

GATE PSUs

State Engg. Exams

MADE EASY
WORKBOOK 2024



**Detailed Explanations of
Try Yourself *Questions***

Computer Science & IT
Computer Organization and
Architecture



1

Data Representation



Detailed Explanation of Try Yourself Questions

T1 : Solution

(a)

$$14.25 = 1110.010 = 1.110010 \times 2^3$$
$$M = 110010$$
$$E = 127 + 3 = 130 = 10000010$$

1	8	23	
S	E	M	= C1640000H
1	10000010	11001000...	

T2 : Solution

(a)

$$3.248 \times 10^4 = (32480)_{10} = (111111011100000)_2$$
$$= 1.11111011100000 \times 2^{14}$$
$$= 1.M \times 2^E$$
$$M = 111110111$$
$$E = 14 + 127 = (141)_{10}$$

	E	M	
0	10001101	111110111100000000000000	
1 bit	8 bits	23 bits	

T3 : Solution

(b)

$b = 28E + 11$ (maximum decimal value represented in IEEE 754 format with 32 bit representation)

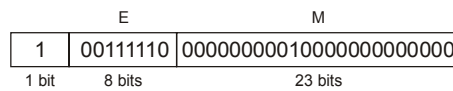
$a = 0.052$

$b + a = (28E + 11) + (0.052) = 28E + 11$ (Overflow)

$b + a - b = b - b = 0.$

T4 : Solution

(c)



$$(-1) \times 1.M \times 2^E$$

$$(-1) \times 1.0000000001 \times 2^{62-127}$$

$$\text{Exponent (E)} = 62 - 127 = -65$$

$$(-1) \times (1 + 0.000000001) \times 2^{-65}$$

$$(-1) \times (1 + 2^{-10}) \times 2^{-65}$$

T5 : Solution

(d)

Sign extension is the operation, in computer arithmetic, of increasing the number of bits of a binary number while preserving the number's sign (positive/negative) and value.

Therefore, it is converting a signed integer from one size to another.



2

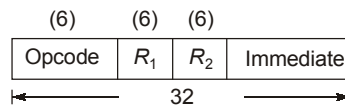
Machine Instructions and Addressing Modes



Detailed Explanation of Try Yourself Questions

T1 : Solution

(16383)



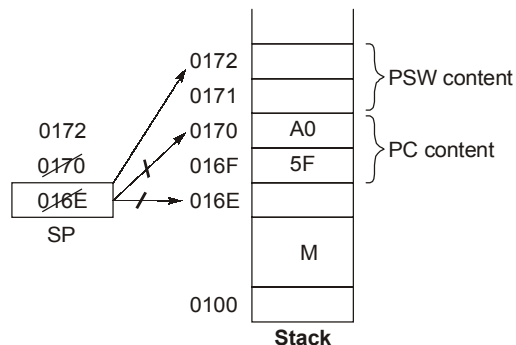
$$32 - (6 + 6 + 6) = 14 \text{ bits for immediate field}$$

⇒

$$2^{14} - 1 = 16383 \text{ maximum possible value of immediate operand.}$$

T2 : Solution

(d)



Just before CALL instruction execution, SP contains 016E

While CALL execution:

- (i) PC contents are pushed i.e., SP incremented by 2 ⇒ SP = 0170
 - (ii) PSW contents are pushed i.e., SP incremented by 2 ⇒ SP = 0172
- ∴ The value of stack pointer is $(0172)_{16}$.

T3 : Solution

(b)



Number of register are $2^x \Rightarrow x$ bits for register in opcode

Size of memory cell is $2^y \Rightarrow y$ bits for word offset

Hence number of bits for opcode = $32 - (x + y)$

Therefore number of opcodes = $2^{32 - (x + y)}$

There are only 'z' two address instructions and hence remaining opcodes = $2^{32 - (x + y)} - z$

These remaining are used for one address register reference instructions.

