



ESE | GATE | PSUs

ELECTRICAL ENGINEERING

COMPUTER FUNDAMENTALS

Volume - 1 : Study Material with Classroom Practice Questions



Computer Fundamentals

(Solutions for Classroom Practice Questions)

1. Number Systems & Boolean Algebra

01. Ans: (c)

Sol: Given 2's complement numbers of sign bits are x & y . z is the sign bit obtained by adding above two numbers.

\therefore Overflow is indicated by $= \bar{x}\bar{y}z + xy\bar{z}$

Examples

$$\begin{array}{r} 1. A = +7 \quad 0111 \\ B = +7 \quad 0111 \\ \underline{14} \quad 1110 \Rightarrow \bar{x}\bar{y}z \end{array}$$

$$\begin{array}{r} 2. A = +7 \quad 0111 \\ B = +5 \quad 0101 \\ \underline{12} \quad 1100 \Rightarrow \bar{x}\bar{y}z \end{array}$$

$$\begin{array}{r} 3. A = -7 \quad 1001 \\ B = -7 \quad 1001 \\ \underline{-14} \quad 10010 \Rightarrow xy\bar{z} \end{array}$$

$$\begin{array}{r} 4. A = -7 \quad 0111 \\ B = -5 \quad 0101 \\ \underline{-12} \quad 10100 \Rightarrow xy\bar{z} \end{array}$$

02. Ans: (b)

Sol: Truth table of XOR

A	B	o/p
0	0	0
0	1	1
1	0	1
1	1	0

Stage 1:

Given one i/p = 1 Always.

$$1 \quad X \quad \text{o/p}$$

$$1 \quad 0 \quad 1 = \bar{X}$$

$$1 \quad 1 \quad 0 = X$$

For First XOR gate o/p = \bar{X}

Stage 2:

$$\bar{X} \quad X \quad \text{o/p}$$

$$0 \quad 1 \quad 1$$

$$1 \quad 0 \quad 1$$

For second XOR gate o/p = 1.

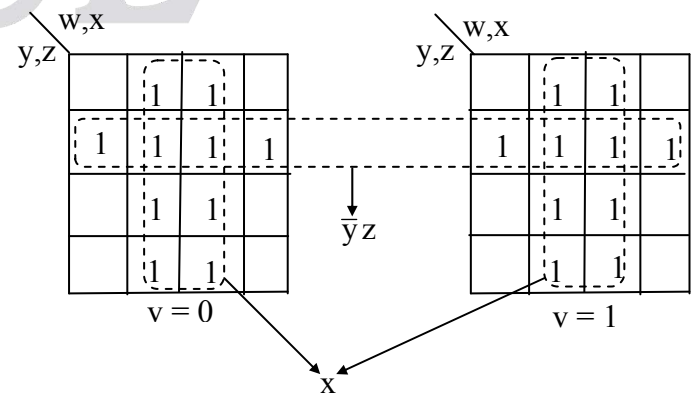
Similarly for third XOR gate o/p = \bar{X} & for fourth o/p = 1

For Even number of XOR gates o/p = 1

For 20 XOR gates cascaded o/p = 1.

03. Ans: (b)

Sol:



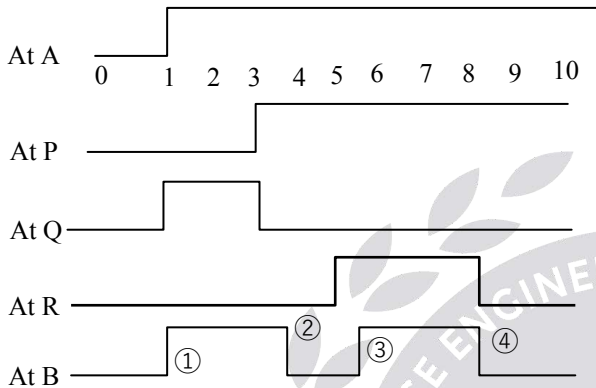
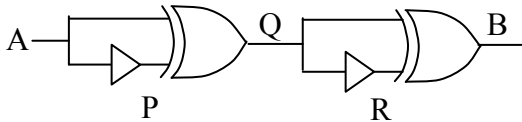


04. Ans: (c)

Sol: $f = f_1 f_2 + f_3$

05. Ans: (d)

Sol:



06. Ans: (c)

Sol: For all cases option A, B, D not satisfy.

