GATE

MADE EASY Workbook 2026

Detailed Explanations of Try Yourself *Questions*

Computer Science & IT

Computer Organization and Architecture

MADE EASY PUBLICATIONS

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

(c)

	E	M
1	00111110	00000000010000000000000
1 bit	8 bits	23 bits

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

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

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

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

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

T3 : 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.

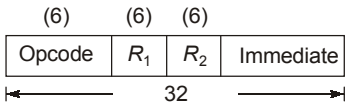
■■■■

2 Machine Instructions and Addressing Modes

Detailed Explanation of Try Yourself Questions

T1 : Solution

(16383)

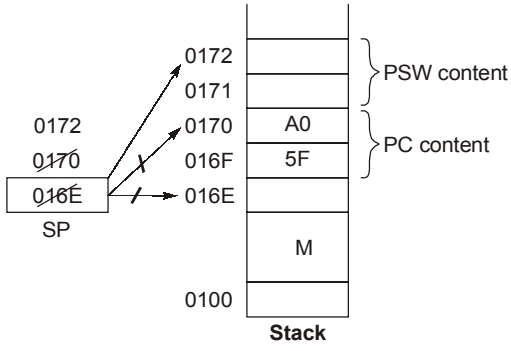


$32 - (6 + 6 + 6) = 14$ bits for immediate field

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

T2 : Solution

(d)



Just before CALL instruction execution, SP contains 016E

While CALL execution:

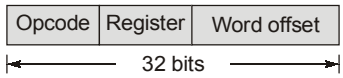
(*i*) PC contents are pushed i.e., SP incremented by 2 $\Rightarrow$ SP = 0170

(*ii*) PSW contents are pushed i.e., SP incremented by 2 $\Rightarrow$ SP = 0172

$\therefore$ 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 + 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.

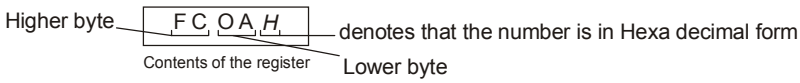
So Index Address bit = Register bit + Address bit
= 7 + 20 = 27

∴ $n + 7 + 27 = n + 34$ 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	
	I_4	4 cycles	5 times
	I_5	4 cycles	
	I_6	2 cycles	

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

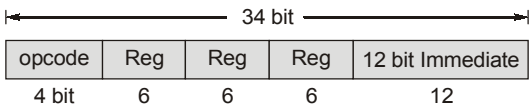
$$\text{Time} = 12 \text{ cycles} + 80 \text{ cycles} = 92 \text{ cycles}$$

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

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

T8 : Solution

(500)



One instruction size = 34 bit = 5 bytes

100 instruction occupies 500 bytes.

T9 : Solution

(b)

Average of time = {(0.2 × 0) + (0.2 × 0) + (0.2 × 4) + (0.1 × 8) + (0.17 × 6) + (0.13 × 6)} = 3.4 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

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

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

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.

T12 : Solution

(a)

$$X = (A + B \times C) / (D - E \times F)$$

1-Address Machine:

1. LOAD E
2. MUL F
3. STORE T
4. LOAD D
5. SUB T
6. STORE F
7. LOAD B
8. MUL C
9. ADD A
10. DIV T
11. STORE X

So, total 11 instruction in 1-address machine.

2-Address Machine:

1. MOV R_0, E
2. MUL R_0, F
3. MOV R_1, D
4. SUB R_1, R_0
5. MOV R_0, B
6. MUL R_0, C
7. ADD R_0, A
8. DIV R_0, R_1
9. MOV X, R_0

Total 9 instructions in 1-address machine.
So, 2 extra instruction is required in 1-address machine.

■■■■

3 CPU Design

Detailed Explanation of Try Yourself Questions

T1 : Solution

(a)

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

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

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

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

1 instruction $\Rightarrow$ 0.019375 μsec

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 (25) bits required i.e. 5.

For 2 control signals = 5 × 2 = 10 bits required.

■■■■

4 Instruction Pipelining

Detailed Explanation of Try Yourself Questions

T1 : Solution

(4)

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

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

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

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.

Speedup = $1.6\, n / 0.5\, n = 3.2$

T5 : 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.

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

$$ET = (K + n - 1) + t_p$$
$$= (4 + 13 - 1)\, 4 \text{ ns} = 64 \text{ ns}$$

T6 : Solution

(a)

A : $K = 6$, $n = 16$, $t_p = 8$ ns

$x = (K + n - 1)\, t_p = 168$ ns

B : $K = 9$, $n = 16$, $t_p = 6$ ns

$y = (9 + 16 - 1)\, 6 = 144$ ns

$x / y = 1.16$

T7 : 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

T8 : Solution

(b) S_2 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.

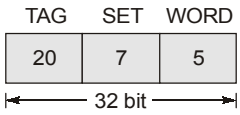
■■■■

5 Memory Organization

Detailed Explanation of Try Yourself Questions

T1 : Solution

(20)



WORD offset = 5 bits [$\because$ word length = 32 bits]

SET offset = 7 bits

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

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

$\therefore$ Number of TAG bits = 32 – (7 + 5) = 20 bits.

