underbrace{\hspace{1.5cm}}_E \underbrace{\hspace{1.5cm}}_M$$

05. Ans: (d)

Sol: Implicit Normalized form is  $1.11101 \times 2^{10}$

$$M = 11101000, e = 10 \quad b = 64$$

$$E = 74 = 1001010$$

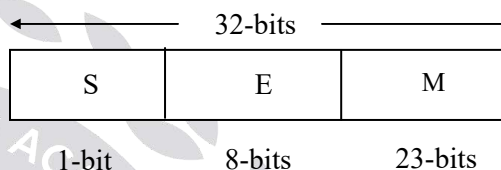
$$0100 \ 1010 \ 11101 \ 000 = 4AE8$$

$$S \underbrace{\hspace{1.5cm}}_E \underbrace{\hspace{1.5cm}}_M$$

06. Ans: (d)

Sol:  $R1 = 0x42200000 = 0100 \ 0010 \ 0010 \ 0000$   
 $0000 \dots 0$

In single precision floating point format (IEEE-754) a number is represented as follows:



From above value  $R1 \ S = 0$  (+ve value)

$$E = 10000100 = 132$$

$$M = 0100 \dots 0$$

$$\begin{aligned} \text{value of } R1 &= +(1.01 \times 2^{132-127}) \text{ bias} = 127 \\ &= +1.01 \times 2^5 = +101000 \\ &= +40 \end{aligned}$$

$$R2 = 0xC1200000 = 1100 \ 0001 \ 0010 \ 0000 \dots 0$$

same as above format of number  $S = 1$  (-ve value)

$$E = 10000010 = 130$$

$$M = 010 \dots 0$$

$$\begin{aligned} \text{Value of } R2 &= -(1.01 \times 2^{130-127}) \\ &= -(1.01 \times 2^3) \\ &= -(1010) \\ &= -10 \end{aligned}$$

$$R3 = \frac{R1}{R2} = \frac{+40}{-10} = -4 = -(100)_2$$

Implicit normalization  $\Rightarrow -(1.00 \times 2^2)$

$$E = 2 + 127 = 129 = 10000001$$

$$M = 000 \dots 0$$

$$S = 1 \text{ (-ve)}$$

32-bit value in IEEE-754 single precision

= 11000000100000....0

Hexadecimal  $\Rightarrow$  0xC0800000

**07. Ans: (b)**

**Sol:** In given data EX-OR gate is not available so for one EX-OR gate 2 level network is needed because complemented and un-complemented inputs are available.

So, total delay =  $2 + 2 + 2 = 6$

In the above, first 2 is for first EX-OR gate

Second 2 is for CLG

Third 2 is for 2<sup>nd</sup> EX-OR gate

**08. Ans: (b)**

**Sol:** Four number of carries are needed to design CLG

For  $C_1$ , one AND gate + one OR gate

For  $C_2$ , two AND gates + one OR gate

For  $C_3$ , three AND gates + one OR gate

For  $C_4$ , four AND gates + one OR gate are required

So total **ten** AND gates are needed and **four** OR gates are needed

**09. Ans: (b) & (c)**

**Sol:** All 32 bit '0's represent +0 but not  $\pm 0$ ;  
 Option 'a' is false but 'b' is True, Option 'c' is True because it is used to represent  $\pm 0$ ,  $\pm \infty$  and NAN.

**10. Ans: (d)**

**Sol:** Integer size is n bits.

Type of data is unsigned

Fraction part size = f bits

Integer part size =  $(n - f)$  bits

Range of the Decimal values for only integer part is  $(0 \text{ to } 2^i - 1)$

Range of the Decimal values for only fraction part is  $(0 \text{ to } (1 - 2^{-f}))$

Range of the Real number is

$$0 \text{ to } (2^i - 1 + 1 - 2^{-f}) \\ = 0 \text{ to } (2^i - 2^{-f})$$

## 2. Memory Organization

### Memory Basics

**01. Ans: (d)**

**Sol:** A memory has 16 bit Address i.e. it's

Starting Address = 0000H

Ending Address = FFFFH

$\therefore$  Maximum number of memory locations it can address is  $2^{16} = 64K$

$$= 64 \times 1024$$

$$= 65536.$$

**02. Ans: (d)**

**Sol:** Number of memory locations  $\times$  size of one location.

**03. Ans: (c)**

**Sol:** DRAM needs refresh logic CKT to avoid the discharge of capacitor but capacitor is not needed in SRAM.

**04. Ans: (b)**

**Sol:**  $2048 \times 8 = 2^{11} \times 8$

$\therefore$  Number of Address lines = 11

**05. Ans: (a)**

**Sol:** Number of chips needed

$$= \frac{\text{Target size}}{\text{Basic size}}$$

$$= \frac{64 \times 8}{16 \times 4}$$

$$= 8$$

**06. Ans: (c)**

**Sol:**  $T = 200 \text{ ns}$

$\therefore$  word frequency

$$= \frac{1}{200 \times 10^{-9}} = 5 \times 10^6 \text{ words/sec}$$

**07. Ans: (a)**

**Sol:** ROM is used for function table with large size because after program; ROM content is not possible to destroy and design cost is cheap.

**08. Ans: (c)**

**Sol:** Remain same because it is non-volatile memory

**09. Ans: (d)**

**Sol:**  $16 * 16 * 8 \text{ bits} = 2 \text{ K bits}$

**10. Ans: (d)**

**Sol:** Address size is 16 bit and data size is also 16 bit so memory space =  $2^{16} \times 16 = 64 \text{ K} \times 16$ .

### Cache Concept

**01. Ans: (c)**

**Sol:** Memory size =  $4 \text{ K} \times 16 = 2^{12} \times 16$

$\therefore$  Number of Address lines = 12

Data lines = 16

**02. Ans: 31**

**Sol:** Word size = 16 bit,

So memory size =  $2^{31} \text{ words} = 2^{32} \text{ bytes}$   
 $= 4 \text{ GB}$

$\therefore$  31 address bits are needed.