07. Ans: (b)

Sol: $M(a,b,c) = ab + bc + ca$

$$\overline{M(a,b,c)} = \overline{bc} + \overline{ab} + \overline{ac}$$

$$M(a,b,\overline{c}) = ab + b\overline{c} + \overline{c}a$$

$$M(\overline{M(a,b,c)}, M(a,b,\overline{c}), c)$$

$$= (\overline{bc} + \overline{ab} + \overline{ac})(ab + b\overline{c} + \overline{c}a) + (ab + \overline{bc} + \overline{ca})c + (\overline{bc} + \overline{ab} + \overline{ac})c$$

$$= (\overline{bc} + \overline{ab} + \overline{ac})(ab + b\overline{c} + \overline{c}a) + (\overline{bc} + \overline{ab} + \overline{ac})(c) + abc$$

$$= ab\overline{c} + ab\overline{c} + abc + ab\overline{c}$$

$$= \overline{c}[ab + ab] + c[ab + ab]$$

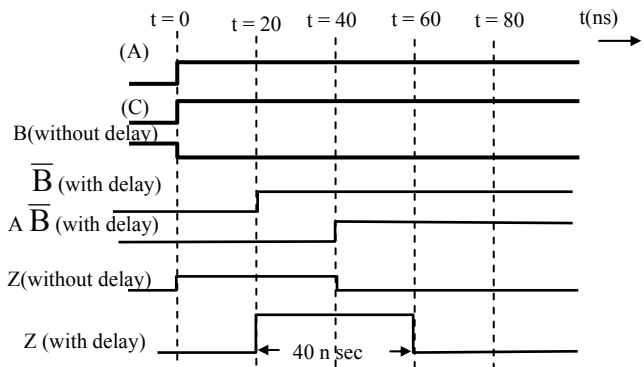
$$= \sum m(1,2,4,7)$$

$$\therefore M(x,y,z) = a \oplus b \oplus c$$

Where $x = \overline{M(a,b,c)}$, $y = M(a,b,\overline{c})$, $z = c$

08. Ans: (b)

Sol:



$\therefore Z$ is 1 for 40 nsec

09. Ans: (c)

Sol: Logic gates $\overline{X} + Y = \overline{X\overline{Y}} = \overline{XY_1}$

Where $Y_1 = \overline{Y}$

It is a NAND gate and thus the gate is 'Universal gate'.

10. Ans: (d)

Sol: A. $X = \overline{A} + \overline{B} = \overline{AB}$

B. $X = \overline{A+B}$

C. $X = \overline{\overline{A} + \overline{B}} = AB$

D. $X = \overline{\overline{A} \overline{B}} = A+B$

11. Ans: (a)

Sol: XOR gate is not a universal gate, because it is not possible to realize any Boolean function using only XOR gates.



13. Ans: (b)

Sol: (A) $ab + bc + ca + abc$
 $bc(1 + a) + ca + ab$
 $bc + ca + ab$

Inverse function $\overline{(ab + bc + ca)}$
 $= \bar{a}\bar{b} + \bar{b}\bar{c} + \bar{c}\bar{a}$

(B) $ab + \bar{a}\bar{b} + \bar{c}$

Inverse function = $\overline{ab + \bar{a}\bar{b} + \bar{c}}$
 $= (\bar{a} + \bar{b})(a + b)c$
 $= (\bar{a}b + a\bar{b})c$
 $= (a \oplus b)c$

(C) $(a + bc)$

Inverse function = $\overline{a + bc}$
 $= \bar{a}(\bar{b} + \bar{c})$

(D) $(\bar{a} + \bar{b} + \bar{c})(a + \bar{b} + \bar{c})(\bar{a} + \bar{b} + \bar{c})$

Inverse function
 $\overline{(\bar{a} + \bar{b} + \bar{c})(a + \bar{b} + \bar{c})(\bar{a} + \bar{b} + \bar{c})}$
 $= abc + \bar{a}bc + ab\bar{c}$

2. Central Processing Unit (CPU)

01. Ans: (d)

Sol: The given program is:

Instruction	Operation	Instruction Size (No. of words)
MOV R1, (300)	R1 ← M[3000]	2
LOOP: MOV R2, (R3)	R2 ← M[R3]	1
ADD R2, R1	R2 ← R2 + R1	1
MOV (R3), R2	M[R3] ← R2	1
INC R3	R3 ← R3 + 1	1
DEC R1	R1 ← R1 - 1	1
BNZ LOOP	Branch on not zero	2
HALT	Stop	

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 ← M[3000]	1
MOV R2, (R3)	R2 ← M[R3]	10 (Loop is repeated 10 times)
MOV (R3), R2	M[R3] ← R2	10 (loop is repeated 10 times)

Total number of memory references are: 21

02. 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.



03. 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		
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	

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.