Number of one address instructions = $(2^{32 - (x + y)} - z)2^y$

T4 : Solution

(c)

$$EA = PC + \text{Address field value}$$

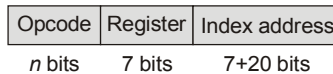
$$EA = 38248 + (-12) = 38236$$

T5 : Solution

(b)

Number of registers = 128

Number of bits = $\log 128 = 7$



In indexing addressing mode effective memory location is stored in register.

So Index Address bit = Register bit + Address bit

$$= 7 + 20 = 27$$

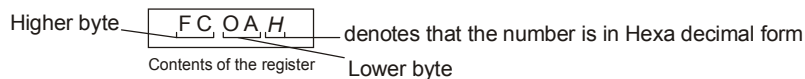
$\therefore n + 7 + 27 = n + 34$ is length of the instruction.

T6 : Solution

(c)

The given instruction is stored in 16 bits register. The first byte (lower byte) of the instruction store at the memory location 4002 and second byte (higher byte) stored at 4003.

When we use little-endian mechanism the lower byte of the instruction is copied into lower byte of the register and higher byte of the instruction is copied into higher byte of the register.



When we use big endian mechanism the lower byte of the instruction is copied into higher byte of the register and higher byte of the instruction is copied into lower byte of register.



T7 : Solution

(a)

	I_1	6 cycles	
	I_2	6 cycles	
Label	I_3	6 cycles	} 5 times
	I_4	4 cycles	
	I_5	4 cycles	
	I_6	2 cycles	

$$\text{Time} = I_1 + I_2 + 5(I_3 + I_4 + I_5 + I_6)$$

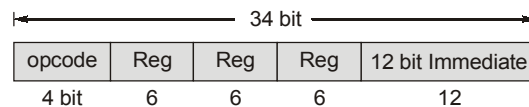
$$\text{Time} = 12 \text{ cycles} + 80 \text{ cycles} = 92 \text{ cycles}$$

$$\text{Cycle time} = \frac{1}{1 \text{ KHz}} \text{ sec} = 1 \text{ msec}$$

$$\text{Program execution time} = 92 \text{ cycles} \times 1 \text{ msec} = 92 \text{ msec.}$$

T8 : Solution

(500)



One instruction size = 34 bit = 5 bytes

100 instruction occupies 500 bytes.

T9 : Solution

(b)

$$\text{Average of time} = \{(0.2 \times 0) + (0.2 \times 0) + (0.2 \times 4) + (0.1 \times 8) + (0.17 \times 6) + (0.13 \times 6)\} = 3.4 \text{ cycles}$$

$$\text{Clock cycle time} = \frac{1}{1} \text{ GHz} = 1 \text{ ns}$$

$$\text{So, average of time} = 3.4 \text{ cycles} \times 1 \text{ ns} = 3.4 \text{ ns}$$

1 operand 3.4 ns

Number of operands in 1 sec

$$\text{Number of operands} = \frac{1 \text{ operand}}{3.4 \text{ ns}} = 0.29411 \times 10^9 \text{ operand/sec.}$$

$$\therefore \text{Operand fetch rate} = 294.11 \text{ million words/sec}$$

T10 : Solution**(b)**2000-2007 : I_1 2008-2011 : I_2 2012-2015 : I_3 2016-2019 : I_4 2020-2027 : I_5 2028-2031 : I_6 2032-2033 : I_7

Return address pushed on to the stack is 2032 because it is a HALT instruction and its only return address will be its starting address.

T11 : Solution**(a)**

- Indirect addressing → Passing array as parameter.
- Indexed addressing → Array implementation.
- Base register addressing → Writing relocatable code.



3

CPU Design



Detailed Explanation of Try Yourself Questions

T1 : Solution

(a)

Average CPI = $\Sigma(\text{IC} \times \text{CPI})$ cycle time

$$\text{Total IC} = \frac{(45000 \times 1) + (32000 \times 2) + (15000 \times 2) + (8000 \times 2)}{45000 + 32000 + 15000 + 8000} = 1.55 \text{ cycles}$$

$$\text{Cycle time} = \frac{1}{80 \text{ MHz}} \text{ sec} = 0.0125 \mu\text{sec}$$

Average instruction ET = $(1.55 \times 0.0125) = 0.019375 \mu\text{sec}$

1 instruction $\Rightarrow 0.019375 \mu\text{sec}$

Number of instructions in 1 sec

$$\text{Number of instructions} = \frac{1}{0.019375} \times 10^6 \text{ inst./sec} = 51.61 \text{ MIPS}$$

T2 : Solution

(c)

In the given μ -program we have used the register MAR, MBR and Instruction Register (IR).