**03. Ans: 14**

**Sol:**  $HC + (1 - H)M = (0.8 \times 5) + (0.2 \times 50)$   
 $= 14 \text{ ns}$

**04. Ans: (b)**

**Sol:** Number of cache blocks =  $2c$

Associativity = 2

Mapping expression is  $k \text{ mod } S$

Where,  $k = \text{M.M block Number}$  and

$S = \text{Number of cache sets}$

Number of Cache sets

$$= \frac{\text{Number of cache blocks}}{\text{Associativity}}$$

$$= \frac{2c}{2} = c$$

$\therefore k \text{ mod } c$

**05. Ans: (b)**

**Sol:** Physical Address formats for the given

Block number and Tag number 6 are

Tag      Block   word   word Ranges

1 0      00      xxxx→128 to 147

1 0      01      xxxx→144 to 159

0 0      10      xxxx→32 to 47

0 1      11      xxxx→112 to 127

128 64	32 16	8 4 2 1
--------	-------	---------

∴ 150 and 132 are available

**06. Ans: (c)**

**Sol:** The cache and main memory are divided into blocks of 64 bytes each. The direct mapped cache consists of 32 blocks (mod block i/32). The array is stored from main memory locations 1100 H. The array is placed in MM from 68 Block onwards. The total array consists of 2500 bytes, so they require a total of 40 blocks. In the cache all the 32 blocks are filled and the remaining 8 blocks are replacing the previous blocks. A total of 40 data misses will occur during first access. During the second access once again the 16 blocks are replaced for conflict misses, so 16 cache misses occur.

Total numbers of cache misses

$$= 40 + 16 = 56$$

**07. Ans: (a)**

**Sol:** The lines 4 to 11 gets conflict misses frequently.

**08. Ans: (a)**

**Sol:**

Tag   Line offset   Word offset

1110	0010000000001	1111
------	---------------	------

$$\text{Tag} = E_{16} \quad \text{line} = 201_{16}$$

**09. Ans: (d)**

**Sol:** Direct-Mapping

Total Blocks = 256

Number of Tag bits = 19

$$\begin{aligned} \text{Tag Directory size} &= (19+1+1) \times 256 \\ &= 5376 \end{aligned}$$

**10. Ans: 20**

**Sol:** Associativity = 4

Cache Size = 16 KB

Block size = 8 words = 32 bytes

Word size = 32 bits

∴ Number of cache blocks

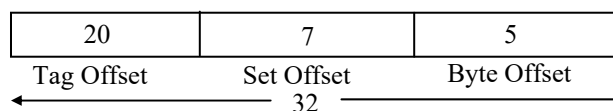
$$= \frac{16\text{KB}}{32\text{B}} = \frac{2^{14}\text{B}}{2^5\text{B}} = 2^9$$

$$\therefore \text{Number of cache sets} = \frac{2^9}{2^2} = 2^7$$

$$\therefore \text{Block size} = 8 \times 4 \text{ Bytes} = 32 \text{ B} = 2^5 \text{ B}$$

Physical Address size = 32 bits

Physical Address format:



**11. Ans: (b)**
**Sol:**

Tag	set	word
6	6	4
← 16 →		

 $\therefore$  Number of tag bits = 6

**12. Ans: (a)**
**Sol:** Physical Address = 32 bits

 Cache size = 256 KB =  $2^{18}$  B

Associativity = 4

Block size = 16 B

$$\begin{aligned} \text{Number of cache blocks} &= 2^{18} \text{ B} / 2^4 \text{ B} \\ &= 2^{14} = 16 \text{ K} \end{aligned}$$

$$\text{Number of cache sets} = \frac{2^{14}}{4} = 2^{12}$$

16	12	4
← 32 →		

**13. Ans: (c)**
**Sol:** Number of tag comparators needed

 $= \text{Associativity} = 4$ 

Size of each comparator

 $= \text{number of tag bits} = 16$ 
**14. Ans: (d)**

**Sol:** If associativity is doubled, then number of tag bits will be increased and set offset size is reduced and size of MUX is directly proportional to associativity only Physical address size and Data bus size are not altered.

**15. Ans: (c)**
**Sol:** Number of cache blocks =  $\frac{256 \text{ KB}}{32 \text{ B}} = 2^{13}$ 

Associativity = 4

$$\text{Number of Sets} = \frac{2^{13}}{4} = 2^{11}$$
 $\therefore$  Tag size =  $32 - 11 - 5 = 16$ .

**16. Ans: (a)**
**Sol:** Number of blocks =  $256 \text{ KB} / 32 \text{ B} = 8 \text{ K}$ 

$$\begin{aligned} \text{Number of Sets with 4-way set-associative} \\ &= 8 \text{ K} / 4 = 2 \text{ K} \end{aligned}$$
 $8 \text{ K} (16+1+2+1)\text{-bits} = 160 \text{ Kbits}$ 
**17. Ans: (d)**
**Sol:** The cache consists of 4 sets with each set consists of 4 blocks.

 Set 0 contains the blocks with  
 $\lfloor \text{main memory block} / 4 \rfloor = 0$ 

 Set 1 contains the blocks with  
 $\lfloor \text{main memory block} / 4 \rfloor = 1$ 

 Set 2 contains the blocks with  
 $\lfloor \text{main memory block} / 4 \rfloor = 2$ 

 Set 3 contains the blocks with  
 $\lfloor \text{main memory block} / 4 \rfloor = 3$ 

Set 0: 0, 4, 8, 16, 48, 32, 92

Set 1: 1, 129, 73

Set 2:

Set 3: 255, 3, 159, 63, 155

So 216 will not be in cache if LRU is used.

**18. Ans: (b)**
**Sol:** Cache accepts only 8 blocks and it uses LRU.

H = Hit, M = Miss Block arrival Address

0	<del>4</del>	45
1	<del>2</del>	2
2		2
3		8
4	<del>19</del>	3
5	<del>6</del>	7
6		1
7		35

Cache Block number 5 consists 7

**19. Ans: (c)**

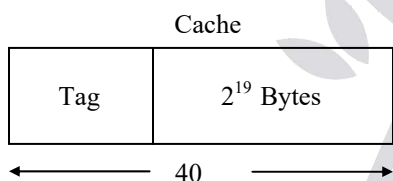
**Sol:** All blocks are mapped to set 0 only, but each set permits only 2 blocks

Total Number of misses = 4

8	12	0	12	8
M	M	M	H	M

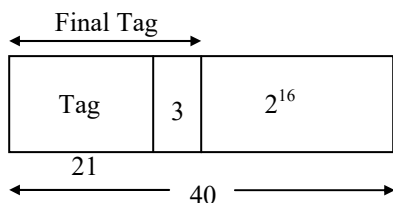
**20. Ans: 24**

**Sol:**



It uses 8 way set associative

∴ Tag size = 24 bits.



**21. Ans: (b)**

**Sol:** Number of bits required for addressing a byte in physical memory = P

Number of bits required for addressing a byte in cache = N

In Direct Map, Tag field size = P – N

In K-way Block Set Associative mapping,

Tag field size = (P – N) + log<sub>2</sub> K

**22. Ans: 76**

**Sol:** When a block is referred first time in cache memory, it is known as compulsory miss and it will be loaded in the cache memory, when a same block is referred in future i.e. 2<sup>nd</sup> time onwards; then it is considered as conflict miss.