T2 : Solution

(d)

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

So it incurs lower cache miss penalty.

T3 : Solution

(d)

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

T4 : Solution

(10000)

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

So, one words store → 100 ns

? Number of words store → 1 msec

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

T5 : Solution

(1.68)

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

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 × 2.8 ns = 112 ns

Total time for instruction execution time for both read and write = 224 + 112 = 336 ns.

200 times, access takes _____ 336 ns.

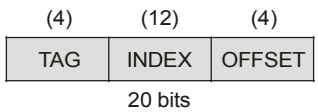
1 time, access takes _____ ?

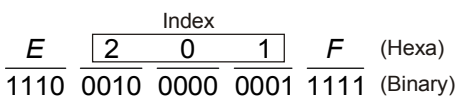
$$\text{Average memory access time} = \frac{336}{200} = 1.68\text{ ns}$$

T6 : Solution

(a)

$$\text{Block size} = 16\text{ bytes} = 2^4\text{ bytes}$$
$$\text{Offset} = 4\text{ bits}$$
$$\text{Index} = 12\text{ bits } [\because \text{ number of lines} = 2^{12}]$$
$$\text{Main memory size} = 2^{20}\text{ bytes}$$
$$\Rightarrow \quad \text{Physical address} = 20\text{ bits}$$





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

T7 : Solution

(b)

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

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

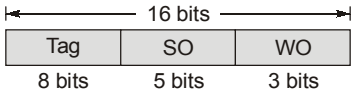
$2^S - 2$ goes to cache line $m - 2$

⋮

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

T8 : Solution

(b)



Therefore the block offset bits for the given addresses are:

10011 – 19

00100 – 4

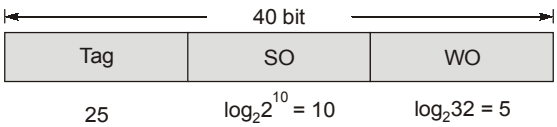
01101 – 13

T9 : Solution

(b)

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

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



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

T10 : Solution

(30)

1. Miss rate = 0.8, 10 MB

   T_c = ms

   T_d = 10 ms

   $T_{avg} = HT_c + (1 - H)(T_d + T_c) = 9$ ns

2. Miss rate = 0.6, 20 MB

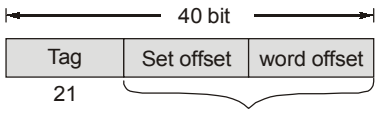
   T_c = 7 ms

3. Miss rate = 0.4, 30 MB
   T_{avg} = 5 ms
4. Miss rate = 0.35, 40 MB
   T_{avg} = 4.5 ms
5. Miss rate = 0.3, 50 MB
   T_{avg} = 4 ms
6. Miss rate = 0.25, 60 MB
   T_{avg} = 3.5 ms
7. Miss rate = 0.2, 70 MB
   T_{avg} = 3 ms
8. Miss rate = 0.15, 80 MB
   T_{avg} = 2.5 ms

So, 30 MB will the answer.

T11 : Solution

(24)



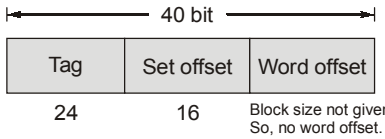
Cache index ($\log_2$512 K = 19 bits)
Block size not given so, cache index is considered

Or

Hypothetically, line size considered as cell size.

$\therefore$ Number of lines = 512 K

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



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-chinning 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

(a)

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

T4 : Solution

(d)

Cycle stealing mode:

Transferring control to I/O = 300 ns

Transferring control I/O to CPU = 300 ns

Time to transfer one byte = 300 + 700 + 300

= 1300 nsec = 1.3 μsec

Total time to transfer 100 bytes = 100 × 1.3 μsec = 130 μsec

T5 : Solution

(b)

It is transferring 16000 bits in 1 second

$$1 \text{ character} = 8 \text{ bits}$$

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

$$2000 \text{ characters} = 1 \text{ second}$$

1 character takes 500 micro seconds.

Processor accesses main memory in every μs

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

T6 : Solution

(c)

I/O transfers at 8 KB/sec

$\Rightarrow$ 1 byte it takes 125 μ sec

Each interrupt process takes 75 μ sec

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

T7 : 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.)

T8 : Solution

(b)

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

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

$$\text{Avg access time} = \text{Avg latency} + \text{Avg seek time}$$

$$= 5 \text{ ms} + 10 \text{ ms} = 15 \text{ msec}$$

Time to load 1 library is 15 msec

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

T9 : 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.

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

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

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

(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

Average time for transferring 250 bytes = 249.5 + 50 + 0.5 = 300 ms.

T10 : 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 ms/600 = 0.01 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 × 2000 = 14020 ms

T11 : Solution

(456)

$$\text{Data count register} = 16 \text{ bits}$$

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

One time control, transfer — 64 K bytes

Number of controls to transfer — 29154 K bytes

$$\text{So, number of time bus control required} = \left\lceil \frac{29154}{64} \right\rceil = 456$$

■■■■