IR is used only during fetching of the instruction.

Hence operation is instruction fetching.

T3 : Solution

(b)

Configurations for CPU in decreasing order of operating speeds: Hardwired control > Horizontal micro-programming > Vertical micro-programming (slowest because it involves decoding).

T5 : Solution**(10)**

Atmost 2 control signals are active, that means for one signals minimum $\log_2(25)$ bits required i.e. 5.
For 2 control signals = $5 \times 2 = 10$ bits required.



4

Instruction Pipelining



Detailed Explanation of Try Yourself Questions

T1 : Solution

(4)

$$\text{Speed up}(S) = \frac{\text{Pipe depth}}{(1 + \# \text{ stalls/instruction})}$$

$$\text{Number of stalls/instruction} = 0.75 \times 0 + 0.25 \times 2 = 0.5$$

$$\therefore S = \frac{6}{1+0.5} = 4$$

T2 : Solution

(c)

Clock cycle time = Maximum of stage delays.

For P1: clock cycle time = Maximum is 2

For P2: clock cycle time = Maximum is 1.5

For P3: clock cycle time = Maximum is 1

For P4: clock cycle time = Maximum is 1.1

Minimum clock cycle time gives the high clock rate.

Minimum of (2, 1.5, 1, 1.1) = 1

\therefore P3 has peak clock cycle rate.

T3 : Solution

(1.54)

$$T_{avg} = (1 + \text{stall freq.} \times \text{stall cycle}) \times T_{clock}$$

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

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

$$\frac{P}{Q} = \frac{TP_{avg}}{TQ_{avg}} = \frac{3.08}{2} = 1.54$$

T4 : Solution

(3.2)

Non-pipelined processor: For n instructions execution time = $(n \times 4) / 2.5 = 1.6 n$ nanoseconds.

Pipelined processor: For n instructions execution time = $n / 2 = 0.5 n$ nanoseconds.

$$\text{Speedup} = 1.6 n / 0.5 n = 3.2$$

T5 : Solution

(13)

↑	SUB				IF	-	-	-	-	-	-	OF	PO	WB
	ADD			IF	-	-	OF	-	-	-	-	PO	WB	
	DIV		IF	OF	-	-	PO	PO	PO	PO	PO	WB		
	MUL	IF	OF	PO	PO	PO	WB							
		1	2	3	4	5	6	7	8	9	10	11	12	13

∴ It takes 13 clock cycles.

T6 : Solution

(c)

S ₄				I ₁	I ₂	I ₃	I ₄	I ₅	I ₆					
S ₃			I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇					
S ₂		I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	I ₈					
S ₁	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	I ₈	I ₁₂	I ₁₃	I ₁₄	I ₁₅	I ₁₆	

So total 13 instruction are executed.

$$\therefore K = 4; n = 13, t_p = 4 \text{ ns}$$

$$ET = (K + n - 1) \times t_p$$

$$= (4 + 13 - 1) \times 4 \text{ ns} = 64 \text{ ns}$$

T7 : Solution

(c)

$$S_x = \frac{\text{Pipeline depth}}{(1 + \text{Frequency} \times \text{Number of stalls per instruction})}$$

$$= \frac{5}{1 + (0.3 \times 4)} = 2.27$$

$$S_y = \frac{9}{1 + (0.3 \times 8)} = 2.64$$

$$\frac{S_x}{S_y} = 0.859$$

T8 : Solution

(a)