For first iteration, 4 conflict misses and for each iteration from 2<sup>nd</sup> time onwards 8 conflict misses occur.

Total number of conflict misses occur for 10 times = 4 + (8 × 9) = 76.

**Conflict misses in first iteration**

0, 128, 256, 128, (0), 128, (256), 128, 1, 129, 257, 129, (1), 129, (257), 129.

Total 4 times

**Conflict misses in second time iteration**

(0), 128, (256), 128, (0), 128, (256), 128, (1), 129, (257), 129, (1), 129, (257), 129,

Total 8 times

**23. Ans: (a)**

**Sol:** First cache organization: 32 KB with 2-way set associative cache.

The size of address = 32 bits.

Tag (18-bits)	Set offset (9-bits)	Word offset (5-bits)
------------------	------------------------	-------------------------

Multiplexer latency = 0.6 ns and k-bit comparator latency is (K/10) ns

The hit latency of this cache =  $0.6 + (18/10)$   
 $= 2.4$  ns.

**24. Ans: (d)**

**Sol:** Second cache organization: 32 KB with direct mapped cache.

The size of address = 32 bits.

Tag (17-bits)	Set offset (10-bits)	Word offset (5-bits)
------------------	-------------------------	-------------------------

Only one TAG comparator & no multiplexer

The hit latency of this cache =  $(17/10)$   
 $= 1.7$  ns.

**25. Ans: (a), (b) & (c)**

**Sol:** Option (d) is false because Associative memory is costliest memory, hence it is used to design Tag memory in cache design but not for Main memory design.

**26. Ans: 1.68**

**Sol:** Time taken for fetching instructions to execute:

$$90 \times 1 \text{ ns} + 10 \times 5 \text{ ns} = 140 \text{ ns}$$

Time taken by executing instructions:

For memory operand read operations.

$$= \text{hit} + \text{miss} = 54 \times 1 \text{ ns} + 6 \times 5 \text{ ns} \\ = 54 \text{ ns} + 30 \text{ ns} = 84 \text{ ns}$$

For memory operand write operation

$$= \text{hit} + \text{miss} = 36 \times 2 \text{ ns} + 4 \times 10 \text{ ns} \\ = 72 \text{ ns} + 40 \text{ ns} \\ = 112 \text{ ns}$$

Total time taken for 100 instructions  
 $(140 + 84 + 112) \text{ ns} = 336 \text{ ns}$

Total number of memory references for fetching the instructions and operand read, write operations =  $100 + 60 + 40 = 200$

The average access time is

$$= \frac{336}{200} \text{ ns} = 1.68 \text{ ns}$$

**27. Ans: 4.72**

**Sol:** Given information

I-cache RD time = 2 ns

H = 0.8 = Instruction cache

'D' cache RD time = 2 ns

H = 0.9 = Data cache

L<sub>2</sub> cache RD time = 8 ns, H = 0.9

Main memory access time = 90 ns, H = 1

In a program 60% memory reads are used for instruction fetch (from I-cache) and remaining 40% memory reads are used for Data Read (from D- cache)

T<sub>avg</sub> (for instruction fetch)

$$= H_1 T_1 + (1 - H_1) H_2 (T_2 + T_1) + (1 - H_1) \\ (1 - H_2) (T_1 + T_2 + T_M) \\ = (0.8 \times 2) + ((1 - 0.8) \times (0.9) (8 + 2)) \\ + (1 - 0.8) (1 - 0.9) (90 + 8 + 2) \\ = (1.6 + 1.8 + 2) \text{ ns} = 5.4 \text{ ns}$$

Average read access is 60% i.e

$$5.4 \times 0.6 = 3.24 \text{ ns}$$

Data Read time

$$\begin{aligned}
 &= (0.9 \times 2) + ((1 - 0.9) \times 0.9 \times (8 + 2)) \\
 &\quad + (1 - 0.9) (1 - 0.9) (90 + 8 + 2) \\
 &= 1.8 + 0.9 \text{ ns} + 1 \text{ ns} = 3.7 \text{ ns}
 \end{aligned}$$

Average data read is 40%,  $0.4 \times 3.7 \text{ ns}$   
 $= 1.48 \text{ ns}$

Total time for memory access =  $3.24 + 1.48$   
 $= 4.72 \text{ ns}$

**28. Ans: 160**

**Sol:** cache size = 8 words

word size = 4B

cache size =  $8 \times 4\text{B} = 32 \text{ B}$

Memory clock rate = 60 – MHz

Memory cycle time =  $\frac{1}{60\text{MHz}}$   
 $= \frac{1}{60 \times 10^6} \text{ seconds}$

No. of cycles needed to transfer 1 block  
 (8 words)

$\Rightarrow$  1 cycle for address  
 $\quad + 3 \text{ cycles to fetch 8 words}$   
 $\quad + 8 \times 1 = 8 \text{ cycles to transmit}$

$\Rightarrow 12 - \text{cycles}$

Time required to access and transfer 8 words

(32B) from memory =  $12 \times \frac{1}{60 \times 10^6} \text{ seconds}$   
 $= \frac{1}{5 \times 10^6} \text{ seconds}$

In  $\frac{1}{5 \times 10^6}$  second amount of data accessed = 32 bytes

In 1 second amount of data accessed

$$\begin{aligned}
 &= \frac{32\text{B}}{\frac{1}{5 \times 10^6} \text{ seconds}} \\
 &= 32 \times 5 \times 10^6 \text{ Bytes / second} \\
 &= 160 \times 10^6 \text{ Bytes / second}
 \end{aligned}$$

**29. Ans: (a)**

**Sol:** Word size = 32-bits = 4Bytes

Main memory size = 16MB =  $16\text{M} \times 1\text{B} = 2^{24} \times 1\text{B}$

Main memory address = 24-bits

Cache size = 64kb

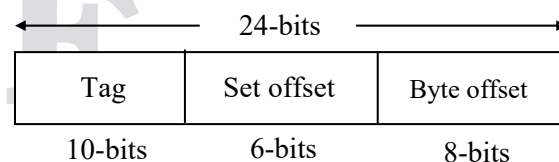
Block size = 256 Bytes =  $2^8 \text{ Bytes} \Rightarrow \text{Byte offset}$   
 $= 8\text{-bits}$

Number of blocks in cache =  $\frac{64\text{kb}}{256\text{B}} = 256$

Number of sets in cache =  $\frac{256}{4} = 64 = 2^6$

$\Rightarrow \text{set offset} = 6\text{-bits}$

In set associative mapping, main memory address is divided into 3 fields as follows:



A1= 0x42C8A4 =	0100001011	001000	10100100
----------------	------------	--------	----------