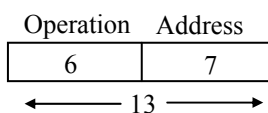
04. Ans: (d)

Sol: Opcode size = 13-bit

Control memory size = 7-bit

$$= 128 \text{ word} = 2^7$$

Maximum number of one address instructions to be formulated



$$= 2^6 = 64$$

∴ Remaining number of zero address instructions to be formulated

$$= (64 - 32) \times 2^7$$

$$= 4096$$

05. Ans: (c)

Sol: Stack works on LIFO.

06. Ans: (b)

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

07. Ans: (c)

Sol: Base register addressing permits relocation without any change in the code.

08. 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 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.

09. 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 R₅ in a processor which is having 16 Register from R₀ to R₁₅, it's Address field is '0101', and for implied Register; no address is specified in the instruction.



10. Ans: (d)

Sol: Here R_2 will act as base or index register and 20 is the displacement.

11. Ans: (a)

Sol: Word size = 32 bit

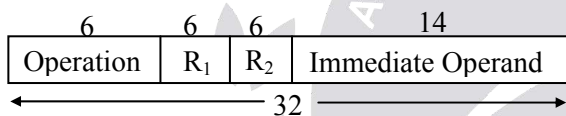
Number of CPU Registers = $64 = 2^6$

So, for addressing a Register 6 bits are needed.

Instruction Opcode size is 32-bits.

Number of supporting Instructions = 50, 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.

12. Ans: (d)

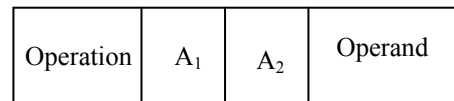
Sol: Maximum number of two address instructions = 24.

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

∴ Maximum number of one address instructions: $(2^4 - n) \times 2^6$

13. Ans: 16

Sol: ←----- 32 -----→



$\log_2 40$ $\log_2 24$ $\log_2 24$
6 5 5 xxx

∴ $32 - 16 = 16$

14. Ans: 500

Sol: One instruction needs 34 bit,

So number of bytes needed = 5

Program size = 100

∴ Size of the memory in bytes = 500

3. I/O Organization

01. Ans: (b)

Sol: External interrupts comes from input-output devices, from a timing device, from a circuit monitoring the power supply or internal from other external source

Example: I/O devices requesting transfer of data

- A DMA controller manages the data transfer
- A DMA controller can be implemented as separated controller from the processor or integrated controller in the processor.
- Based on the request length, the DMA controller optimizes transfer performance between source and destination with different external data by widths



02. Ans: (d)

Sol: The DMA is the most suitable for Hard disk

03. Ans: (b)

Sol: On Receiving an interrupt from an I/O device, the CPU branches off to the interrupt service routine after completion of the current instruction.

04. Ans: (c)

Sol: In microprocessor based systems DMA facility is required to increase the speed of data transfer between memory and the I/O devices.

05. Ans: (b)

Sol: An I/O processor controls the flow of information between main memory and I/O devices. It is the extension of concept of DMA.

4. Memory Organization & Secondary Memory

01. Ans: (d)

Sol: **Cache memory** - A special very- high-speed memory called a cache is sometimes used to increase the speed of processing by making current programs and data available to the CPU at a rapid rate. The cache memory is employed in computer systems to compensate for the speed differential between main memory access time and processor Logic.

02. Ans: (c)

Sol: The use of a cache in computer system increases the Average speed of memory access.

05. Ans: (b)

Sol: Average Latency time = $\frac{1}{2}$ rotation time

Speed = 7200 rpm \Rightarrow 120 rps

For 1 rotation = $\frac{1}{120} = 8.33$ ms

For $\frac{1}{2}$ rotation = 4.166 ms

06. Ans: (a)

Sol: **LIFO:** Last In First Out

07. Ans: (d)

Sol: Hit ratio is defined as the number of hits divided by the total number of CPU references to the memory (hits plus misses)

$$T_{avg} = H C + (1 - H)M$$

Where H = hit ratio of cache memory

C = time to access information in cache memory

M = Miss penalty

= main memory access time

+ cache memory access time.

08. Ans: (c)

Sol: DRAM access much slower than SRAM

- More bits \rightarrow longer wires
- Buffered access with two-level addressing
- SRAM access latency : 2-3 ns
- DRAM access latency 20-35 ns
- Static RAM – 10 nano sec
- Hard disk for personal computers boast access times of about a to 15 ms
- 200 times slower than average DRAM.