A :

$$K = 6$$

$$n = 16$$

$$t_p = 8 \text{ ns}$$

$$x = (K + n - 1) t_p = 168 \text{ ns}$$

$$x / y = 1.16$$

B : $K = 9$

$$n = 16$$

$$t_p = 6 \text{ ns}$$

$$y = (9 + 16 - 1) 6 = 144 \text{ ns}$$

T9 : Solution

(a)

	CC ₁	CC ₂	CC ₃	CC ₄	CC ₅	CC ₆	CC ₇	CC ₈	CC ₉	CC ₁₀
I ₁	IF	ID	OF	EX	MA	WB				
I ₂		IF	ID	OF	///	EX	MA	WB		
I ₃			IF	ID	///	OF	EX	MA	WB	
I ₄				IF	///	ID	OF	EX	MA	WB

T10 : Solution(b) S_1 is true

S_2 is true because there is an anti-dependence between instructions I_2 and I_4 .

S_3 is false because anti-dependence stalls may be avoided when register renaming is used.

T11 : Solution

(33.28)

P_1 : 4-stage

$$t_p = \text{Max (stage delay)} = 800 \text{ ps}$$

1 instruction — 800 ps

? number of instruction — 1 sec

$$TP_{P_1}(\text{throughput}) = 1250 \text{ instruction/sec}$$

P_2 : 5-stage

$$t_p = 600 \text{ ps}$$

1 instruction — 600 ps

? number of instruction — 1 sec

$$TP_{P_2}(\text{throughput}) = 1666 \text{ instruction/sec}$$

1250 — 100% (old)

(1666 – 1250) — ? (New)

$$\Rightarrow \frac{416}{1250} = 0.3328$$

i.e., 33.28%

T12 : Solution

(4)

Given that, 3 stage pipeline with stage latencies

$$\tau_1 = \frac{3\tau_2}{4} = 2\tau_3 = \frac{3\tau_2}{4}, \frac{3\tau_2}{4} = 2\tau_3$$

$$\tau_1 : \tau_2 = 3 : 4, \tau_2 : \tau_3 = 8 : 3$$

Ratio of $\tau_1 : \tau_2 : \tau_3$ is $6x, 8x, 3x$ respectively. So, largest stage time $8x$.

So, calculate frequency = $\frac{1}{8x}$

$$\Rightarrow \frac{1}{8x} = 3 \text{ GHz}$$

$$\Rightarrow \frac{1}{x} = 24 \text{ GHz} \quad \dots(1)$$

Now, largest stage latency divide into 2 half parts i.e., $8x$ divide into $4x$ and $4x$.

So, 4 pipeline with stage latencies are $6x, 4x, 4x, 3x$.

Now, largest stage time = $6x$

So, calculate new frequency = $\frac{1}{6x}$

$$\frac{1}{6x} = \frac{24}{6} = 4 \text{ GHz}$$

[$\because \frac{1}{x} = 24 \text{ GHz}$ from eq. (1)]



5

Memory Organization



Detailed Explanation of Try Yourself Questions

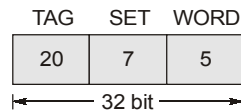
T1 : Solution

(a)

Miss ratio = n/N , if the access sequence is passed through a cache of associativity $A \geq k$ exercising least-recently used replacement policy.

T2 : Solution

(20)



$$\text{WORD offset} = 5 \text{ bits} \quad [\because \text{word length} = 32 \text{ bits}]$$

$$\text{SET offset} = 7 \text{ bits}$$

$$[\because \text{Number of blocks} = \frac{16\text{kB}}{8\text{Words}} = 512 \text{ blocks}]$$

$$\text{Number of sets} = \frac{512}{4} = 128$$

$$\therefore \text{Number of TAG bits} = 32 - (7 + 5) = 20 \text{ bits.}$$

T3 : Solution

(d)

A smaller cache block implies a larger cache tag and hence higher cache hit time.

So it incurs lower cache miss penalty.

T4 : Solution**(d)**

By doubling the associativity of the cache, width of the processor to main memory databus is guaranteed to be not affected.

T5 : Solution**(10000)**

To send one word data to the bus takes 100 ns.