A2 = 0x546888 =	0101010001	101000	10001000
-----------------	------------	--------	----------

A3 = 0x6A289e=	0110101000	101000	10011100
----------------	------------	--------	----------

A4 = 0x5E4880=	0101111001	001000	10000000
----------------	------------	--------	----------

A1 and A4 maps to same set

A2 and A3 maps to same set

### 30. Ans: 59 to 60

**Sol:** Number of rows in a chip =  $2^{14} = 16384$

One refresh time =  $50 \times 10^{-9}$  sec

Chip refresh time =  $16384 \times 50 \times 10^{-9}$  sec  
 $= 819200 \times 10^{-9}$  sec  
 $= 0.8192$  msec

Given refresh period = 2 msec

Amount of time required for RD/WR operation

$$= 2 - 0.8192$$

$$= 1.1808 \text{ msec}$$

Amount of time used for RD/WR operation in percentage

$$= \frac{1.1808 \text{ msec}}{2 \text{ msec}} \times 100 = 59.04$$

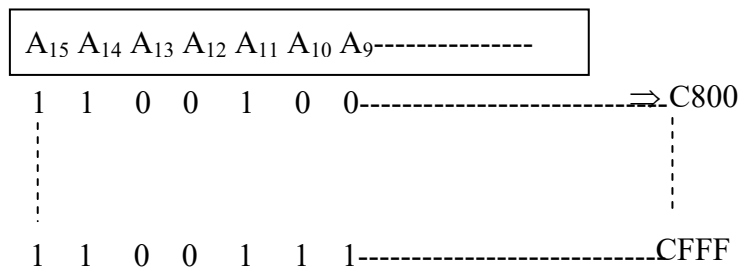
$\therefore$  Closest integer value = 59

### 31. Ans: (a)

**Sol:** Based on given system design  $A_{13}$  and  $A_{12}$  should be 0 (zero) ;

$A_{15}$   $A_{14}$  and  $A_{11}$  should be '1' (one); to enable chip select (CS).

Hence, address range will differ for address bits  $A_{10}$  to  $A_0$ .



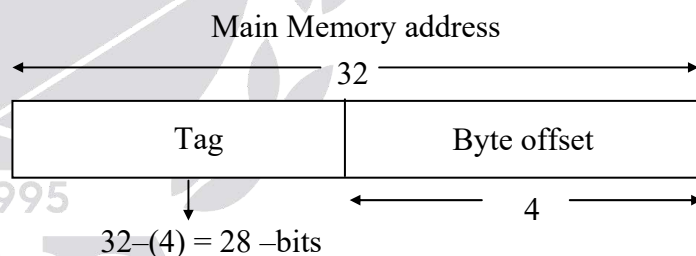
### 32. Ans: (d)

**Sol:** Cache size = 16 KB

Block Size = 16B =  $2^4$  B  $\Rightarrow$  Block offset  
 $= 4$  - bits

Main memory address = 32 - bits

In fully associative cache main memory address format



In fully associative cache, there is no any index.

Hence,

Tag = 28 bits

Index = 0 bits

### 3. Secondary Memories

**01. Ans: (P = 12.5), (Q = 2500000)**

**Sol:** T = 200, RPM = 2400, RPS = 40

Track capacity = 62500 bits

One revolution time is 25 ms

∴ Average latency time = 12.5 ms

∴ P = 12.5 ms

Q = Data Transfer rate = no. of bits/sec.

In one second it can complete 40 tracks.

∴ Q = 40 × 62500 bits/sec

= 2500000 bits/sec

**02. Ans: (d)**

**Sol:** Transfer rate = 10 K.K bytes/sec

For 10K bytes it takes 1 msec

For 20K bytes it takes 2 ms

CPU frequency is 600 MHz,

$$T = \frac{10^{-6}}{600} = \frac{10}{6} \text{ ns}$$

For initializing it takes 300 clk

$$= 500 \text{ ns } \left( \frac{10}{6} \times 300 \right)$$

For completing 900 clk = 1500ns  $\left( \frac{10}{6} \times 900 \right)$

Total CPU time is 2000ns = 2 μs

Processor consumes 2 μs for each 2 msec.

∴ In percentage it is 0.1%

**03. Ans: (b)**

**Sol:** Given seek time = 10 ms

RPM = 6000, RPS = 100.

One revolution takes 10 ms

∴ Average rotational Delay = 5 ms

Transfer delay is neglected.

$$\begin{aligned} T_{\text{access/one library}} &= t_{\text{seek}} + t_{\text{rotational}} + t_{\text{transfer}} \\ &= 10 \text{ ms} + 5 \text{ ms} + 0 \\ &= 15 \text{ ms} \end{aligned}$$

So, for 100 libraries loading;

it takes (15 ms × 100) = 1500 ms

= 1.5 sec

**04. Ans: 14020**

**Sol:** Seek time = 4 milli seconds per each sector

Reading

RPM = 10000 i.e.,

1 Track rotation time = 6 ms

∴ Average rotational delay is 3 ms.

One Track has 600 sectors.

So, one sector transfer time is

$$\frac{6 \text{ ms}}{600} = 0.01 \text{ ms}$$

$$\begin{aligned} \therefore \text{One sector Access time} &= 0.01 + 4 + 3 \\ &= 7.01 \text{ ms} \end{aligned}$$

So, 2000 sectors time = 2000 × 7.01

= 14,020 ms

**05. Ans: 6.1**

**Sol:** Transfer rate =  $50 \times 10^6$  Bytes/sec

So, 0.5 KB takes 0.1 ms

RPM = 15000; RPS = 250

1 rotation takes 4 milliseconds

Average rotational delay is 2 ms;

Seek time is 4 ms

$$\begin{aligned} \therefore \text{Average time} &= \text{transfer time} + \text{seek time} \\ &\quad + \text{rotational delay} \\ &= 0.1 \text{ ms} + 4 \text{ ms} + 2 \text{ ms} \\ &= 6.1 \text{ ms} \end{aligned}$$

**06. Ans: (c)**

**Sol:** The address (400, 16, 29) corresponds to the Sector number

$$\begin{aligned} &= (\text{cylinder number} \times \text{number of sectors/cylinder}) \\ &\quad + (\text{surface number} \times \text{number of sectors/track}) \\ &\quad + \text{Present sector number} \\ &= (400 \times 20 \times 63) + (16 \times 63) + 29 \\ &= 505037 \end{aligned}$$

**07. Ans: (c)**

**Sol:**  $\frac{1039}{63 \times 20}$  : 0<sup>th</sup> cylinder

$\frac{1039}{63}$  : 16<sup>th</sup> surface and remainder gives  
sector number: 31 (0, 16, 31)

**08. Ans: (d)**

**Sol:** Size of the data to be transferred = 42,797 KB

One sector capacity = 512 B

$\therefore$  Number of Sectors to store 42797 KB

$$= \frac{42797 \times 1024}{512} = 85,594$$

One cylinder has  $64 \times 16 = 1024$  sectors

83 cylinders can store  $83 \times 1024$

= 84,992 sectors

$\therefore$  Remaining number of sectors = 602

602 sectors occupy more than half of one cylinder capacity

But the given cylinder has started with

<1200, 9, 40> means more than half of that cylinder, so next cylinder is also needed for storing complete data.