09. Ans: (d)

Sol: Cache size = 4 K word

One set has 4 blocks

$$\therefore \text{Number of sets} = \frac{\text{Total number of blocks}}{4}$$

$$\text{Number of words} = 2^6$$

$$\Rightarrow W = 6$$

$$\text{Number of cache blocks} = \frac{4K}{64} = \frac{2^{12}}{2^6} = 2^6$$

$$\text{Number of cache sets} = \frac{2^6}{4} = 16 = 2^4$$

$$S = 4$$

10. Ans: (c)

Sol: Cache memory is used to improve the overall system performance.

11. Ans: (a)

Sol: Word size = 16-bit

Access time = 80 ns

$$\begin{aligned} \text{Number of bytes to be transferred} &= 1024 \text{ B} \\ &= 512 \text{ Words.} \end{aligned}$$

$$\text{Total time} = 512 \times 80 \text{ ns} = 40960 \text{ ns} = 40.96 \mu\text{s}$$

12. Ans : (c)

$$\text{Sol: } 2^{12} \times 8 = 4096 \times 8 = 32768\text{-bits}$$

5. Basics of Operating Systems

02. Ans: (d)

Sol: Primary goal of Operating System is 'Convenience'. But primary goal of Real Time Operating System is 'Reliability'.

03. Ans: (a)

Sol: Those systems which allows more than one process execution at a time, are called multiprogramming systems.

Uni-programming means only one processor.

04. Ans: (a)

Sol: MS-DOS is single user operating system

05. Ans: (c)

Sol: Multiprogramming systems execute more jobs in the same time

06. Ans: (c)

Sol: Environment for execution of user programs is provided by Operating system

07. Ans: (d)

Sol: Device interface is not a user interface of operating system

08. Ans: (a)

Sol: Networking is not provided by operating system

10. Ans: (d)

Sol: In Real-time OS, the response time is very critical.



12. **Ans: (d)**

Sol: Multiprogramming is introduced as single job cannot keep both I/O device and CPU busy.

16. **Ans: (d)**

Sol: Bootstrap Loader is executed when a computer is turned off or restarted.

17. **Ans: (c)**

Sol: Swapping is used to move processes from main memory to disk

6. Memory Systems & Virtual Memory

01. **Ans: (a)**

Sol: Memory management unit does Runtime mapping from virtual to physical address

02. **Ans: (b)**

Sol: Paging is used to store and retrieve data from secondary storage for use in main memory

03. **Ans: (a)**

Sol: Program always deals with Logical address

04. **Ans: (a)**

Sol: Operating System maintains the page table for each process

05. **Ans: (a)**

Sol: With relocation and limit registers, each logical address must be less than the limit register.

06. **Ans: (a)**

Sol: When memory is divided into several fixed sized partitions, each partition may contain exactly one process

07. **Ans: (a)**

Sol: In fixed sized partition, the degree of multiprogramming is bounded by the number of partitions.

10. **Ans: (c)**

Sol: A virtual-memory block is known as page, and a virtual-memory miss is called a Page fault

7. File Systems

01. **Ans: (c)**

Sol: Make an entry for new file in directory and allocate the space in file system to create a file in memory.

02. **Ans: (b)**

Sol: File extension is used to represent file type.

03. **Ans: (c)**

Sol: Mounting of File system is attaching portion of the file system into a directory structure

04. **Ans: (a)**

Sol: Fragmentation occurs when unused space or single file are not contiguous



06. Ans: (a)

Sol: The directory can be viewed as a symbol table that translates file names into their directory entries.

07. Ans: (b)

Sol: In the single level directory all files are contained in the same directory

08. Ans: (b)

Sol: In the single level directory all files must have unique names.

09. Ans: (c)

Sol: In the tree structured directories the tree has the root directory

10. Ans: (a)

Sol: A folder is a named location on a disk where files are stored.

8. Basics of Networking

01. Ans: (b)

Sol: Data Link Layer is responsible for decoding bit stream into **frames**.

03. Ans: (c)

Sol: Given: At each layer, n bits of information is added/appended.

$$= nh$$

Total message = original message+overhead

$$= m + nh$$

$$\% \text{ of overhead} = \frac{nh}{m + nh} \times 100$$

04. Ans: (c)

Sol: Frame encapsulates packet

05. Ans: (d)

Sol: Network layer 4 times, Data link layer 6 times.

AL				AL
TL				TL
NL	NL		NL	NL
DL	DL	DL	DL	DL
PL	PL	PL	PL	PL

NL = 4 Times

DL = 6 times

06. Ans: (d)

Sol: The End-to-End delivery of the entire message is the responsibility of the Transport layer.

07. Ans: (b)

Sol: As the data packets moves from the upper to lower layers, headers are added.