So, one words store \rightarrow 100 ns

? Number of words store \rightarrow 1 msec

$$\therefore \text{Number of words store} = \frac{1 \text{ msec}}{100 \text{ ns}} = 10000$$

T6 : Solution**(1.68)**

$$\begin{aligned} \text{Average read time} &= 0.9 \times 1 \text{ ns} + 0.1 \times 5 \text{ ns} \\ &= 0.9 + 0.5 = 1.4 \text{ ns} \end{aligned}$$

In the execution sequence number of read operations = 160

So total time required for read operation

$$= 160 \times 1.4 \text{ ns} = 224 \text{ ns}$$

$$\text{Average write time} = 0.9 \times 2 \text{ ns} + 0.1 \times 10 \text{ ns}$$

$$= 1.8 + 1 = 2.8 \text{ ns}$$

In the execution sequence number write operation = 40.

So, the total time required for write operation = $40 \times 2.8 \text{ ns} = 112 \text{ ns}$

Total time for instruction execution time for both read and write = $224 + 112 = 336 \text{ ns}$.

200 times, access takes _____ 336 ns.

1 time, access takes _____ ?

$$\text{Average memory access time} = \frac{336}{200} = 1.68 \text{ ns}$$

T7 : Solution**(14)**

$$T_{\text{readmiss}} = 50 \text{ nsec}$$

$$T_{\text{readhit}} = 5 \text{ nsec}$$

$$h = 0.8$$

$$T_{\text{avgread}} = h \times T_{\text{readhit}} + (1 - h) \times T_{\text{readmiss}}$$

$$= 0.8 \times 5 + 0.2 \times 50$$

$$= 4 + 10 = 14 \text{ nsec}$$

T8 : Solution

(a)

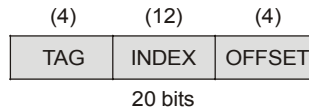
$$\text{Block size} = 16 \text{ bytes} = 2^4 \text{ bytes}$$

$$\text{Offset} = 4 \text{ bits}$$

$$\text{Index} = 12 \text{ bits } [\because \text{number of lines} = 2^{12}]$$

$$\text{Main memory size} = 2^{20} \text{ bytes}$$

$$\Rightarrow \text{Physical address} = 20 \text{ bits}$$



$$\begin{array}{c} \text{Index} \\ \hline E \quad \boxed{2 \quad 0 \quad 1} \quad F \quad (\text{Hexa}) \\ \hline 1110 \quad 0010 \quad 0000 \quad 0001 \quad 1111 \quad (\text{Binary}) \end{array}$$

$\therefore E$ is tag and 201 is line address (index).

T9 : Solution

(b)

Range of blocks is: 0 to $2^S - 1$

$2^S - 1$ goes to cache line $m - 1$

$2^S - 2$ goes to cache line $m - 2$

⋮

$2^S - m$ goes to cache line 0.

T10 : Solution

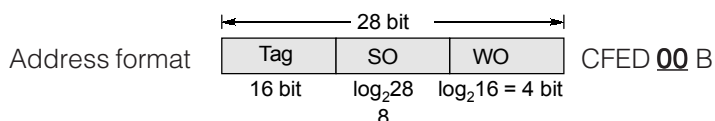
(c)

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

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

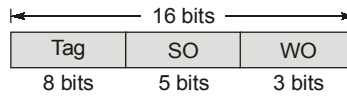
Physical address size = 28 bits

2 way set associative cache.