$\therefore$  Last cylinder number

$$\begin{aligned} &= 1200 + 83 + \text{next one} \\ &= 1284 \end{aligned}$$

**09. Ans: (a), (b) & (d)**

**Sol:** Option (c) is false because DVD memory is cheaper than flash memory

## 4. CPU Organization

**01. Ans: (c)**

**Sol:** Stack works on LIFO.

**02. Ans: (d)**

**Sol:** R1  $\rightarrow$  M [100]

M[100]  $\rightarrow$  R2

M[100]  $\rightarrow$  R3

The above instructions are used for transferring  $R_1$  content to  $R_2$  and  $R_3$  through memory address 100.

So, option (d) is correct.

**03. Ans: (a)**

**Sol:** Address field in the instruction is used to specify Memory Address or One of the processor Register Address.

For example to specify  $R_5$  in a processor which is having 16 bit Register from  $R_0$  to  $R_{15}$ , it's Address field is '0101' and for implied Register; no address is specified in the instruction.

**04. Ans: (d)**

**Sol:** Max one address instruction =  $2^6 = 64$   
But number of one address instructions used = 32.

Max Number of zero address instructions  
=  $32 \times 128 = 4096$

**05. Ans: (c)**

**Sol:** If 63 one address instructions are used then  
Number of zero address instructions  
=  $(64 - 63) \times 128 = 128$

**06. Ans: (d)**

**Sol:** Max. number of two address instructions =  $2^4$ .

When it uses only 'n', two address instructions then remaining  $(2^4 - n)$  with '6' bit combinations are used for one address instructions.

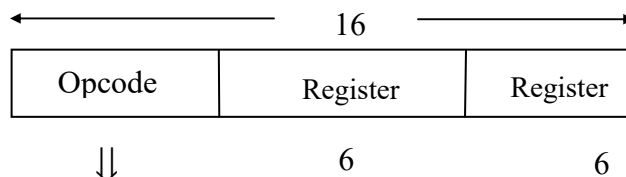
$\therefore$  Max. number of one address instructions:  
 $(2^4 - n) \times 2^6$

**07. Ans: 14**

**Sol:** Number of registers = 64

Register name size =  $\log_2 64 = 6$ -bits

Type R-Instruction:

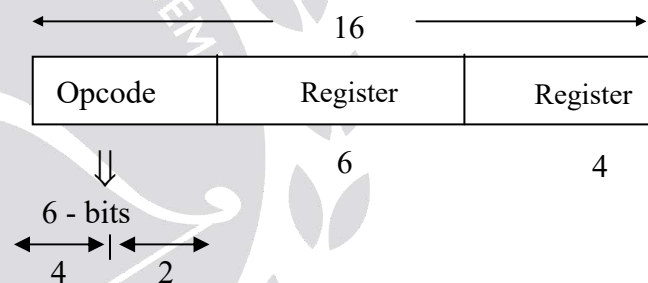


4-bits

Max R-type instructions =  $2^4$

Assume used R-type instruction = n  
unused opcode =  $2^4 - n$

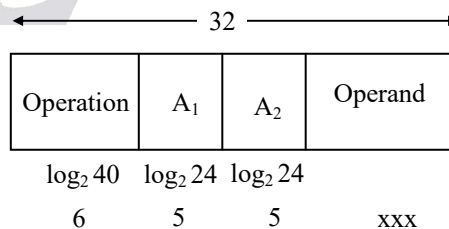
Type I- Instruction



Max I-type instructions =  $(2^4 - n) \times 2^2$   
= 8 (given 8)  
=  $2^4 - n = 2$   
n = 14

**08. Ans: 16**

**Sol:**



$\therefore 32 - 16 = 16$

**09. Ans: 16383**

**Sol:** Word size = 32 bit

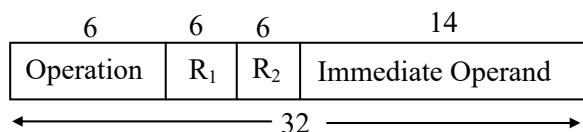
Number of CPU Registers =  $64 = 2^6$

So, for addressing a Register 6 bits are needed.

Instruction Opcode size is 32-bits.

Number of supporting Instructions = 45, so minimum 6 bits are needed.

Instruction is having with operation part, Reg1, Reg2 and Immediate operand



The Range of unsigned operand with 14-bit is 0 to  $2^{14}-1$

∴ Max unsigned integer is 16383.

**10. Ans: 500**

**Sol:** One instruction needs 34 bits,

So number of bytes needed = 5

Program size = 100

∴ Size of the memory in bytes = 500

**11. Ans: (b)**

**Sol:** Relative Addressing mode is used to relocate the program from one memory segment to other segment-without-change in code so, it is known as Position Independence Addressing mode.

**12. Ans: (c)**
**13. Ans: (b)**

**Sol:** In instruction execution cycle, to get the first operand through index addressing mode it takes one machine cycle. To get the second operand through indirect addressing mode B, it takes two more machine cycles because B is the address.

After the addition is completed the result is needed to send to the destination by using the index addressing mode, which requires one more machine cycle.

So a total of four machine cycles are required to execute the above instruction.

(Except fetch cycle)

**14. Ans: (d)**

**Sol:** Here R2 will act as base or indexed register and 20 is the displacement.

**15. Ans: -16**

**Sol:** While executing the  $i + 3$  instruction, the PC content will be the starting address of the  $i+4$ . If the target of the branch instruction is 'i' then processor takes 4 instructions addresses back (Backward jump)

Hence the displacement value is  $-4*4 = -16$ , because each instruction opcode size is 4 bytes.