08. Ans: (a)

Sol: Data link layer: Ensures reliable transport of data over a Physical point-to-point link

Network layer: Routes data from one network node to the next

Transport layer: Allows end - to - end communication between two processes.

09. Ans: (d)

Sol: Segmentation & reassembly is responsibility of transport layer.



10. Ans: (d)

Sol: when data transmitted from device A to device B, the header from A's layer 4 is read by B's Transport layer.

11. Ans: (c)

Sol: Ethernet adapter receives all frames and accepts frames if it has been placed in promiscuous mode.

12. Ans: (b)

Sol: Ethernet implements Connectionless service for its operation

13. Ans: (d)

Sol: TCP/IP model does not have session layer and presentation layer but OSI model have these layer.

14. Ans: (a)

Sol: The number of layers in Internet protocol stack is 5.

15. Ans: (b)

Sol: ISO OSI reference model has 7 layers.

16. Ans: (a)

Sol: The physical layer concerns with Bit - by - bit delivery

17. Ans: (a)

Sol: The data link layer takes the packets from Network layer and encapsulates them into frames for transmission

18. Ans: (b)

Sol: The network layer protocol of internet is Internet protocol

19. Ans: (a)

Sol: User datagram protocol is called connectionless because all UDP packets are treated independently by transport layer

20. Ans: (a)

Sol: SMTP protocol deals with emails in application layer.

21. Ans: (a)

Sol: Ethernet frame consists of MAC address

22. Ans: (a)

Sol: A point to point protocol over Ethernet is a network protocol for encapsulates PPP frames inside Ethernet frames.

23. Ans: (a)

Sol: In Star topology there is a central controller or hub.

24. Ans: (b)

Sol: WAN is a Data communication system spanning states, countries, or the whole world.



**9. C Language
(Elements of PL: Typical Examples)**

01. Ans: (a)

Sol: A program that translates a high-level language program into a object module is called a compiler.

02. Ans: (b)

Sol: double is used to define BIG floating point numbers. It reserves twice the storage for the number.

03. Ans: (a)

Sol: Characters are assigned variables by using single quotes, where strings we use double quotes.

05. Ans: (c)

Sol: printf (“%d %c%f”, 23, ‘z’, 4.1);

07. Ans: (b)

Sol: The value of x is decremented by one after the print statement.

08. Ans: (c)

Sol: Left side of assignment operator allows only one variable, not a expression
In option (b) the meaning of
 $a * = b$ gives $a = a * b$
In option (d) $b = 0$ is first evaluate the result is assign to ‘a’
Because ‘=’ operator is right to left associativity.
Finally option (c) is invalid

09. Ans: (b)

Sol: except option (b) remaining all expressions having different precedence levels but in option (b) ‘*’, ‘/’ operators are having same precedence levels, so for evaluation we need associativity. Here associativity is ‘left to right’.

10. Ans: (a)

Sol: Option (b), (c), (d) are logical operators. Where as option (a) is a statement.

11. Ans: (a)

Sol: Option (b), (c), (d) are relational operators which compare the two variables and gives result in the form of boolean data type i.e true (or) false.

Where as option (a) is assignment operator which assign the value to variable.

12. Ans: (d)

Sol: The complement of equal (==) operator is not equal (!=)

13. Ans: (b)

Sol: for, while and do.....while are looping constructs but switch case is a statement.

14. Ans: (c)

Sol: Before entering into the loop while checks the condition i.e., while is pre-test loop, where ‘do-while’ is post test loop.

15. Ans: (a)

Sol: The address of (&) operator is used to extract the address for a variable.



16. Ans: (c)

Sol: The syntax `int *p;` is used to declare a pointer variable to an integer.

17. Ans: (c)

Sol: For creating a pointer we use '*' and address of variable we use '&'.

18. Ans: (c)

Sol: We can increment the pointer value and comparing but we can't divide `prt + 5` is the points value 5 elements away. `++ ptr` pre-increment of pointer & that points to next address.

19. Ans: (b)

Sol: A variable length string can be controlled by a length or a delimiter. The C language uses a null to terminate variable length strings.

20. Ans: (c)

Sol: The statement `employee.name` is used to declare the name of the employee.

21. Ans: (b)

Sol: For loop iterates 10 times because every time 'i' value getting to double, means 'i' initially 1, then 2, 4, 8, 16, 32, 64, 128, 256, 512 finally 1024 give false value to for loop. So total 10 times of iteration.

22. Ans: (c)

Sol: For every 'i' value 'j' loop iterates 10 times and 'i' will iterate 10 times. So $10 \times 10 = 100$ iterations possible.