T11 : Solution

(b)



Therefore the block offset bits for the given addresses are:

10011 – 19

00100 – 4

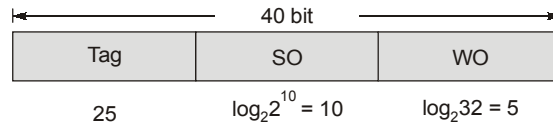
01101 – 13

T12 : Solution

(b)

$$\text{Number of lines} = \frac{2^{18}}{2^5} \Rightarrow 2^{13}$$

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



$$\begin{aligned} \text{Tag memory size} &= S \times P \times \text{Tag bits} \\ &= 2^{10} \times 8 \times (25 + 4 + 1 + 2) \text{ bits} = 256 \text{ K bits} \end{aligned}$$

T13 : Solution

(c)

$$H_r = 0.85$$

$$f_r = 0.6$$

$$f_w = 0.4$$

$$H_w = 1 \text{ (Simultaneous memory organization)}$$

$$T_m = 72 \text{ ns/word} = 288 \text{ ns/block}$$

$$S = \frac{T_m}{T_c}$$

$$T_c = \frac{T_m}{S} \Rightarrow \frac{288}{5} = 57.6 \text{ ns}$$

$$T_w = 72 \text{ ns}$$

$$T_{\text{avgr}} = H_r T_c + (1 - H_r) T_m = 92.16 \text{ ns}$$

$$T_{\text{avgw}} = H_w T_w = 72 \text{ ns}$$

$$\begin{aligned} T_{\text{avgwt}} &= (f_r \times T_{\text{avgr}}) + (f_w \times T_{\text{avgw}}) \\ &= 84.09 \text{ ns} \end{aligned}$$

T14 : Solution

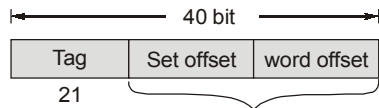
(30)

1. Miss rate = 0.8, 10 MB
 $T_c = \text{ms}$
 $T_d = 10 \text{ ms}$
 $T_{\text{avg}} = HT_c + (1 - H)(T_d + T_c) = 9 \text{ ns}$
2. Miss rate = 0.6, 20 MB
 $T_c = 7 \text{ ms}$
3. Miss rate = 0.4, 30 MB
 $T_{\text{avg}} = 5 \text{ ms}$
4. Miss rate = 0.35, 40 MB
 $T_{\text{avg}} = 4.5 \text{ ms}$
5. Miss rate = 0.3, 50 MB
 $T_{\text{avg}} = 4 \text{ ms}$
6. Miss rate = 0.25, 60 MB
 $T_{\text{avg}} = 3.5 \text{ ms}$
7. Miss rate = 0.2, 70 MB
 $T_{\text{avg}} = 3 \text{ ms}$
8. Miss rate = 0.15, 80 MB
 $T_{\text{avg}} = 2.5 \text{ ms}$

So, 30 MB will be the answer.

T15 : Solution

(24)



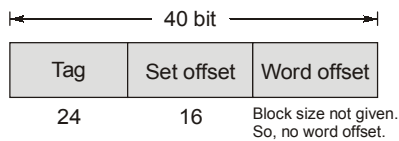
Cache index ($\log_2 512 \text{ K} = 19 \text{ bits}$)
 Block size not given so, cache index is considered

Or

Hypothetically, line size considered as cell size.

\therefore Number of lines = 512 K

So, Number of sets = $\frac{512\text{K}}{8} \Rightarrow \frac{2^{19}}{2^3} = 2^{16}$



So, TAG size is 24 bits.



6

Input Output Interface



Detailed Explanation of Try Yourself Questions

T1 : Solution

(c)

In the programmed input output, CPU checks periodically for input output request.

In the interrupt driven input output, CPU need not wait until input output completes.

T2 : Solution

(a)

The daisy-chaining method of establishing priority consists of a serial connection of all devices that request an interrupt. The device with the highest priority is placed in the first position, followed by lower-priority devices up to the device with the lowest priority, which is placed last in the chain. The farther the device is from the first position, the lower is its priority. Therefore daisy-chain gives non-uniform priority to various devices.

T3 : Solution

(c)

I/O redirection implies connection two programs through a pipe.

T4 : Solution

(b)

Program counter can be changed in user mode, but remaining all execute only in system mode.

T5 : Solution

(a)

In cycle stealing mode, the DMA gets control over address bus and control bus from CPU.