**16. Ans: (b)**

**Sol:** Absolute Addressing Mode is also known as memory direct Addressing Mode.

**17. Ans: (a)**

**Sol:** After issuing an interrupt, while processing L is under execution.

Processor follows the below steps:

**Step 1:** Completion of current instruction execution

**Step 2:** Pushes the result of the current instruction status on stack

**Step 3:** Gets the new address to PC for starting ISR and executes the ISR.

**Step 4:** Pops the status from stack to continue the interrupted instruction.

**18. Ans: (d)**

**Sol:** • Stack grows upword means SP is incremented for PUSH operation and decremented for POP operation.

• One Memory location can store only one word (i.e., one byte)

• After 'CALL' execution; to store PC and PSW content; SP is incremented by '4'

$$016E_{16} + 4 = (0172)_H$$

**19. Ans: (b)**

**Sol:** The Value in register A is rotated through right 8 times. During each rotation operation, if carry flag is set the value of register B is incremented. After 8 rotations B register contains the number of 1's in register A.

**20. Ans: (a)**

**Sol:** Extending the previous question, if the contents of register A is rotated right once again, and then register A will retain its value. Therefore the instruction at X will be RRC A, #1.

**21. Ans: (d)**

**Sol:** The given program is:

Instruction	Operation	Instruction Size (No. of words)
MOV R1,(3000)	$R1 \leftarrow M[3000]$	2
<b>LOOP:</b> MOV R2,(R3)	$R2 \leftarrow M[R3]$	1
ADD R2, R1	$R2 \leftarrow R2 + R1$	1
MOV(R3), R2	$M[R3] \leftarrow R2$	1
INC R3	$R3 \leftarrow R3 + 1$	1
DEC R1	$R1 \leftarrow R1 - 1$	1
BNZ <b>LOOP</b>	Branch on not zero	2
HALT	Stop	1

Let the data at memory 3000 is 10.

The contents of R3 are 2000.

The content of memory locations from 2000 to 2010 is 100.

The number of memory references for accessing the data is

Instruction	Operation	No. of memory references
MOV R1(3000)	$R1 \leftarrow M[3000]$	1
MOV R2, (R3)	$R2 \leftarrow M[R3]$	10(loop is repeated 10 times)
MOV (R3), R2	$M[R3] \leftarrow R2$	10 (loop is repeated 10 times)

Total number of memory references are: 21

**22. Ans: (a)**

**Sol:** As the memory locations are incremented 10 times from 2000 to 2009, when the loop is terminated R3 consists of 2010, whose value will be 100(previous value) only.

**23. Ans: (c)**

**Sol:** The program is loaded from memory location 1000 onwards. The word size is 32 bits and the memory is byte addressable.

Address	Instruction		Word size
1000 to 1007	MOV R1, (3000)	$R1 \leftarrow M[3000]$	2
1008 to 1011	LOOP: MOV R2, (R3)	$R2 \leftarrow M[R3]$	1
1012 to 1015	ADD R2, R1	$R2 \leftarrow R2 + R1$	1
1016 to 1019	MOV(R3), R2	$M[R3] \leftarrow R2$	1
1020 to 1023	INC R3	$R3 \leftarrow R3 + 1$	1
1024 to 1027	DEC R1	$R1 \leftarrow R1 - 1$	1
1028 to 1035	BNZ LOOP	Branch on not zero	2
1036 to 1039	HALT	Stop	1

If the interrupt occurs at INC R3 instruction, then first the instruction is executed and the program counter consists of 1024, which is stored in stack.

**24. Ans: (a), (c) & (d)**

**Sol:** Option (b) is false because in PC relative Addressing Mode; the Effective address is placed in PC after computation.

**25. Ans: (a)**

**Sol:** (3)  $PC_r$ ,  $MAR_w$ ,  $MEM_r$ : Read PC value and write that address into MAR, to read memory (next instruction)  
 (5)  $MDR_r$ ,  $IR_w$ : Next instruction is copied to IR from MDR (Memory read copies contents in MDR first)  
 (2)  $R1_r$ ,  $TEMP1_w$ : Read value of R1 in TEMP1  
 (1)  $R2_r$ ,  $TEMP1_r$ ,  $ALU_{add}$ ,  $TEMP2_w$ : This operation says  $TEMP2 \leftarrow R2 + TEMP1$   
 (4)  $TEMP2_r$ ,  $R0_w$ : This operation copies final result of addition from TEMP2 to R0.

**26. Ans: (d)**

**Sol:** To execute interrupt cycle, the present content of PC will be pushed to stack with the help of MBR and MAR before placing ISR address in PC. (Always only MAR and MBR are used to address Memory in basic computer).

## 5. Pipeline Organization

**01. Ans: (c)**

**Sol:** Max. stage delay = 160 ns

Buffer delay = 5 ns

Pipeline clock = 165 ns

$T_{1000} = (K + n - 1) T_p \text{ clock}$

$$= (4 + 999) * 165 = \left( \frac{165495}{1000} \right) \mu s$$

$$= 165.5 \mu s$$

**02. Ans: (b)**

**Sol:** For  $D_1$  processor, maximum  $T_{\text{seg}} = 4 \text{ ns}$ ,

$$n = 100, k = 5$$

$$\text{Time} = 104 \times 4 \text{ ns} = 416 \text{ ns}$$

For  $D_2$  processor,

$$n = 100, k = 8, T_{\text{seg}} = 2 \text{ ns}$$

$$\text{Time} = 107 \times 2 \text{ ns} = 214 \text{ ns}$$

Hence, 202 ns time will be saved

**03. Ans: (b)**

**Sol:**  $t_n = 12 \text{ ns}$ , maximum  $T_{\text{seg}} = 6 \text{ ns}$

$$\therefore S = t_n / t_p = 2$$

**04. Ans: 1.51**

**Sol:** For Naive pipelined CPU

$$K = 5, T_{\text{seg}} = 20 + 2 = 22 \text{ ns}, n = 20.$$

Total time needed for 20 instructions

$$= (5 + 20 - 1) \times 22 \text{ ns} = 24 \times 22 \text{ ns} \\ = 528 \text{ ns}$$

**For Efficient pipelined processor**

$$T_{\text{seg}} = 12 + 2 = 14 \text{ ns}; k = 6, n = 20$$

Total time for 20 instructions

$$(6 + 20 - 1) \times 14 \text{ ns} = 350 \text{ ns}.$$

$$\text{Speed up} = \frac{t_n}{t_e} = \frac{528}{350}$$

$$= 1.50857$$

$$\cong 1.51$$

**05. Ans: (c)**

**Sol:**  $f \propto 1/T$

Minimum clock time gives highest clock frequency for the given pipelined processor.

For P1 Largest clock time is 2 ns.

For P2 Largest clock time is 1.5 ns.

For P3 Largest clock time is 1 ns.

For P4 Largest clock time is 1.1 ns.

So, P3 gives highest peak clock frequency.

**06. Ans: (c)**

**Sol:** Efficiency =  $\frac{S}{K}$

Where S = speed up, K = number of stages

$$K = \frac{S}{\text{efficiency}}$$

$$= \frac{6.6}{.88} = 7.5$$

So, minimum 8 stages are needed

**07. Ans: 3.2**

**Sol:** Non-pipeline CPU frequency = 2.5 GHz

$$T = 0.4 \text{ ns}$$

$\therefore$  One instruction time =  $4 \times 0.4 \text{ ns}$

$$t_n = 1.6 \text{ ns}$$

Pipeline CPU frequency = 2 GHz

$$\therefore T = 0.5 \text{ ns} = t_p$$

Only one clock cycle time is sufficient to execute one instruction.

$$S = \frac{t_n}{t_p}$$

$$= \frac{1.6 \text{ ns}}{0.5 \text{ ns}} = 3.2$$

**08. Ans: 4**

**Sol:** For 'n' number of instructions

$$t_n = 6 \times n \text{ clks (k = 6).}$$

Highest speed up  $k = 6$  if there is no stall cycle and all stage delays are equal but 25% of instructions need 2 stalls.

$$t_p = (0.75 n \times 1) + (0.25 n \times (2 + 1)) = 1.5 n \text{ clks.}$$

$$\therefore S = \frac{t_n}{t_p} = \frac{6 n \text{ clks}}{1.5 n \text{ clks}} = 4$$

**09. Ans: 219**

**Sol:** Number of stages = 5(IF, ID, OF, PO, WR),

$k = 5$

$n = 100$ , except PO, all stages take one clock

In P.O stage

40 instructions take 3 clocks

35 instructions take 2 clocks

and remaining 25 instructions take 1 clk

If all instructions requires one clock in all stages, total clocks required =  $(k + n - 1)$   
 $= 5 + 100 - 1 = 104$

But, 40 instructions requires 3 clocks each i.e. 40 instructions execution requires '80' more clocks and 35 instructions requires 2 clocks i.e. 35 instructions require 35 more clocks.

So, total number of clocks required  
 $= 104 + 80 + 35 = 219$

**10. Ans: (b)**

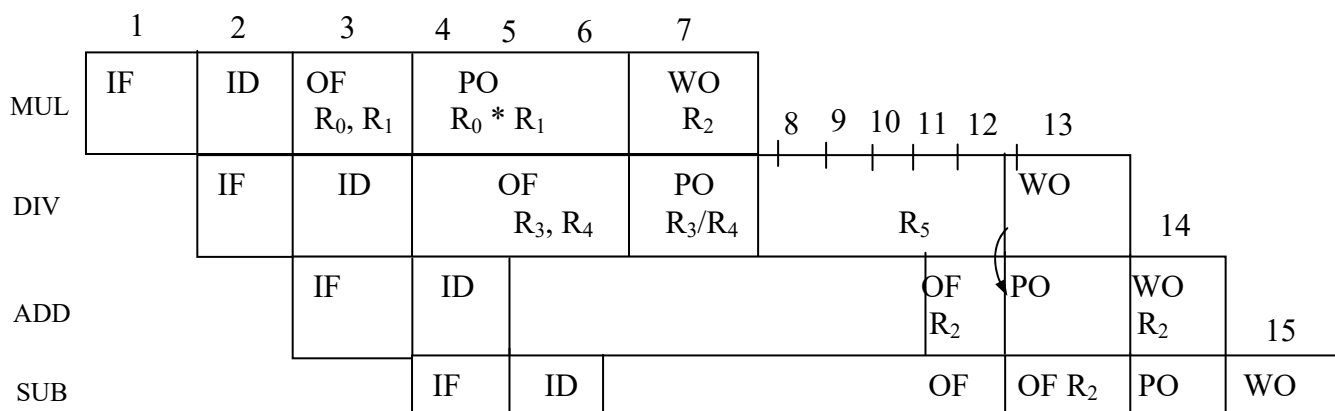
**Sol:** Non-pipelined system delay = 30 ns

Max. Pipeline delay = 12 ns

$$S = 30 \text{ ns} / 12 \text{ ns} = 2.5$$

**11. Ans: (b)**

**Sol:**



**12. Ans: 13**

**Sol:**

	IF	OF	PO	WB
MUL	1	2	5	6
DIV	2	3	10	11
ADD	3	4	11	12
SUB	4	5	12	13

**13. Ans: (c)**

**14. Ans: (d)**

**Sol:** Number of stages = 5

One stage delay = 2 ns

- While executing more number of instructions only one stage delay is sufficient for executing one instruction when there is no Hazard.
- Number of Non Hazard instructions = 80%

$\therefore$  it's Average time =  $0.8 \times 2 \text{ ns} = 1.6 \text{ ns}$

For executing one Hazard instruction it takes all stage delays i.e., 10 ns.

$\therefore$  It's average time is  $0.2 \times 10 \text{ ns} = 2 \text{ ns}$

Average instruction time

$$= 1.6 \text{ ns} + 2 \text{ ns} = 3.6 \text{ ns}$$

**15. Ans: (c)**

**Sol:** 2 Stall cycles

CPU clock frequency = 1GHz

Out of  $10^9$  instructions 20% of instructions are branch instructions, which requires 3 clock cycles. The remaining 80%

instructions require only one clock pulse for their completion.

Total execution time

$$= 10^9 \times (80/100) \times 10^{-9} + 10^9 \times (20/100) \times 3 \times 10^{-9} \\ = 0.8 + 0.6 = 1.4 \text{ sec}$$

**16. Ans: (b)**

**Sol:** Pipeline clock = Max (stage delay + Overhead)

$$= \text{Max} (5, 7, 10, 8, 6) + 1 = 11 \text{ ns}$$

CPU gets target address after completion of branch instruction in EX stage only.

$(n+k-1) \times 11 \text{ ns} + \text{stall delay} (3)$

$$= ((8 + 5 - 1) \times 11 \text{ ns}) + (3 \times 11) \text{ ns} \\ = 165 \text{ ns}$$

**17. Ans: 1.54**

**Sol:** 'P' has 5 stages

IF	ID/RF	EX	MEM	WB
1	2.2	2	1	0.75 ns

'Q' has 8 stages

IF	ID	RF1	RF2	EX1	EX2	MEM	WB
1	2.2/3	2.2/3	2.2/3	1	1	1	0.75

P - highest clock cycle time = 2.2 ns.

Q - highest clock cycle time = 1 ns.

In 'P' pipeline new instruction fetching is stopped for 2 stage delays

Where in 'Q' pipeline new instruction fetching is stopped for 5 stage delays

Number of branch instructions = 20%.