T6 : Solution

(d)

Cycle stealing mode:

Tranfering control to I/O = 300 ns

Tranfering control I/O to CPU = 300 ns

Time to transfer one byte = 300 + 700 + 300

= 1300 nsec = 1.3 μ secTotal time to transfer 100 bytes = 100 \times 1.3 μ sec = 130 μ sec**T7 : Solution**

(b)

It is transferring 16000 bits in 1 second

1 character = 8 bits

$$\text{Number of characters} = \frac{16000}{8} = 2000$$

2000 characters = 1 second

1 character takes 500 micro seconds.

Processor accesses main memory in every μ s

$$\therefore \frac{1}{500} \times 100\% = \frac{1}{5} = 0.25\%$$

T8 : Solution

(c)

I/O transfers at 8 KB/sec

 \Rightarrow 1 byte it takes 125 μ secEach interrupt process takes 75 μ sec

$$\Rightarrow \frac{75}{125} \times 100 = 60\% \text{ of CPU time consumed.}$$

T9 : Solution

(b)

Burst Mode:Beginning and end of the transfer bus control takes \Rightarrow 300 + 300 = 600 ns time

Time to transfer 100 bytes from I/O:

$$= \frac{100 \text{ bytes}}{50 \text{ KB/sec}} = 2 \text{ msec}$$

 \therefore Total transfer time = 2 msec + 600 ns = 2 msec (approx.)

T11 : Solution

(b)

$$\text{Rotational speed} = 6000 \text{ rpm}$$

$$\text{Average latency} = \frac{\left(\frac{60 \text{ sec}}{6000}\right)}{2} = 5 \text{ msec}$$

$$\begin{aligned} \text{Avg access time} &= \text{Avg latency} + \text{Avg seek time} \\ &= 5 \text{ ms} + 10 \text{ ms} = 15 \text{ msec} \end{aligned}$$

Time to load 1 library is 15 msec

$$\text{Time to load 100 libraries} = 100 \times 15 \text{ msec} = 1.5 \text{ sec}$$

T12 : Solution

(d)

$$\text{RPM} = 600$$

$$\text{So, rotational delay} = \frac{60}{600} = 0.1 \text{ sec.}$$

In 1 rotation, we can transfer the whole data in a track.

$$\text{Tack capacity} = \text{Track} * \text{bytes per track} = 100 * 500 = 50,000 \text{ bytes.}$$

In 0.1 sec, we can transfer 50,000 bytes.

$$\text{Hence time to transfer 250 bytes} = 0.1 * \frac{250}{50000} = 0.5 \text{ ms}$$

$$\text{Avg. rotational delay} = 0.5 * \text{rotational delay} = 0.5 * 0.1 \text{ s} = 50 \text{ ms}$$

$$\text{Average seek time} = (0 + 1 + 2 + \dots + 499)/500$$

(as time to move between successive tracks is 1 ms and we have 500 such tracks) = $499 * 250/500 = 249.5$

Average time to transfer = Average seek time + Average rotational delay + Data transfer time

$$\text{Average time for transferring 250 bytes} = 249.5 + 50 + 0.5 = 300 \text{ ms.}$$

T13 : Solution

(14020)

$$\text{Seek time} = 4 \text{ ms}$$

$$\text{Rotational latency} = \frac{60}{10000} \text{ sec} = 6 \text{ ms}$$

$$\text{Average rotational latency} = \frac{R}{2} = 3 \text{ ms}$$

For each sector, we require Seek time + Rotational Latency + Transfer time.

In one rotation, 512 B × 600 data is read in 6 ms

One sector of 512 B can read in $6 \text{ ms}/600 = 0.01 \text{ ms}$ (transfer time for 1 sector)

Total time required for 1 sector = 4 ms + 3 ms + 0.01 = 7.01

For 2000 sectors $7.01 \times 2000 = 14020 \text{ ms}$

T14 : Solution**(456)**

Data count register = 16 bits

So, Count value = $2^{16} = 64$ K bytes

One time control, transfer — 64 K bytes

Number of controls to transfer — 29154 K bytes

So, number of time bus control required

$$= \left\lceil \frac{29154}{64} \right\rceil = 456$$

■ ■ ■ ■