∴ 'P' total time is

$$(0.8 \times 2.2 \text{ ns}) + (0.2 \times (2 + 1))2.2 \text{ ns} = 3.08 \text{ ns}$$

'Q' total time is

$$0.8 \times 1 \text{ ns} + (0.2 \times (5 + 1)) \times 1 \text{ ns} = 2 \text{ ns}$$

$$\begin{aligned} \therefore \frac{P}{Q} &= \frac{3.08 \text{ ns}}{2 \text{ ns}} \\ &= 1.54. \end{aligned}$$

**18. Ans: 2.16**

**Sol:** Cycle time in non-pipeline processor

$$= \frac{1}{2.5 \text{ GHz}} = 0.4 \text{ n sec}$$

Time required to execute an instruction

$$= 5 * 0.4 = 2 \text{ nsec}$$

Cycle time in pipeline processor

$$= \frac{1}{2 \text{ GHz}} = 0.5 \text{ n sec}$$

Average cycles per instruction

$$\begin{aligned} &= 0.3 [0.05 * (50 + 1) + 0.95 * 1] + \\ &\quad 0.6 [1] + 0.1 [0.5 * (2 + 1) + 0.5 * 1] \\ &= 1.05 + 0.6 + 0.2 \\ &= 1.85 \end{aligned}$$

Time required to execute an instruction in pipeline processor =  $1.85 * 0.5$

$$= 0.925 \text{ nsec}$$

Speedup =

$$\begin{aligned} &\frac{\text{avg.instruction execution time in non - pipeline}}{\text{avg.instruction execution time in pipeline}} \\ &= \frac{2}{0.925} \\ &= 2.162 \\ &\cong 2.16 \end{aligned}$$

**19. Ans: (d)**

- Sol:** 1. The  $(j+1)^{\text{th}}$  instruction uses the result of the  $j^{\text{th}}$  instruction as an operand, comes under data dependency and it causes data hazard(RAW).
2. The execution of a conditional jump instruction comes under conditional dependency and it causes control hazard.
3. The  $j^{\text{th}}$  and  $j + 1$  instruction require the ALU at the same time comes under structural hazard(WAR).

**20. Ans: (c)**

**21. Ans: (b)**

**Sol:** It is also known as W.A.R Hazard.

Anti-dependence Hazard creates Hazard (i.e. needs stall) when a low latency instruction is completed before a longer latency instruction that appears earlier in the program only BUT NOT ALWAYS.

**22. Ans: (b) & (c)**

**Sol:** Option (a) is false because RAW is known as True dependency. Option (d) is false because WAR can be minimized by using Three Address instructions but not two Address instructions.

## 6. Control Unit Design

**01. Ans: (d)**

$$\text{Sol: } S_8 = I_1T_4 + I_2T_4 + I_3T_4 + I_4T_4 \\ = I \times T_4 = T_4$$

$$\therefore I = (I_1 + I_2 + I_3 + I_4)$$

$$S_7 = (I_1T_3 + I_2T_3 + I_3T_3 + I_4T_3) \\ + (I_3T_1 + I_3T_2 + I_3T_3 + I_3T_4) + (T_4I_3 + T_4I_4)$$

$$\therefore S_7 = T_3 + I_3 + T_4 \times I_4$$

**02. Ans: (b)**

**Sol:** Fastest Control unit is hard-wired control unit and vertical micro-programming control unit is slowest.

**03. Ans: (d)**

**04. Ans: (a)**

**Sol:** Total Size of micro-instructions = 26 bits  
 Size of micro-operation = 13 bits  
 Total inputs for the multiplexer (Status bits) inputs = 8

So the multiplexer selection lines field(Y)  
 $= 3 \text{ bits } (2^3 = 8)$

The number of bits in the next address field size(X) =  $13 - 3 = 10 \text{ bits}$

Size of control memory =  $2^{10} = 1024$

**05. Ans: (a), (b), (c) & (d)**

**06. Ans: (a), (b) & (c)**

**07. Ans: (d)**

**Sol:** All the given characteristics are belonging to RISC processor.

**08. Ans: (a), (b), (c) & (d)**

**Sol:** All statements are true.

## 7. I/O Organization

**01. Ans: (b)**

**02. Ans: (b)**

**Sol:** 10 KBPS = 10 KB is transferred in 1 sec =  $10^4 \text{ B}$

1 byte takes =  $0.1 \text{ ms} = 100 \mu\text{s}$

$\therefore$  Minimum waiting time needed is  $100 \mu\text{s}$  for system

Programmed I/O takes  $100 \mu\text{s}$

Interrupt driven takes  $4 \mu\text{s}$

$$\therefore \text{Gain} = \frac{100 \mu\text{s}}{4 \mu\text{s}} = 25$$

**03. Ans: (a)**

**Sol:** CPU gives highest priority for high speed devices and least priority for low speed devices. Hard disk has higher priority than others because it is fastest secondary memory.

**04. Ans: (d)**

**Sol:** CPU first takes care about its temperature.

**05. Ans: 456**

**Sol:** Terminal Count Register size = 16 bit.

So, for one transfer operation of 64 KB, the register content will become zero, so, number of times the content of the register to be filled is

$$\frac{29154 \text{ KB}}{64 \text{ KB}} \cong 456$$

**06. Ans: 80000**

**Sol:** Type of Data transfer is cycle steal i.e. one 8 bit characters / one request

$$f = 2\text{MHz}; T = 0.5\mu\text{sec}$$

$$\therefore \text{Clock cycle time} = 0.5 \mu\text{sec}$$

Hence DMA transfers 8 bits in  $0.5 \mu\text{sec}$

i.e 16 bits in  $1 \mu\text{sec}$

i.e. in one second, it can transfer 160.00000 bits, but only 0.5% processor clocks are used for DMA transfer, Hence the Data transfer rate

$$= 16000000 \times \frac{0.5}{100}$$

$$= 80000 \text{ bits/sec}$$

**07. Ans: (b)**

**Sol:** For vectored hardware interrupt, the interrupting device supplies the respective address with additional hardware.

**08. Ans: (b)**

**Sol:** In single line interrupt system contains a single interrupt request line and an interrupt grant line. In this system it may be possible for

more than one I/O device request interrupt at the same time. By using 8259 IC it is possible to connect more number of IO devices. So in single interrupt system vectored interrupts are not possible but multiple interrupting devices are possible

**09. Ans: (a)**

**Sol:** Non-pipelined system requires

$$(2+2+1+1+1) \times 500 + 2 = 3502 \text{ cycles}$$

$$\text{DMA clock need} = 20 + (2 \times 500)$$

$$= 1020 \text{ cycles}$$

$$\text{Speed up} = 3502/1020 = 3.44$$

**10. Ans: (a) & (b)**

**Sol:** Option 'c' is false because Memory Mapped IO technique is used to connect more number of IO devices. Option (d) is false because Memory Mapped IO technique uses Memory Read and Memory Write control signals.